PREDICTING ENERGY CONSUMPTION PATTERN BASED ON TOP TRENDING VIDEOS YOUTUBE 2021 USING MACHINE LEARNING TECHNIQUES

BY

NG JIUN SHEN

A REPORT

SUBMITTED TO

Universiti Tunku Abdul Rahman

in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)

Faculty of Information and Communication Technology

(Kampar Campus)

JUN 2022

UNIVERSITI TUNKU ABDUL RAHMAN

REPORT STATUS DECLARATION FORM

Title: **Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning technique**

Academic Session: June 2022

I NG JIUN SHEN

declare that I allow this Final Year Project Report to be kept in
Universiti Tunku Abdul Rahman Library subject to the regulations as follows:

1. The dissertation is a property of the Library.
2. The Library is allowed to make copies of this dissertation for academic purposes.

Verified by,

(Author's signature) (Supervisor's signature)

Address:
Jalan University,
Bandar Barat,
31900, Kampar, Perak

Ms Tseu Kwan Lee
Supervisor's name

Date: 9 SEPTEMBER 2022 **Date**: 9 SEPTEMBER 2022

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

UNIVERSITI TUNKU ABDUL RAHMAN

Date: 9 SEPTEMBER 2022

SUBMISSION OF FINAL YEAR PROJECT

It is hereby certified that _____**Ng Jiun Shen**_____ (ID No: __**20ACB00362**__) has completed this final year project entitled " Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning technique " under the supervision of Ms Tseu Kwan Lee (Supervisor) from the Department of Computer Science, Faculty of Information and Communication Technology.

I understand that University will upload softcopy of my final year project in pdf format into UTAR Institutional Repository, which may be made accessible to UTAR community and public.

Yours truly,

Ng Jiun Shen

DECLARATION OF ORIGINALITY

I declare that this report entitled "**Predicting Energy Consumption Pattern based on Top Trending Videos YouTube 2021 using Machine Learning Techniques**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : _____ _____

Name : _____NG JIUN SHEN _____

Date : 9 SEPTEMBER 2022

ACKNOWLEDGEMENTS

I would like to express my sincere thanks and appreciation to my supervisor, Ms. Tseu Kwan Lee who has given me this bright opportunity to engage in an energy predicting project. It is my first step to establishing a career in the machine language field. A million thanks to you.

Next, I would like to thank my friends for giving me support. Finally, I must say thanks to my parents and my family for their love, support, and continuous encouragement throughout the course.

ABSTRACT

This project is about predicting energy consumption patterns based on trending videos on YouTube 2021 by using machine learning techniques. It is important to be aware of how much energy has been consumed while an individual is streaming or watching a video on YouTube on a mobile device. This is because the users will learn how much energy they have wasted as a result of these actions, as well as what the company should do to limit the amount of energy lost. This project will cover the methodology, concept, and design of utilising machine learning language to anticipate energy consumption patterns. The process of putting a machine learning pipeline into action will be explored and analysed. The flow of this project will be presented as follows: exploring the data, pre-processing, model selection and building the model, model validation, fine-tuning the model, testing performance, and finally, simulating it. Jupyter notebook was chosen as the tool for predicting the model, and Python was chosen as the coding language. To help collect the data, AccuBattery was chosen to calculate the energy consumption per video. In this project, several models will be presented and analysed, with the normal equation in Linear Regression will be the algorithm used to simulate it. Furthermore, the data will be clustered in this project utilising threshold-based approaches. Streamlit.io application was chosen as the website to represent the coding. At the end of this project, a formula equation and graph will be thoroughly discussed and analysed for projecting energy consumption patterns based on the most trending videos on YouTube in 2021.

TABLE OF CONTENTS

TITLE PAGE **i**
REPORT STATUS DECLARATION FORM **ii**
FYP THESIS SUBMISSION FORM **iii**
DECLARATION OF ORIGINALITY **iv**
ACKNOWLEDGEMENTS **v**
ABSTRACT **vi**
TABLE OF CONTENTS **vii**
LIST OF FIGURES **x**
LIST OF TABLES **xi**
LIST OF SYMBOLS **xii**
LIST OF ABBREVIATIONS **xiii**

CHAPTER 1 INTRODUCTION **1**
1.1 Problem Statement and Motivation 2
1.1.1 Problem Statement 2
1.1.2 Motivation 3
1.2 Objectives 4
1.3 Project Scope 5
1.4 Contributions 6
1.5 Report Organization 7

CHAPTER 2 LITERATURE REVIEW **8**
2.1 Implementation of Machine learning techniques in predicting energy consumption 8
2.1.1 Implementation of Linear Regression in predicting energy consumption 12
2.2 Strength and Weakness 14
2.2.1 Strength 14
2.2.1 Weakness 15

CHAPTER 3 SYSTEM MODEL (FOR RESEARCH-BASED PROJECT) **16**

3.1 System Design Equation 16

3.2 System Design Diagram 18

3.2.1 System Architecture Diagram 18

3.2.2 Use Case Diagram and Description 19

3.2.3 Activity Diagram 22

CHAPTER 4 SYSTEM DESIGN 23

4.1 System Architecture Diagram 23

4.2 Coding Setup Diagram 27

4.3 Website Setup Diagram 28

CHAPTER 5 SYSTEM IMPLEMENTATION **29**

5.1 System Setup 29

5.1.1 Hardware Setup 29

5.1.2 Software Setup 29

5.2 System and Configuration 30

5.3 Machine Learning Operation 31

5.3.1 Data Exploration 31

5.3.2 Data Preprocessing 33

5.3.3 Model Selection and Building the Model 35

5.3.4 Model Validation 39

5.3.5 Fine-tune the Model 41

5.3.6 Test Performance 44

5.3.7 Simulation using Python – predicting energy consumption 46

5.4 Website Operation **52**

5.4.1 Main Page 52

5.4.2 Explore Page 52

5.4.3 Predict Page 53

5.5 Implementation Issues and Challenges **54**

5.6 Concluding Remark **54**

CHAPTER 6 SYSTEM EVALUATION AND DISCUSSION **55**

6.1 Results on Energy Consumption Pattern 55

6.2 Project Challenges 56

6.3 Objective Evaluation 57

6.4 Concluding Remark 58

CHAPTER 7 CONCLUSION AND RECOMMENDATION **59**

7.1 Conclusion 59

7.2 Recommendation 60

REFERENCES **61**

APPENDIX A – DATA SOURCE **63**

APPENDIX B – DATA EXPLORATION CODING **64**

APPENDIX C – DATA EXPLORATION RESULT **65**

APPENDIX D – WEB PAGE CODING **68**

APPENDIX E – WEB PAGE SIMULATION **74**

WEEKLY LOG **77**

POSTER **84**

PLAGIARISM CHECK RESULT **85**

FYP2 CHECKLIST **92**

LIST OF FIGURES

Figure Number	Title	Page
Figure 2-1-1	Behaviour of the DRL-based policy, q=0.5	11
Figure 2-1-1-1	DCT coefficient of luminance and power(W) of videos	13
Figure 3-2-1-1	System Architecture Diagram	18
Figure 3-2-2-1	Use Case Diagram	19
Figure 3-2-3-1	Activity Diagram	22
Figure 4-1-1	Flow Chart Diagram	23
Figure 4-2-1	Coding Setup Diagram	27
Figure 4-3-1	Website Setup Diagram	28
Figure 5-3-1-1	Relationship between views and total energy attributes	32
Figure 5-3-2-1	Result before and after preprocessing	35
Figure 5-3-3-1	Linear Regression model performance result	37
Figure 5-3-3-2	Decision Tree Regressor model performance result	38
Figure 5-3-3-3	Random Forest Regressor model performance result	39
Figure 5-3-4-1	Linear Regression model validation result	40
Figure 5-3-4-2	Decision Tree Regressor model validation result	40
Figure 5-3-4-3	Random Forest Regressor model performance result	41
Figure 5-3-5-1	Best grid search model and feature	43
Figure 5-3-5-2	Best random search model and feature	44
Figure 5-3-6-1	Result of handling the missing value and RMSE	45
Figure 5-3-7-1	Energy consumption pattern on Linear Regression model	47
Figure 5-3-7-2	Comparison of model’s performance	49
Figure 5-3-7-3	Final energy consumption pattern	51
Figure 6-1-1	YouTube 2021 Energy Consumption Pattern	55

LIST OF TABLES

Table Number	Title	Page
Table 2-1-1	Electricity Dataset	8
Table 2-1-2	Performance of metric learning on the test set	10
Table 2-1-1-1	Overview of linear regression techniques	12
Table 3-2-2-1	Use Case Legend	20
Table 3-2-2-2	Use Case Description – Explore Page	21
Table 3-2-2-3	Use Case Description – Predict Page	21
Table 5-1-1-1	Specifications of laptop	29
Table 5-3-4-1	Validation value of several machine learning models	43
Table 6-3-1	Objective Evaluation	71

LIST OF SYMBOLS

TWh	Terawatt hour
W	Watt (power)
θ	Theta
E	Energy (Joules)
V	Voltage
mAh	Current
e	Exponent of 10
$\bar{x}$	Mean
σ	Standard deviation

LIST OF ABBREVIATIONS

MLP	Multilayer Perception
SVM	Support Vector Machine
RMSE	Root Mean Square Error
LR	Linear Regression
NMI	Normalized Mutual Information
DRL	Deep Reinforcement Learning
SLRM	Simple Linear Regression Model
MLRM	Multiple Linear Regression Model
DCT	Discrete Cosine Transform
NN	Neural Network
NRMSE	Normalized Root Mean Square Error

Chapter 1

This chapter presents the background, problem statement, and motivation of the research. Other than that, the objectives, project scope, and contribution to the field had also been introduced.

Introduction

In today's technological world, new techniques and technology are always being developed and renewed. Every person and corporation should be aware of how much energy they have utilized on any given day and in any given location. As a result, they will be able to compare energy usage to the most recent technology and will always be able to obtain the best technology for themselves or the firm.

If a public-use model for projecting energy usage is available, the preceding difficulty can be alleviated. This research project will use machine learning to present a model for predicting energy consumption patterns based on top popular videos on YouTube in 2021. This is because a statistic showed that on the YouTube platform, there are over 1 billion viewing hours a day, and it would consume a total of over 600 TWh a year which is almost 2.5 per cent of global electricity used [1].

Using the Linear Regression (LR) method, the energy consumption pattern will be examined and drawn out. The result of how much energy individuals have to watch a YouTube video will be analysed and discussed. Aside from that, the threshold method will be used to cluster the events. These two processes will be incorporated into the model to obtain a more accurate performance prediction model. This project is going undertake for solving the issue of wastage of energy consumption by individuals.

1.1 Problem Statement and Motivation

1.1.1 Problem Statement

The first issue is that people are currently unaware of how much energy they utilize daily. They are uninterested in calculating their daily energy waste. For example, a study has stated when doing a national online survey, most of the participants have given a contrasting idea to experts' recommendations for saving energy [2]. After that, those who have never studied computers will also have no idea how to calculate how much energy they used while watching a video. The general public is unsure how to compute energy consumption and examine how much energy has been consumed. When the user watches the video, no information or guidance is displayed.

The next issue is that the general public lacks information to compare which program consumes more energy when watching a video. When a user compares applications, they will lack knowledge regarding energy use and hence may choose the program that consumes the most energy. For example, on Twitter and Reddit, users are unable to compare energy consumption.

The final problem statement in this project will be how energy usage contributes to IT and its impact on the environment. People nowadays have benefited greatly from technological advancements like the mobile phone and computer. Due to the pandemic, people now rely more on these machines than on paper. As a result, there has been a significant impact and the rate of non-renewable energy consumption has accelerated. In addition to the non-renewable energy shortage, it has accelerated global warming and affected the ecological and economic systems [3].

1.1.2 Motivation

The goal of this research is to offer a new model that uses LR to predict energy consumption trends based on the top trending video on YouTube in 2021. Although this project has just forecasted data based on the most popular YouTube videos in 2021. However, it can be used as a starting point for estimating a video's energy use in future research. The researcher can use various machine learning techniques to forecast video energy usage patterns.

This study is significant since environmental consciousness is becoming more prominent in recent years. People begin to safeguard their children and provide a better living environment for them. Energy demand, on the other hand, has been rising in recent years. More advanced technologies and machines are being developed, and they will mostly require energy. It has increased carbon dioxide emissions. For example, by comparing the carbon dioxide emission in the world, the year 1990 presented a higher result than the year 1970 [4]. As a result, this study is being conducted so that the users can forecast the amount of energy used by the video the users view regularly and reduce the amount of energy consumed by watching videos.

1.2 Objectives

The thesis aims to propose LR on predicting energy consumption patterns based on top trending videos on YouTube 2021. Furthermore, this objective can be divided into several sub-objectives of the project.

In this report project, the sub-objectives will be:

(1) Collect and cluster the data set from YouTube.

(2) Work out the data set using Linear Regression.

(3) Determine the accuracy of the predicted energy consumption pattern.

(4) Created an energy consumption pattern web page

For the first sub-objective, after the dataset has been imported, will check that all of the data is right by using the threshold method. For example, all of the data will be in the same unit, and the data set will have no empty values. The second sub-objective is to use the LR approach to analyse the data set and anticipate the energy usage trend. The third sub-objective will be to determine the precision of the expected energy consumption pattern. It helps to see how accurate the energy consumption pattern is by using LR. In the end, an energy consumption pattern web page will be created successfully.

1.3 Project Scope

This project will focus on using the LR method to predict energy consumption patterns based on top trending videos on YouTube 2021. This is because it can aid in determining the strength of the relationship between the outcome and the predictor variable. It can also aid in quantifying when a new dataset is added [5].

Python and CSS have been used in this project for the coding of machine learning and web design throughout the implementation phase. Several tools have also been used to aid in the development of this project. AccuBattery tool for collecting data on energy usage[6], Jupyter notebook as the platform for developing the code[7], GitHub for storing the code [8], and the code will be related to Streamlit.io for displaying the code's web page[9].

At the end of this project, a model system using the LR method for predicting energy consumption patterns based on top trending videos on YouTube 2021 will be developed. The model will be functioning to handle the video data which are taken from YouTube and then build up an energy consumption pattern of it. There will be a successfully functioning website for the code.

Aside from that, this model can be applied to any future YouTube years to anticipate the total energy utilised by an individual when using a mobile device. Since the data is collected from multiple distinct mobile models, any mobile will work well on it.

1.4 Contribution

This experiment and analysis confirm the feasibility of the proposed Learning Regression methods for predicting the energy consumption patterns of a video. Firstly, the energy consumption pattern of a video is shown to be highly accurate to the assumption and it provides a foundation for further research on predicting energy consumption patterns based on machine learning. Second, Learning Regression is explained extensively in this study. By referring to this project, the researcher can conduct more research on the application of LR.

The project has also made a significant contribution to raising public awareness. This is because people can learn how energy consumption works when watching a video by studying this project. They will be able to predict and take action to reduce the energy used to watch videos. The general public will make efforts to reduce their use of non-renewable energy and save money.

1.5 Report Organization

The details of this research will be further breakdown into the following chapters. Chapter 2 will analysis about the related background of this project. Aside from that, the literature section of Chapter 2 has been split into two sections, one for discussing the LR technique and another one for comparing other Machine Learning techniques like Deep Reinforcement Learning with the LR technique. It gives a reference to find the parts needed more quickly. At the end of Chapter 2, will discuss the summary of the strengths and weaknesses of the literature that has been reviewed.

The proposed method used in this project will be discussed in Chapter 3. It will go through the system required for the project as well as any system design equations or diagrams that were used.

A flow chart diagram will be presented for chapter 4, and this chapter will go over each block's functionality. The machine learning coding flow will be described.

In addition, the report will discuss what has been accomplished so far in this project in Chapter 5 on how to begin by setting up this project and defining which hardware and software tools would use. The coding will be discussed in chapter 5. It will also go into the implementation issues and challenges. A concluding remark has been included in this chapter.

Chapter 6 is a chapter to present the results of the work thus far. The prediction algorithm and the energy consumption pattern based on top trending videos on YouTube 2021 using machine learning techniques will be plotted out. It will also discuss the challenges faced in the project. An objectives evaluation and concluding remark have been included in this chapter.

In the final chapter, Chapter 7, a conclusion will be reported including what has been accomplished thus far in this project. It will describe the summary of the project. Aspects of the project about which yet to complete and what is the plan to accomplish or improve in the future project will be discussed here.

Chapter 2
Literature Review

2.1 Implementation of Machine learning techniques in predicting energy consumption

Authors [10] provided a review study that looked at which machine learning models were most accurate in predicting electricity use.

In this study, LR and Support Vector Regression yielded the maximum accuracy of 85.7 per cent in determining power use. The K-Nearest Neighbors and Decision Tree algorithms follow. Random Forest has the worst accuracy (79.9 per cent) when it comes to estimating power use. A one-year data collection from a shoe store was used to train machine learning models for predicting electrical use. The table for the data set is the one displayed below.

Date	Day of week	Day of year	Week	Weekend	Prev. day electricity consumption	Electricity consumption
2016-01-05	1	5	1	0	26.0950	20.374
2016-01-06	2	6	1	0	16.1960	12.018
2016-01-07	3	7	1	0	11.6495	11.281
2016-01-08	4	8	1	0	11.1915	11.102
2016-01-09	5	9	1	1	11.2080	11.314

Table 2-1-1 Electricity Dataset

Author [11] proposed a report article that used several machines learning techniques, including Multilayer perception(MLP), Support Vector Machine(SVM), and decision tree learning solely, to estimate energy consumption using historical data and Twitter.

The author has done several experiments to discuss the paper: historical electrical load statistics, and smiley-based mood prediction. M5P' decision tree was chosen as the algorithm to estimate energy consumption from decision tree learning. It is a model which extended from the algorithm of M5 and is reconstructed by the author [12]. the result of the different classifiers is verified by comparing their root mean squared error (RMSE).

The author will create an energy consumption of electrical load statistics in the first section, with data classified as day, weekday, year, month, week, holiday, and others. The results show that MLP has the best RMSE of 96.19 per cent based on one year of data. In the next section, the author forecasts the number of emoji occurrences on Twitter for an hour. The emojis used in the experiment were happy, laugh, sad, and cry. The RMSE of MLP and M5P decision tree was quite high as a result of this part. MLP has an 1881 value, while M5P has 1896. However, after experimenting with section two, an issue was discovered: the M5P classifier consistently predicted the same result.

In the summary of this paper, after incorporating the machine learning language into the data, the author has determined that using MLP to estimate energy usage in one year of historical data and Twitter will get the best results. SVM and decision tree learning will perform better than MLP if the training year is set to two years.

Author [13] discussed how clustering events of social media can be clustered based on similarity metric learning.

This study will learn how to determine which metrics perform best when clustering several events and documents. There are other six approaches to similarity metric learning techniques. The approaches covered include all-text classification, tag classification, ensemble-based approach to combining partitions, ensemble-based approach to combining similarity scores, Support Vector Machine (CLASS-SVM), and Logistic Regression (CLASS-Logistic Regression). These techniques are applied to the learning of multi-feature similarity metrics for media documents. Data is selected from a large size, real-world collection of event photographs from Flickr. The final result is shown in Table 2-1-2.

Algorithm	NMI	B-Cubed
All-Text	0.9240	0.7697
Tags	0.9229	0.7676
ENS-PART	0.9296	0.7819
ENS-SIM	0.9322	0.7861
CLASS-SVM	0.9425	0.8095
CLASS-LR	**0.9444**	**0.8155**

Table 2-1-2 Performance of metric learning on the test set.

In the table, Normalized Mutual Information (NMI) is a quantity to measure how much a random variable depends on another. The author used a 0 to 1 NMI border range in this study, with the greater the NMI value, the higher the perfect correlation between the random variables. B-cubed is a precision and recall algorithm for evaluating the test set's performance. In this table, an observation of CLASS-Logistic Regression and CLASS-SVM have the best performance in terms of NMI and B-Cubed shown. In terms of NMI, CLASS-Logistic Regression and CLASS-SVM performed 0.9444 and 0.9425 in the test set, respectively, while CLASS-Logistic Regression and CLASS-SVM performed 0.8155 and 0.8095 in the test set, respectively.

Authors [14] suggested a Deep Reinforcement Learning (DRL) policy to estimate how much energy is consumed when a video is streaming over mobile networks. The goal of this study is to determine the least energy consumption for video transmission without causing video stalling. Figure 2-1-1 depicts the ability to predict and optimize the power of video streaming.

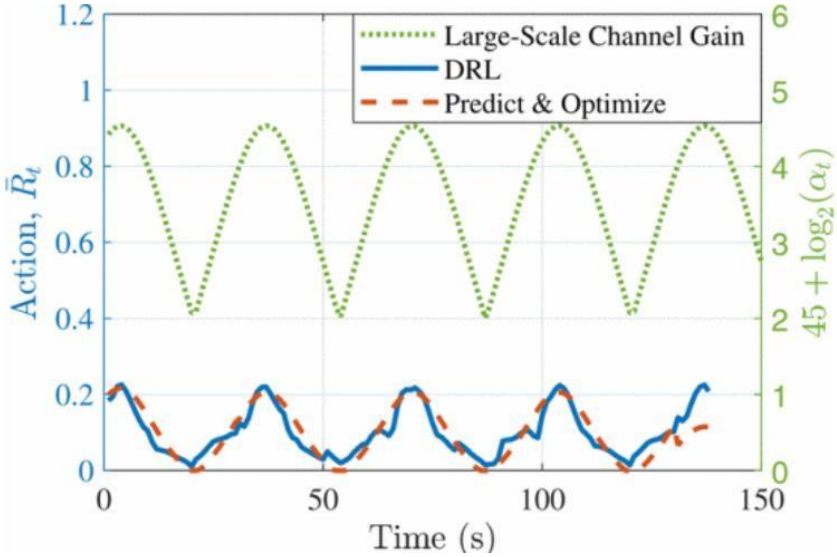


Figure 2-1-1 Behaviour of the DRL-based policy, q=0.5

It is the result of one snapshot from a random episode of a random user in this figure. DRL behaves similarly to "Predict and Optimize," as evidenced by this. The probability terms Q and the learning rate are shown in the diagram. Aside from that, the outcome does not appear to be consistent since each user watches videos at a different pace.

2.1.1 Implementation of Linear Regression in predicting energy consumption

The authors [15] proposed a complete review of LR in machine learning. The purpose of this research is to determine which LR types perform best in terms of prediction and precision in various datasets. The results are shown in the table below after all machine learning approaches have been implemented.

Ref.	Dataset	Technique(s)	Pros and Cons	Results and Accuracy
[54], 2020	simulated Medical Data	Polynomial	The optimization has turned all variables into a smaller set of restricted decimal places and the initial correlation of variables which can be cost-effective for subsequent statistical and computer processing.	82.7%
[11], 2019	ANTLR	SLRM	The well-designed linear approach can be an outstanding tool for interpretation and classification of software code, but the drawbacks are mostly because of complicated internal software constructs which are not enough to be represented solely with code vectors.	90.08%
[14], 2019	Pima Indian Diabetes	MLRM	PCA-LRM achieves higher precision than other approaches.	82.1%
[44], 2018	Aero-Material	MLRM	The very strong fit effect of the MLR can be shown to be smaller than 0.8 per cent for all relative prediction errors such that good prediction outcomes are produced by the MLR prediction.	99.89%
[45], 2018	3D coordinate Medical Data	MLRM	Provides a reliable basis for a detailed study of Chinese bone-setting manipulation and the utility of creation essence in Chinese traditional medicine	X: 92%, Y: 90%, Z: 80%
[47], 2018	Marketing	MLRM	Level of prediction really strong.	84%
[21], 2018	Rice, Wheat and Maize	MLRM	MARS production for rice and wheat is better than MLRM and RFR and MLRM is better than maize RFR n and MARS.	*Rice*: 97%, *Wheat*: 92%, *Maize*: 80%,
[40], 2017	Weather	MLRM	The fit of the model is very similar to 1, which means that the fit of the model is very good and the value of F statistics is very high. There is a strong linear correlation between variables, which satisfies the criteria for MLR.	30 min100%, 1 hour 99%, 3 hours 95%
[49], 2017	Color reading	MLRM	The fits of the regression model is better, the effects of the temperature and refractive index and concentration on the model are not taken into account.	82%
[53], 2017	Historical data set	Polynomial, SLRM	The polynomial curve fit is better than basic linear regression to forecast power generation.	Polynomial 92%, SLR 67%
[37], 2016	Different Medical datasets	MLWNBC	The algorithm blends the simpler linear regression with WNBC's high efficiency. The data attributes are processed to obtain a successful attributes sub-set and reduce the algorithm complexity; on the other hand, the value determines the outcome of the classification, and the weight correlation coefficient is used to increase the classification performance.	82.92%

Table 2-1-1-1 Overview of linear regression techniques

Table 2-1-1-1 has deducted that different LR algorithms will perform better in different situations. For example, in the historical data set, the polynomial regression model has a high accuracy of 92 per cent, compared to the 67 per cent accuracy of the Simple Linear Regression Model (SLRM). This is because the line will behave like a curve in the polynomial technique, making it better suited to forecast power generation. The Multiple Linear Regression Model(MLRM) works well when the Y-variable relates to more than two X variables.

Author [16] has developed a model to forecast the energy consumption of a video based on calculating video frame resolution, frame rate, and bitrate.

The purpose of this research is to determine how much energy is consumed in a video, as well as the error rate and how to reduce it in the dataset. The relationship between power(*W*) and the Discrete Cosine Transform(DCT) coefficient of luminance is shown in Figure 2-1-1-1.

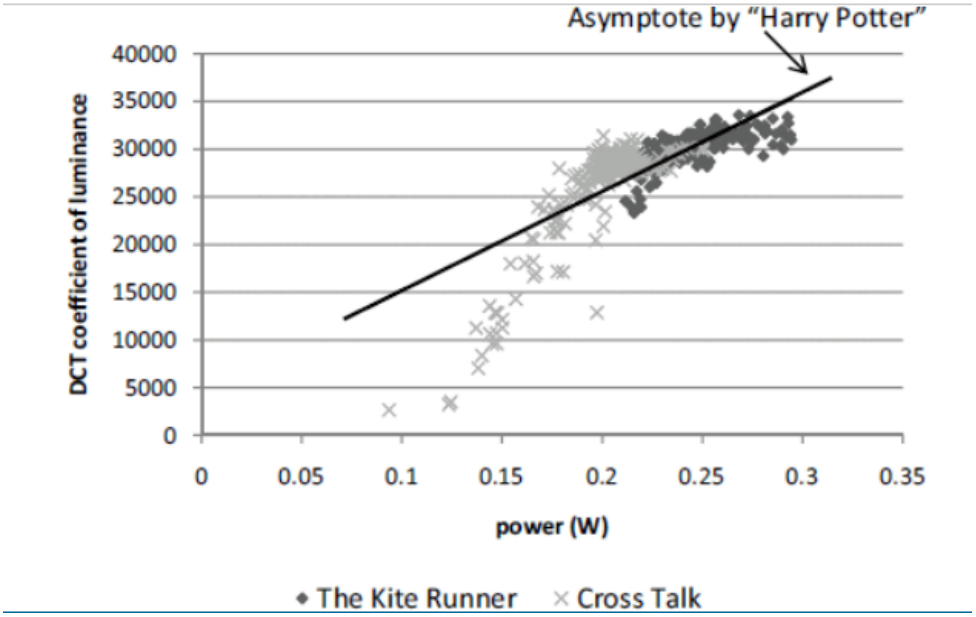


Figure 2-1-1-1 DCT coefficient of luminance and power(W) of videos

In figure 2-1-1-1, it is shown that the line is performing a strong positive correlation with power and the DCT coefficient of luminance. DCT coefficient value represents the bit length of a macroblock.

2.2 Strengths and Weakness

2.2.1 Strength

The first strength of the study [10] is that it thoroughly examines each of the machine learning models used to forecast electrical usage. Based on this study, a mapping for dataset training or machine learning models has been observed. After that, a mapping of how the SVM, MLP, and decision tree learning work in predicting energy consumption has been generated from the study [11].

In study [13]'s strength is that it has generated a map of how similarity learning techniques function in video clustering. It also serves as a starting point for arranging media from real-life events. The strength of the study [14] is it has generated the energy consumption prediction model by using other machine learning techniques. This paper also serves as a reference for using the LR model to estimate the energy consumption pattern.

The strength of the study [15] is that it has developed a table of how to select LR models in various datasets' events. It can help to obtain a sense of how the sample of the final model will look by reviewing this paper and can get a head start on the project doing so since the model to predict energy consumption is the same as the study [16].

2.2.2 Weakness

The first weakness of this study is Neural Network(NN) machine learning is not included in this study. This is because by referencing to study [10], after including NN, it will over-parameterize the data set, resulting in the employment of very large NN models. Errors are easy to cause and are not properly evaluated. Next, traditional clustering is not used for clustering social media events, according to the study [13]. This is because social media platforms are always changing and expanding in size. Thus, it is difficult to cluster such events, traditional clustering is not recommended.

A limitation is also included in the study [14] which stated that it is hard to get a consistent outcome since the users can watch videos at a different pace. To achieve a high level of accuracy, the video's velocity must be kept constant. In the study [15], a study has shown that when compared to other standard AI methodologies, Deep Reinforcement Learning is more difficult and complex to employ, but it can understand higher-level causal relationships [17]. This skill will need more data and time to have a nice performance.

Chapter 3
System Methodology/Approach OR System Model

Proposed Method/Approach

The processes of the project were categorized into different phases in the development, which were project pre-development, data exploration, data pre-processing, model selection and build model, model validation, model fine-tune, model testing, and prediction on the YouTube dataset. In the end, the results were represented on the website.

3.1 System Design Equation

In this project, the system utilised machine learning techniques to estimate the energy consumption pattern.

Here is the format for the scientific equation for the hypothesis function of SLRM:

$y = \theta_0 + \theta_1 x$ (1)

Where

θ_0 is the bias term

θ_1 is the feature weight

x is the input value

y is the target value

By using this formula, the equation of the energy consumed by YouTube can be found.

Here is the format for the scientific equation for the Normal Equation:

$\theta = (X^T X)^{-1} X^T y$ (2)

where

θ is the parameter θ that minimizes the cost function

X is the Input matrix

y is the vector of target values

By using this formula, the best θ can be found and fitted into the model.

Here is the format for the scientific equation for Energy Equation:

$E = V * mAh * 3.6$ (3)

where

E is the energy of the battery in Joules per hour

V is the battery voltage, assume voltage = 1 volts

mAh is the battery capacity

Using this formula helps to calculate how much energy has been consumed by watching a YouTube video.

Here is the format for the scientific equation for calculating Normalized Root Mean Square Error by normalizing using the mean:

$NRMSE = RMSE/\bar{x}$ (4)

where

$NRMSE$ is the normalized root mean square error

$RMSE$ is the root mean square error

$\bar{x}$ is the mean

Using this formula helps to calculate the NRMSE of the data.

3.2 System Design Diagram

3.2.1 System Architecture Diagram

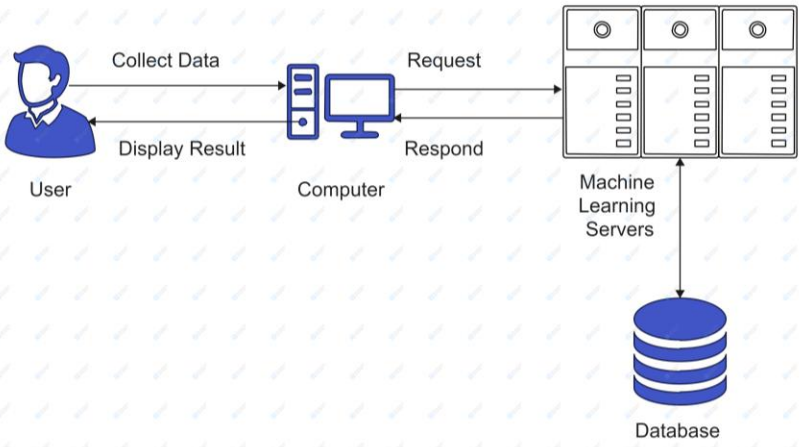


Figure 3-2-1-1 System Architecture Diagram

Figure 3-2-1-1 showed a system architectural diagram of how a user would interact with the website of this project. The computer would first be served as a front end and asked the user to input the data. After the user selected which function to obtain, the computer would then send a request to the backend and let the machine learning servers establish a connection with the database to compute the result and sent it back to the computer. The user was then shown the result once the computer had successfully obtained it.

In the system, a cloud database type would be used. This is because there is no strict location for a user to use the machine for predicting video energy consumption. By using the cloud database, the user was able to access his dataset file whenever and wherever as long as he had an Internet connection.

3.2.2 Use Case Diagram and Description

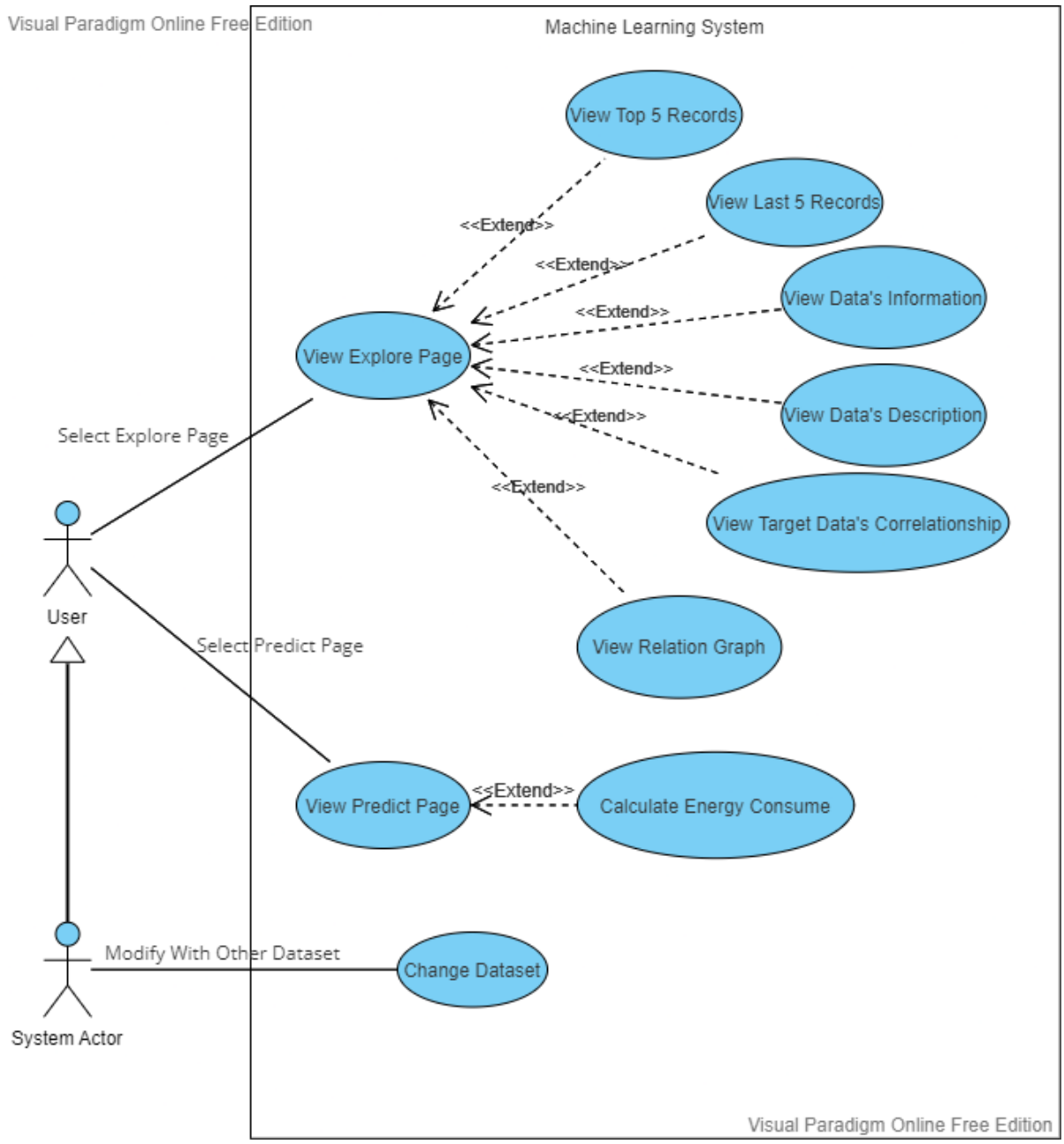


Figure 3-2-2-1 Use Case Diagram

Use Case Legend:

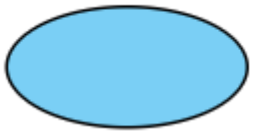
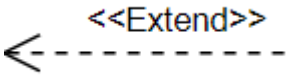
Figure	Name	Explanation
	User	It represents the people who use the web page
	System Actor	It represents the people who moderate the web page system
	Use Case	It represents a system or a process to interact with the people
	Association relationship	It connects actors and uses cases if there exists an interaction between them
	Generalization relationship	It means one actor can inherit the role of another actor.
	Extend relationship	It is an additional use case related to a use case and this use case can select to either carry out or not its functions.

Table 3-2-2-1 Use Case Legend

Use Case Description:

Use Case Name:	Explore Page
Use Case ID:	FR1
Description:	This use case allows user to modify a new dataset and display explore page
Extend Use Case:	1. View top 5 Records 2. View last 5 Records 3. View data's information 4. View data's description 5. View target data's correlation ship 6. View relation graph
Primary Actor:	User
Secondary Actor:	System Actor
Precondition:	1. The model load dataset successfully
Main Flow:	1. The system actor changes the file of the dataset 2. The user selects to display explore page from the side bar 3. The system computes and displays the explore page 4. The use case ends

Table 3-2-2-2 Use Case Description – Explore Page

Use Case Name:	Predict Page
Use Case ID:	FR2
Description:	This use case allows user to modify a new dataset and display predict page
Extend Use Case:	1. Calculate energy consume
Primary Actor:	User
Secondary Actor:	System Actor
Precondition:	1. The model load dataset successfully
Main Flow:	1. The system actor changes the file of the dataset 2. The user selects to display predict page from the side bar 3. The system computes and displays the predict page 4. The use case ends

Table 3-2-2-3 Use Case Description – Predict Page

3.2.3 Activity Diagram

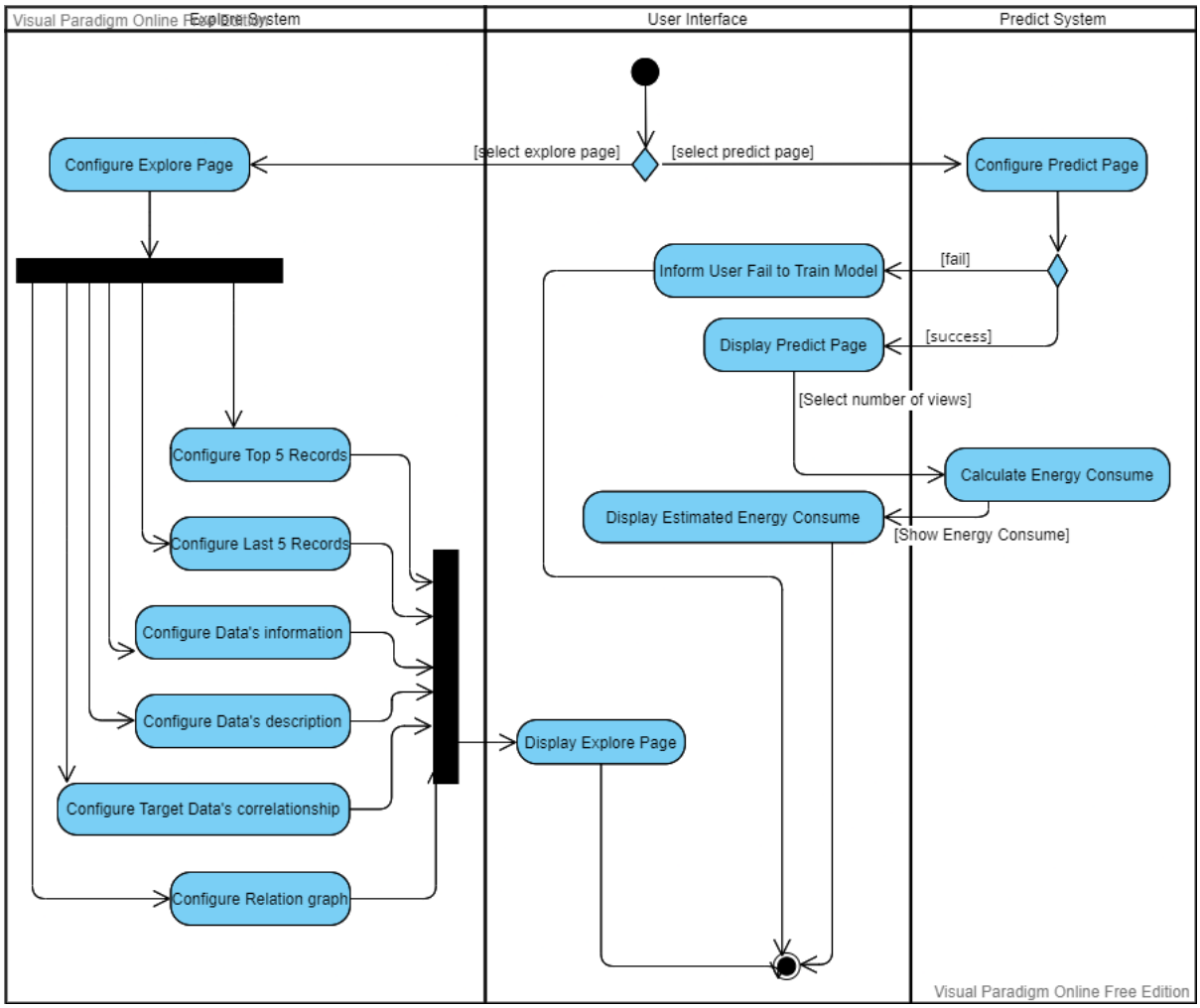


Figure 3-2-3-1 Activity Diagram

Each step in the website's flow was depicted in Figure 3-2-3-1. The user could first pick whether to configure the explore page or the predict page in the user interface. The explore system would divide into 6 sections in preparation for the user to decide which portions to display if the user chose to select the explore page. The outcome would be displayed in the user interface. After that, if the user selected the predict page, the predict system would then take the data from the dataset file and trained with the model. A message would be sent to the user to notify them if the model was unable to train. If succeed, displayed the predict page and the predicted result. From the predict page, the user could select to perform calculations on estimating energy consumption by entering the number of views to it.

Chapter 4
System Design

4.1 System Architecture Diagram

In figure 4-1-1, a flow chart of the machine learning pipeline has been drawn out.

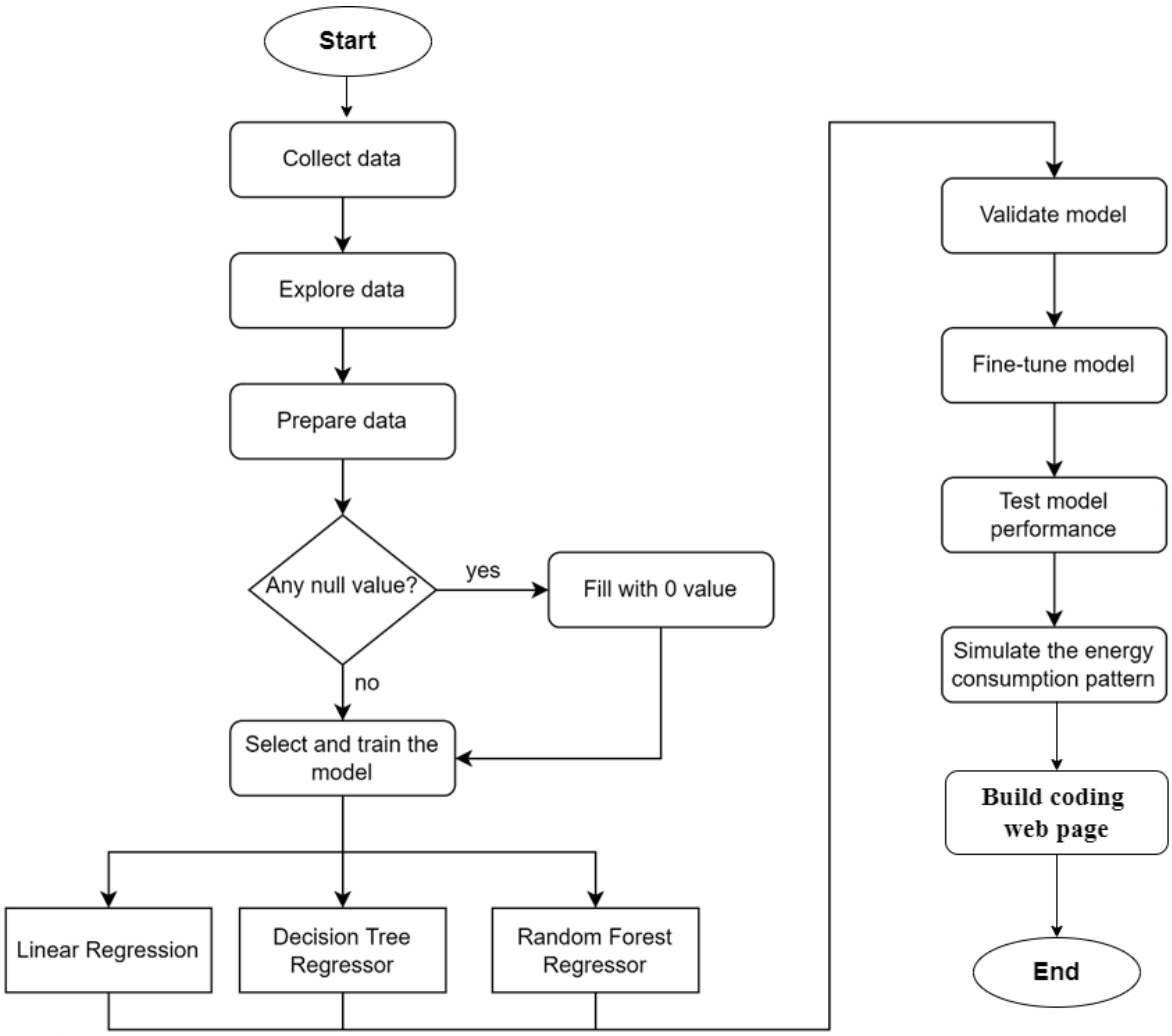


Figure 4-1-1 Flow Chart Diagram

The first step was to collect the data. Accubattery software was used to collect data in this step. The data was gathered when watching a YouTube video on a smartphone. Accubattery Software should be running in the background so that it can compute how much current has been used once the video has ended. It is a tool that can separate every program running on the phone and analyses how much battery it uses. To convert the current value to energy value for the model to anticipate later, the energy equation formula (3) was used. Since this project has

utilised several different model smartphones to estimate the current, an average current spent was also added to get a higher accuracy percentage.

Jupyter notebook software was then used to configure the coding for this project. The machine began to investigate and analyse the data which has been obtained. This was one of the longest steps because it is critical for the machine to fully comprehend the dataset before moving forward. Before the analysing begins, three Python classes, Pandas, Numpy, and Matplotlib. pyplot has been imported. Pandas was a class for analysing high-level data structures, Numpy was a class for performing scientific computing, and Matploylib. pyplot class was used to plot graphs. Several techniques were inserted to let the user have a more perspective view of how the relationship between the data.

It was about coding cleaning and summarizing the data for analysis later in the data preparation process. The data would be more complete in this step by filling in the missing variables and normalising the data. StandardScaler class has been imported from sklearn. preprocessing to perform data standardisation, which means the data would be standardised to have a mean of 0 and a standard deviation of 1. Train_test_split class has been imported from sklearn. model selection to split the origin data into two parts one for training and one for testing the trained set. LabelBinarizer, which was imported from sklearn. preprocessing, was the last class. This class could assist machine learning in performing one hot encode category data which means converting multi-class labels of category data to binary labels for an easier process in coding [18]. Thus, this stage was crucial since machine learning cannot do the operations below without first preprocessing the data.

Following that, the machine was about to start selecting and training the models. In this step, three different models were chosen and trained to see which one performed best in this dataset. Linear Regression, Decision Tree Regressor, and Random Forest Regressor models were used to modify the model. This is because these models would work well in predicting the result. LR worked well in analysing the relationship between the variables, Decision Tree worked well in evaluating the potential outcomes and Random Forest Regressor would create a huge number of decision trees and feed every data into it to observe the final output [19]. These three models were also easy to use.

Each model has its own set of classes that must be imported. These three models were imported from *sklearn.linear_model*, *sklearn_tree*, and *sklearn.ensemble* libraries, respectively. *Sklearn_metrics* was imported to conduct RMSE coding on the models.

After the models have been selected and trained, a perspective view on the trained set was imported for the user to view. First, Mean_squared_error class which was imported from sklearn. metrics were performed to find the RMSE for the models. After this step has been completed, the models have successfully obtained and been analysed.

Next, coding for validating the model on the trained models is required. To execute cross-validation coding, the function *cross_val_score* was imported from *sklearn.model selection*. Cross-validation was a statistical method for estimating the model's skill that was previously trained. It was a machine learning strategy that involves comparing and repeating to obtain the result of how the models worked[20].

Grid search and random search were applied to fine-tune the Random Forest Regressor model in machine learning to find the best model. This process could assist in determining the optimal model in Random Forest Regressor. GridSearchCV and RandomSearchCV classes have been imported from *sklearn.model_selection* to execute the result later. These methods were also used to determine which feature is most important and has highly accurate for the total energy attribute [21]. Thus, the best parameter for the model would be obtained as the result of this step.

A test model performance step came next. This was the process that would initially assess the performance of test data with train data before beginning to mimic the energy consumption pattern.

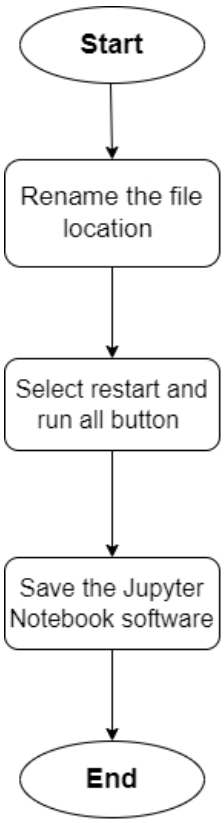
The final phase of the machine learning coding would be to simulate the YouTube dataset's energy consumption pattern. In this case, a simple linear regression class was used. This was because referring to the result obtained in the steps above, it showed that the relationship between total energy attribute and views would perform a linear line. Thus, it was well for this class to be implemented the energy consumption pattern. In this step, machine learning would prepare and show 3 different graphs to the user, which were a reference graph, a comparison theta graph, and a final graph simulated out. In this step, the user could be discussed how the graphs were been plotted out and got to know the

relationship between the target variable and the initial variable. The final step of the coding would be finished here since the user has got the result on the energy consumption pattern and the equation for this energy consumption pattern.

After the coding on predicting energy consumption patterns based on the top trending video YouTube 2021 was done, the result was displayed on a web page to make it easier for non-professional users to estimate the pattern of energy consumption. During the web development phase, the machine learning coding was saved in the GitHub software so that it could subsequently be called out to the streamlit.io software. The main page, the explore page, and the predicted page were the three pages that made up the coding web page. The main page combined the other pages; the explore page allowed users to browse the dataset, and the predicted page allowed them to choose and forecast the energy consumption pattern.

A requirements text was also provided in the GitHub software since the streamlit.io platform was lacking some library functions that were necessary for the application to run. It is a text file that records all the missing libraries in the platform. First, the platform will run through the library names in the requirement text file. Then, the platform will download the library recorded in the text file. In the website configuration, two gif images have also been included to let the website become more interactive and attractive.

4.2 Coding Setup Diagram



4-2-1 Coding Setup Diagram

The setup flowchart for the machine learning system is given in Figure 4-2-1. At the starting point, the Jupyter notebook program was already open and was reading the original dataset path. Thus, for the first step, a rename of the file location to the new file location was necessary. When the file location has changed, click the option labeled "Restart and run all" to restart the system. Save the Jupyter Notebook once the model has been successfully built.

4.3 Website Setup Diagram

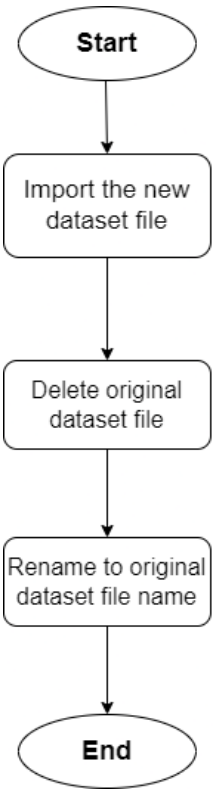


Figure 4-3-1 Website Setup Diagram

The flowchart for setting up the coding for the web page is given in Figure 4-3-1. The GitHub software has opened at the starting point. The new dataset file should first be imported into the GitHub repository. Remove the original dataset file next. Finally, change the name of the new dataset file to the name of the original dataset file. Refresh the streamlit.io website after completing all the procedures, and the result should be successfully constructed upon reading the new dataset file.

Chapter 5
System Implementation

5.1 System Setup

5.1.1 Hardware Setup

This project used a computer as well as numerous different mobile devices as hardware. The data was collected using a Vivo Y20s smartphone and a Pocox3 Pro smartphone. To construct an energy consumption pattern from the YouTube 2021 dataset, an Asus computer was provided. Table 5-1-1-1 describes the details of hardware specification.

Description	Specifications
Model	Asus A456U series
Processor	Intel Core i5-7200U
Operating System	Windows 10
Graphic	NVIDIA GeForce GT 930MX 2GB DDR3
Memory	4GB DDR4 RAM
Storage	1TB SATA HDD

Table 5-1-1-1 Specifications of laptop

5.1.2 Software Setup

Before starting to develop the energy consumption pattern, there were some software needed to be installed and downloaded on the laptop and the phone:

1. Jupyter Notebook (laptop)
2. AccuBattery (phone)
3. GitHub (laptop)
4. Streamlit.io (laptop)

5.2 System and Configuration

At the beginning of the machine learning process, to ensure the dataset would receive an equal energy consumption result from each video, the videos were set to 240 pixels quality, the sound was muted, and it was played with original speed. It was also necessary to remove some data to prevent the dataset from having outliner data. A total of 656 lines of data have been collected.

During the web page development, a notification box was added to the predicted page to show whether the page will work successfully or not. An error message will appear if the dataset file cannot be uploaded to the website. The user's range for entering the number of views to estimate energy consumption was specified to be between 0 and 15,000,000.

5.3 Machine Learning Operation

5.3.1 Data Exploration

This section would discuss the steps of coding machine learning. To explore the data exploration coding and result, please refer to Appendix B and Appendix C respectively. For the coding section of data exploration, first, three Python classes, Pandas, NumPy, and Matplotlib. pyplot has been imported. The dataset file has been read into the coding. the *head* and *tail* commands were for displaying the top 5 and final 5 rows of the data to see if it is entirely loaded in.

The *info* command was then used to understand this data's properties, such as data types and memory. From the result, there was a total of 655 numbers of data, no null numbers existing, and memory usage was 56.5+ KB.

Next, the *describe* command was then used to decipher the statistical information on the attributes of the YouTube dataset. The mean, standard deviation, maximum, minimum, and quartile values of the data were calculated. It reveals that the total energy attribute had a mean of 1e+05, a standard deviation of 1e+05, a minimum value of 4e+01, and a maximum value of 1e+06. 9e+03, 2e+04, and 1e+05 were the first, second, and third quartile values, respectively.

The hist command was then used to identify how this data appear on a graph. The data has been divided into 50 bins and is displayed in a 20x15 grid. In the graph shown, there was no capped data and tail heavy exist.

After that, as for the reference for the coding later, the graph of a relationship between duration with energy attributes, and duration with total energy attributes have been plotted out to show the relationship. In the result, duration has a directly proportional relationship with energy but with total energy, it shows less relationship.

In the YouTube dataset, a correlation matrix was used to determine the connection between the variables, using the duration as the input variable and the energy and total energy as the target variables. A *corr_matrix* command was used to determine which variables have

the highest relationship to the target value, total energy. Views, Duration(h), Energy(J), and Total Energy(kJ) were known as the primary variables to investigate in the dataset. The result has shown each connection between each variable.

Since total energy was the target variable for predicting energy consumption, then subtract its value to see which variable has the highest association. The result has discovered that YouTube's views will have the greatest impact on total energy.

A graph of the relationship between views and total energy attributes has then been plotted out to confirm the result gained above. The result showed that it appears a clearer relationship of linear relationship than by using the duration attribute.

```
youtube.plot(x='Total_Energy(kJ)',y='Views',kind='scatter')
plt.show()
```

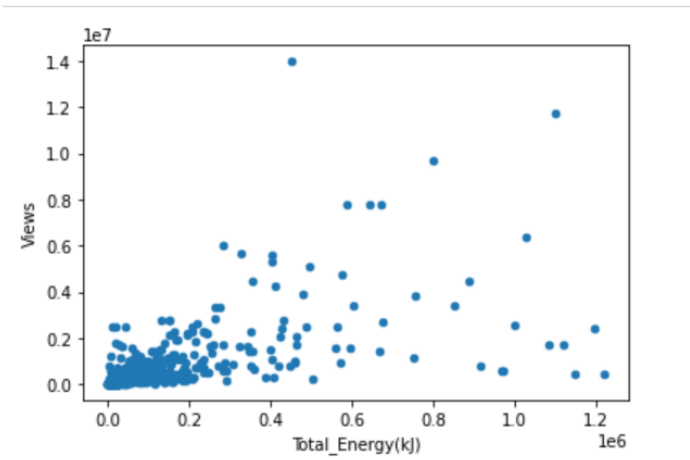


Figure 5-3-1-1 Relationship between views and total energy attributes

5.3.2 Data Preprocessing

To begin, the coding defined X as the input variable and y as the target variable. This is because the total energy column was the result's target variable, a *drop* command was used to remove it from X. Thus, the total energy column would only display on y. Then, since the ID, title, and channel title attributes were not crucial to the dataset, those columns were omitted from the coding. It was only for the user to have a better understanding of the dataset.

Then, several techniques were applied to make the data more appropriate for machine learning. There were about 6 tasks in this step:

1. Split the data into train and test sets, each containing 80 per cent and 20 per cent of the original dataset.
2. Split the train set into numerical and categorical data
3. Filled up the missing value with 0 in the numerical data, if got any
4. Standardized numerical data
5. Transformed the categorical data into the NumPy array
6. Combined numerical and categorical data back into 1 set

As the result, a well-performed dataset was ready to be performed on the coding in the steps later.

```
#2.ML Pipeline - Preprocessing
X = youtube.drop('Total_Energy(kJ)',axis=1)
#ID,title,Channel_title not important
X = X.drop(['ID','Title','Channel_title'],axis=1)
y = youtube['Total_Energy(kJ)']

#split to train and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=30)

#split the matrix into numerical and categorical attributes
X_train_num=X_train.drop(['Country'],axis=1)
X_train_cat=X_train['Country']

#Found missing value
print('Number of missing values for all columns:\n',X_train_num.isnull().sum())
X_train_num=X_train_num.fillna(0)
print('nAfter handling Missing values:\n',X_train_num.isnull().sum())

#Standardize numeric data
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler(copy = False)
scaler.fit(X_train_num)
X_train_num_tr = scaler.transform(X_train_num)

#One-hot encodes categorical data
from sklearn.preprocessing import LabelBinarizer
lb_encoder = LabelBinarizer()
lb_encoder.fit(X_train_cat)
X_train_cat_tr = lb_encoder.transform(X_train_cat) # returns NumPy array

#combine the transformed numerical and categorical sets
X_train_tr = np.hstack([X_train_num_tr,X_train_cat_tr])
y_train = y_train.values; #convert to NumPy array
print('\nX_train_tr shape:',X_train_tr.shape)
print('y_train shape:',y_train.shape)
```

```
Number of missing values for all columns:
 Views                    0
Duration(h)               0
VivoY20s_Usage(mAh)     141
Pocox3pro_Usage(mAh)    382
Avg_Usage(mAh)            0
Energy(J)                 0
dtype: int64

After handling Missing values:
 Views                   0
Duration(h)             0
VivoY20s_Usage(mAh)     0
Pocox3pro_Usage(mAh)    0
Avg_Usage(mAh)          0
Energy(J)               0
dtype: int64

X_train_tr shape: (524, 12)
y_train shape: (524,)
```

Figure 5-3-2-1 Result before and after preprocessing

5.3.3 Model Selection and Building the Model

The Linear Regression model, Decision Tree Regressor model, and Random Forest Regressor models were chosen to suit the dataset during the model selection process. The following are the model-building tasks:

a) Create an instance of Linear Regression, Decision Tree Regressor, or Random Forest Regressor.
b) Fit the input variable X_train_tr and the target variable y train to the model
c) Predict on the training set, a *predict* command was added.
d) Create a function to select 10 random samples from the trained and predicted sets to see how well the actual variable compared to the predicted variable.
e) Using trained regression, make predictions on the training set.
f) Show the RMSE on the training set

As a consequence of the analysis, it was determined that there was a significant difference between actual and projected values. Other than that, RMSE and mean values performed from the model have also been calculated. the Linear Regression model's RMSE was 130899.69. Given that the mean total energy in the YouTube dataset was 101927.86. After calculating NRMSE by using equation (4), a result of 1.28 indicated that the model was underfitting [22].

```
#3.ML Pipeline - Model selection and building the model
#Linear Regression model
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

# create an instance of LinearRegression
lin_reg = LinearRegression()

# Train the regression model
lin_reg.fit(X_train_tr, y_train)

# Predict the training set using trained regression
y_pred =  lin_reg.predict(X_train_tr)

# Function to choose 10 random samples
def show10results(y_train,y_pred):
    print('Result for the 10 random samples:')
    selected = np.random.randint(0, len(y_train), 10)
    for i in selected:
        print('actual = {:7.0f} pred ={:7.0f}'.format(y_train[i], y_pred[i]))

show10results(y_train, y_pred)
# Compute and show the RMSE on the training set
lin_mse = mean_squared_error(y_pred, y_train)
lin_rmse = np.sqrt(lin_mse)
print ('RMSE =', lin_rmse) #Underfitting
print ('mean=',youtube['Total_Energy(kJ)'].mean ())
```

```
Result for the 10 random samples:
actual =  150177 pred = 221259
actual =  136069 pred = 101391
actual =   64278 pred =  84250
actual =   22728 pred =  25414
actual =    5084 pred =  18164
actual =   14647 pred =  23403
actual =  561159 pred = 284924
actual =   19806 pred =  86696
actual =   53344 pred =  48425
actual =    5084 pred =  18164
RMSE = 130899.68802555118
mean= 101927.86280487805
```

Figure 5-3-3-1 Linear Regression model performance result

The coding for the Decision Tree Regressor and Random Forest Regressor was similar to the Linear Regression model. Same actual and predicted values and an RMSE of 0.0 were obtained as the result of the Decision Tree Regressor, which was impossible because the RMSE was too good. This indicated that the model was overfitting when it was being trained.

```
from sklearn .tree import DecisionTreeRegressor

# create an instance of DecisionTreeRegressor
tree_reg = DecisionTreeRegressor(random_state=42)

# Train the regressor model
tree_reg.fit (X_train_tr, y_train)

# Predict the training set using trained regressor
y_pred =  tree_reg.predict(X_train_tr)

# Show the result of 10 random samples
show10results(y_train, y_pred)

#Compute and show the RMSE on the training  set
tree_mse = mean_squared_error(y_pred, y_train)
tree_rmse = np.sqrt(tree_mse)
```

print('RMSE =', tree_rmse) #Overfitting

```
Result for the 10 random samples:
actual =   50181 pred =  50181
actual =    1917 pred =   1917
actual =   34558 pred =  34558
actual =   38979 pred =  38979
actual =  212770 pred = 212770
actual =    5255 pred =   5255
actual =    3576 pred =   3576
actual =   17828 pred =  17828
actual =  387255 pred = 387255
actual =   83483 pred =  83483
RMSE = 0.0
```

Figure 5-3-3-2 Decision Tree Regressor model performance result

Last but not least, for the instance created for the Random Forest Regressor, 10 numbers of trees were built for the model. It obtained a good RMSE value of 29782. By comparing to two other models, the Random Forest Regressor would do well on the dataset.

```
from sklearn.ensemble import RandomForestRegressor

forest_reg = RandomForestRegressor(n_estimators=10, random_state=42)
forest_reg.fit (X_train_tr, y_train)

# Predict on training set
y_pred = forest_reg.predict(X_train_tr)

#Show result of 10 random samples
show10results(y_pred, y_train)

# Show the RMSE
forest_mse = mean_squared_error(y_pred, y_train)
forest_rmse = np.sqrt(forest_mse)
print('RMSE =', forest_rmse)#Forest Regressor will be fine on predicting model
```

```
Result for the 10 random samples:
actual =    3544 pred =   3576
actual =   27005 pred =  26270
actual =    5556 pred =   5387
actual =   67306 pred =  83035
actual =    3142 pred =   2975
actual =  136060 pred = 144860
actual =   21118 pred =  20767
actual =   73845 pred =  70278
actual =   19566 pred =  18306
actual =  369289 pred = 284585
RMSE = 29781.99574261262
```

Figure 5-3-3-3 Random Forest Regressor model performance result

5.3.4 Model Validation

Next, coding for validating the model on the trained models was required. In this step, 3-fold cross-validation has been used to validate the model. To perform 3-fold cross-validation, the original training data would be split into three folds in each iteration. The model's performance would be evaluated using one of the folds as a validation set, while the other two folds would be used as training sets. This procedure would be repeated until each fold has in turned as the validation set. After they have been completed, a more accurate performance has displayed. To double-check, called out the command of scores, mean, and standard deviation.

The values after the models were validated were listed in the table below.

	Linear Regression	Decision Tree Regressor	Random Forest Regressor
Mean, $\bar{x}$	142135.99	81916.92	67072.52
Standard deviation, σ	24032.39	10745.92	18543.39

Table 5-3-4-1 Validation value of several machine learning models

Table 5-3-4-1 showed that the Random Forest Regressor has a mean of 67072.52, which was significantly higher than the RMSE of 29782 observed earlier. It was discovered that the Random Forest Regressor has an overfitting problem as a result of this stage [23].

```
#4.ML Pipeline - Model Validation
#linear Regression Model

from sklearn.model_selection import cross_val_score

#cross validation - 3-fold
k_fold_scores = cross_val_score(lin_reg, X_train_tr, y_train, scoring =
'neg_mean_squared_error', cv=3)
linreg_rmse_scores = np.sqrt(-k_fold_scores)

#function to display scores,mean & standard deviation
def display_scores(scores):
    print('Scores:', scores,'\n')
    print('Mean:', scores.mean())
    print('Standard deviation:', scores.std())

display_scores(linreg_rmse_scores)
```

```
Scores: [168255.53453134 110243.77439102 147908.66697043]

Mean: 142135.99196426306
Standard deviation: 24032.39357186508
```

Figure 5-3-4-1 Linear Regression model validation result

```
k_fold_scores = cross_val_score(tree_reg, X_train_tr, y_train, scoring =
'neg_mean_squared_error', cv=3)

tree_rmse_scores = np.sqrt(-k_fold_scores)

display_scores(tree_rmse_scores)
```

```
Scores: [74103.84119491 97111.90907375 74535.00513987]

Mean: 81916.91846951107
Standard deviation: 10745.922643312222
```

Figure 5-3-4-2 Decision Tree Regressor model validation result

```
k_fold_scores = cross_val_score(forest_reg, X_train_tr, y_train, scoring = 'neg_mean_squared_error', cv=3)

forest_rmse_scores = np.sqrt(-k_fold_scores)

display_scores(forest_rmse_scores)
```

```
Scores: [92733.11547213 58925.81634755 49558.61730272]

Mean: 67072.51637413545
Standard deviation: 18543.38811252023
```

Figure 5-3-4-3 Random Forest Regressor model validation result

5.3.5 Fine-tune the Model

When running the grid search, the coding has specified that the sampling data can be replaced after it has been drawn, [12,20,28,30,38] was the number of trees in the forest, and [4,6,8] was the number of features used to split the node. Then, print out the best model, best hyperparameter values, and the RMSE of the best model to observe the changes after the grid search has been completed.

It was discovered that the Random Forest Regressor model performs best when features = 6, tree = 20, and random state=42 were used. A result RMSE of 65134.16 has resulted from it. A random state was used to ensure that the problem was reproduced in the same way each time the coding was executed. Grid search can also be used to determine which feature was most important for the total energy attribute. After performing a grid search, the views attribute was obtained to be the most important predictor, followed by duration, energy, and average usage.

#5.ML Pipeline - Fine-tune the model using Grid Search

```
from sklearn.model_selection import GridSearchCV

# create an instance of RandomForestRegressor(random_stote=42)
forest_reg=RandomForestRegressor(random_state=42)

# define param_grid
# When sampling data, the data drawn is replaced
#5 number of trees in forest
#3 features consider for splitting the node
param_grid = {'bootstrap':[True,False],'n_estimators':[12,20,28,30,38],'max_features':[4,6,8]}

#create an instance of GridSearchCV using param_grid with 3·fold cv
grid_search=GridSearchCV(forest_reg,param_grid,cv=3,scoring='neg_mean_squared_error')

print ('Performing grid search...', end ='')
#perform grid search
grid_search.fit(X_train_tr, y_train)
print('done')
best_model = grid_search.best_estimator_
print('The best model is:',best_model)
print('Best hyperparameter settings:',grid_search.best_params_)
print('RMSE of the best model =',np.sqrt(-grid_search.best_score_))
# RMSE - need more fine-tuning

feature_importances = grid_search.best_estimator_.feature_importances_
attributes = X_train_num.columns.tolist() + list(lb_encoder.classes_)
sorted(zip(feature_importances,attributes),reverse=True)
#Views is the most important predictor, followed by duration, energy, and average usage
```

```
The best model is: RandomForestRegressor(max_features=8, n_estimators=20, random_state=42)
Best hyperparameter settings: {'bootstrap': True, 'max_features': 8, 'n_estimators': 20}
RMSE of the best model = 65134.16384936281
```

```
[(0.5540287170132199, 'Views'),
 (0.14710352827530254, 'Duration(h)'),
 (0.13715654800477647, 'Energy(J)'),
 (0.0983446984193779, 'Avg_Usage(mAh)'),
 (0.04747218313422934, 'VivoY20s_Usage(mAh)'),
 (0.007126324592125389, 'CA'),
 (0.0038410434130170352, 'Pocox3pro_Usage(mAh)'),
 (0.002015487891274415, 'DE'),
 (0.001487209078674411, 'FR'),
 (0.0013038309948409134, 'KR'),
 (0.00010379569406797745, 'BR'),
 (1.6633489093817758e-05, 'MY')]
```

Figure 5-3-5-1 Best grid search model and feature

In addition to grid search, the random search method was also used to determine whether the performance would be the same if a random combination of hyperparameters were used [24]. The random search method's coding was similar to the grid search. The performance was discovered to be identical to the grid search method when several coding terms were changed from "grid search" to "random search." Thus, it was completely guaranteed that the views attribute would be the key component in predicting total energy consumption.

```
#Fine-tune the model using Random Search
#Random combination of hyperparameters
from sklearn.model_selection import RandomizedSearchCV
random_search=RandomizedSearchCV(forest_reg,param_grid,cv=3,scoring='neg_mean_squ
ared_error')

print ('Performing random search...', end ='')
#perform a random search
random_search.fit(X_train_tr, y_train)
print('done')
best_model = random_search.best_estimator_
print('The best model is:',best_model)
print('Best hyperparameter settings:',random_search.best_params_)
print('RMSE of the best model =',np.sqrt(-random_search.best_score_))
```

```
feature_importances = random_search.best_estimator_.feature_importances_
attributes = X_train_num.columns.tolist() + list(lb_encoder.classes_)
sorted(zip(feature_importances,attributes),reverse=True)
#Views is the most important predictor, followed by duration, energy, and average usage
```

```
The best model is: RandomForestRegressor(max_features=8, n_estimators=20, random_state=42)
Best hyperparameter settings: {'n_estimators': 20, 'max_features': 8, 'bootstrap': True}
RMSE of the best model = 65134.16384936281

[(0.5540287170132199, 'Views'),
 (0.14710352827530254, 'Duration(h)'),
 (0.13715654800477647, 'Energy(J)'),
 (0.0983446984193779, 'Avg_Usage(mAh)'),
 (0.04747218313422934, 'VivoY20s_Usage(mAh)'),
 (0.007126324592125389, 'CA'),
 (0.0038410434130170352, 'Pocox3pro_Usage(mAh)'),
 (0.002015487891274415, 'DE'),
 (0.001487209078674411, 'FR'),
 (0.0013038309948409134, 'KR'),
 (0.00010379569406797745, 'BR'),
 (1.6633489093817758e-05, 'MY')]
```

Figure 5-3-5-2 Best random search model and feature

5.3.6 Test Performance

In this step, the coding will be similar to the data preparation step. The coding word 'train' was replaced with 'test' and the RMSE of the model has also been calculated to make a comparison between both train and test RMSE.

The result from test performance has been determined after the coding has been completed for the train set. After calculating the RMSE of the model, it was discovered that the RMSE of the test model, 42046.95, was significantly lower than the RMSE of the train set after grid-search. The model is still underfitting [23]. This suggested that the train data will require further fine-tuning in future projects.

```
#6.ML Pipeline - Test Performance
# SpLit the test set into numerical and categoricaL sets
X_test_num = X_test.drop('Country',axis = 1)
X_test_cat = X_test['Country']
```

```
# Handle missing value
print('Number of missing values for all columns:\n',X_test_num.isnull().sum())
X_test_num_tr=X_test_num.fillna(0)
print('\nAfter handling Missing values:\n',X_test_num_tr.isnull().sum())

# Perform standardize
X_test_num_tr = scaler.transform(X_test_num_tr)

# Convert categorical attribute into one-hot vector
X_test_cat_tr = lb_encoder.transform(X_test_cat)

# combine the numerical and categorical sets
X_test_cat_tr = np.hstack([X_test_num_tr,X_test_cat_tr])

# Evaluate the best model on the test set and compute the RMSE
y_pred = best_model.predict(X_test_cat_tr)
final_mse = mean_squared_error(y_test,y_pred)
final_rmse = np.sqrt(final_mse)
final_rmse
#lower than train rmse, train rmse need more fine-tuning
```

```
Number of missing values for all columns:
 Views                    0
Duration(h)              0
VivoY20s_Usage(mAh)     34
Pocox3pro_Usage(mAh)    97
Avg_Usage(mAh)           0
Energy(J)                0
dtype: int64

After handling Missing values:
 Views                    0
Duration(h)              0
VivoY20s_Usage(mAh)      0
Pocox3pro_Usage(mAh)     0
Avg_Usage(mAh)           0
Energy(J)                0
dtype: int64

42046.948970740654
```

Figure 5-3-6-1 Result of handling the missing value and RMSE

5.3.7 Simulation using Python - predicting energy consumption

Finally, it was the result of simulating the energy consumption pattern step. In this case, LR was used. The views attribute was chosen as an input variable (x) and the total energy attribute as a target variable (y). First, the coding used these two variables to train the linear model on the dataset and compute the prediction error. The RMSE of the model was calculated as a value of 146498.28. A graph with the samples and a LR line as the starting point was created.

```
#7.ML Pipeline -Simulation using Python - predicting energy consumption
# Algorithm the MyLinearRegression class is selected
X = youtube[['Views']].values # extract input X as 2D NumPy array (m,1)
y = youtube ['Total_Energy(kJ)'].values # extract output y as rank-1 NumPy array (m,)

# Train the Linear model on the dataset and compute the prediction error
# Train linreg using X, y
lin_reg.fit(X,y)

# Predict on X with trained linreg and show RMSE
h_pred=lin_reg.predict(X)
print("Linear model (RMSE):",np.sqrt(mean_squared_error(y,h_pred)))
```

The coding for plotting the graph needed to be defined before being plotted. The X range was defined as 0 to 15,000,000, and the Y range was defined as 0 to 1,500,000. The graph has an 8x6 grid. Dots with a size of 12 and a colour of 'b.' were plotted for the samples. The text size on both the X and Y labels was 18. Legend would be placed in the ideal position computed by the system.

The graph also drew a LR line in addition to visualising the samples. First, the *linspace* command was used to chop out 100 pieces of numbers from the range of 0 to 15,000,000, then reshape them into -1 to 1. After that, start making a prediction and began plotting the graph. This graph would be used as a reference for further coding in the defined MyLinearRegression class.

```
def show_samples (X, y, axis=[0,15000000,0,1500000]):
    fig = plt.figure(figsize=(8, 6))
    plt.plot(X, y, 'b.', markersize=12,label='samples')
    plt.xlabel ('$Views$', fontsize=18)
    plt.ylabel('$TotalEnergy(kJ)$', fontsize=18)
    plt.axis(axis)
    plt.legend(loc='best')

# Visualize the result j
Xtest =  np.linspace(0, 15000000,100).reshape(-1,1)
ypred = lin_reg.predict(Xtest)
show_samples(X, y)
plt.plot (Xtest, ypred, 'r-', label = 'model (LinearRegression) ')
plt.legend()
plt.show()
```

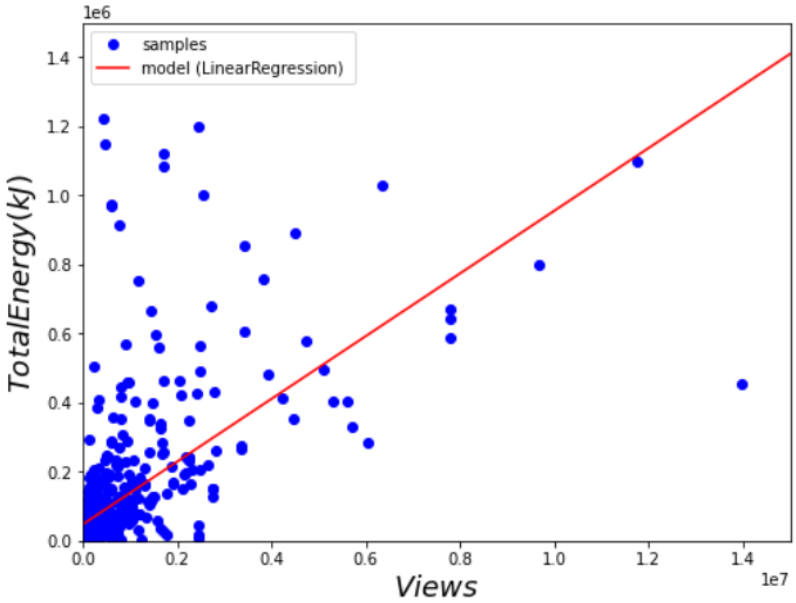


Figure 5-3-7-1 Energy consumption pattern on Linear Regression model

A result observed from Figure 5-3-7-1 was that it was a weak positive relationship. This diagram is not yet complete; it is merely meant to serve as a guide for future machine learning.

There are three functions defined in the MyLinearRegression class: one to initialise the MyLinearRegression, one to train the model with the best theta, and one to predict values for all samples.

```
class MyLinearRegression :

    def __init__(self, theta = None):
        self.theta = np.array(theta)

    def fit(self, X, y):
        m, n = X.shape
        Xo= np.hstack((np.ones((m,1)),X))
        self.theta =np.linalg.pinv(Xo.T.dot(Xo)).dot(Xo.T.dot(y))

    def predict(self, X):
        m,n=X.shape
        Xo=np.hstack((np.ones((m,1)),X))
        h=Xo.dot(self.theta)
        return h
```

A cost function step was included to demonstrate how well the model performs in comparison to other models with various theta. Here, the normal equation (1) and the hypothesis function (2) were used to find out the equation of the model. MyLinearRegression with 2 different assumption theta was plotted on a graph. Each model's RMSE has also been examined and debated.

```
#Cost function - Evaluate  how good is your model compared with another theta
show_samples(X, y)
```

```
#good model
#2 different assumption theta
model1 = MyLinearRegression(theta = [100000, 0.09])
h1 = model1.predict(X)
plt.plot(X, h1, 'r-', label = 'model1 (theta = [100000,0.09])')

#bad  model
model2 = MyLinearRegression(theta = [4,0.3])
h2 = model2.predict(X)
plt.plot(X, h2, 'y-', label = 'model2 (theta = [4, 0.3])')

plt.legend()
plt.show()
```

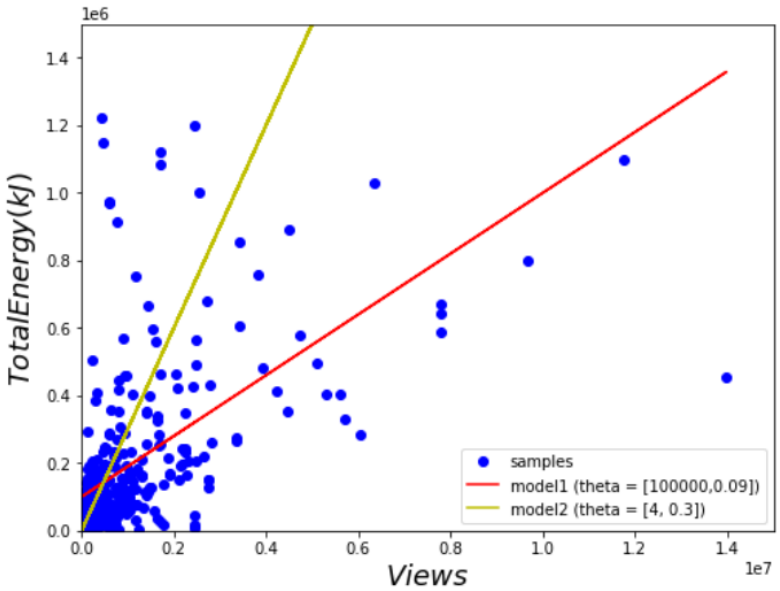


Figure 5-3-7-2 Comparison of models' performance

The outcome of the comparison of using good theta and bad theta was presented in Figure 5-3-7-2. Model 1 was the excellent theta model, while model 2 was the random theta model. The outcome was that model 1 performed better than model 2, with an RMSE of 155332 versus 312333, respectively. Thus, from the result, the theta of the energy prediction was predicted to be similar to model 1 compared to model 2.

```
def RMSE(y_true, y_pred):
    rmse=np.sqrt(np.mean((y_true-y_pred)**2))
    return rmse

print('RMSE for model1 = ',RMSE(y,h1))
print('RMSE for model2 = ',RMSE(y,h2))
#Model 1 fits data than model 2
```

The final step would be to select the optimal line for MyLinearRegression. The coding was similar to the Linear Regression method. It just needed to convert the word 'LinearRegression' to the word 'MyLinearRegression'. *fit* method for fitting the theta and a *predict* method for determining the RMSE.

```
# Implement the training module
# Find optimal
mylinreg = MyLinearRegression()
mylinreg.fit(X, y)
print('Fitted theta:',mylinreg.theta)

# Show the learned model
h_pred = mylinreg.predict(X)
print('Training error (RMSE):', RMSE(y, h_pred))

# Visualize the result
Xtest = np.linspace(0, 15000000, 100).reshape(-1, 1)
h_pred = mylinreg.predict(Xtest)
show_samples(X, y)
plt.plot (Xtest, h_pred, 'r-', label = 'model (MyLinearRegression)')
plt.legend()
plt.show()
```

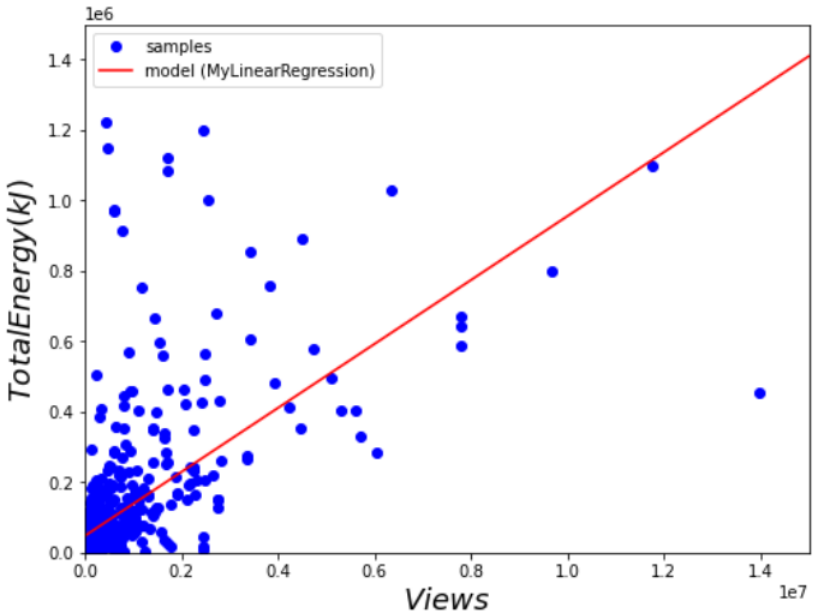


Figure 5-3-7-3 Final energy consumption pattern

As shown in Figure 5-3-7-3, the final energy consumption pattern was shown using the MyLinearRegression class. In MyLinearRegression class, the fitted theta and training error should be the same as Linear Regression. [4.7907e+04, 9e-2] was the fitted theta, and the training error was 146498.28.

5.4 Website Operation

To explore the web page coding and simulation, please refer to Appendix D and Appendix E respectively.

5.4.1 Main Page

This section would discuss the steps of developing the website from the coding of machine learning. First, the main page of the website system has been created. Two specialized pages, *show_explore_page*, and *show_predict_page*, with the Streamlit library, have been imported. To serve as a navigation to other pages, a sidebar has been included. The selection box in the sidebar allowed the user to choose which page to access. Below the selection box, a gif image named "bongo-cat.gif" has been defined.

5.4.2 Explore Page

On this page, Streamlit, Numpy, Pandas, Io, PIL, Matplotlib.pyplot, and Plotly.figure_factory libraries have been imported into the page. First, a CSS style has been defined to hide the first row of the table as in the dataset it has already defined the table head. Next, the coding for explore page has started to define. The title and the description have been written to describe the page function. Then, a resized image has been called. PIL library was used to resize the image.

To explore the data, the YouTube dataset has been read by the system. There were six different functions for exploring the data. Radio technique has been inserted to group the functions. Then, call out the defined table CSS style in the next row with the markdown function. If and else function was then combined with the radio technique in calling the functions thus each time will only appear one way of exploring data. The coding in the if-else function was like the coding in section 5.3.1.

When calling the data’s information, it would appear to error since the output of the function was a standard file object. Io.String.io was used to save the information for the memory file-like object. Other than that, one more existing different coding with section 5.3.1 was in the exploration of the plot relation graph. Since no function can plot out the LR

graph in the Streamlit.io software, *st.plotly_chart()* has been used to create unique chart types. Show_explore_page was called in the end.

5.4.3 Predict Page

On this page, Streamlit, Numpy, Pandas, Matplotlib.pyplot, and Plotly.figure_factory, LinearRegression, and mean_squared_error library have been imported into the page. The coding on this page was like the coding in section 5.3.7. First, create the MyLinearRegression class, *RMSE* function, and *load_data()* function which were the same as section 5.3.7. *st.cache* function was used to cache the data so that whenever the page is refreshed, the system does not need to execute to read data. Then, a calculation has been defined to predict the value of estimated energy consumption when the number of views has been input.

Next, the coding for predict page has started to define. Same as the explore page, a title and description have been written to describe the page function. Then, for the coding to call out the fitted theta, rmse and the graph of the MyLinearRegression were all same as the coding in section 5.3.7. In the prediction, a function *st.success()* has been included to inform whether the prediction of the model was succeeded or not. As same as the explore page, the *st.plotly_chart()* was also used to create the unique chart type.

In the end, a slider was inserted to give the function of choosing how many numbers of views to enter. After a new number has been selected, it will then call the calculation function and show the new estimated energy consumption. Show_predict_page was called in the end.

5.5 Implementation Issues and Challenges

The issue faced was with the implementation of the dataset gathering stage. First, some of the videos were too long to watch, such as the 1.2-hour-long video 'Best Halloween Songs 2018 || Halloween Party Music Mix || The Best Halloween Party Playlist 2018'. The phone must play for 1.2 hours to compute the current spent, which was a waste of time. This was because the AccuBattery software must calculate the video start from the beginning to the conclusion. Then, in some cases, the same video may be broadcast twice in various countries. The dataset may have two copies of the same data. Lastly, the video may have already been removed. This was because the dataset was used from 2019, and some videos have been removed or restricted by nations. Manually removing the video or changing the number of views on YouTube videos was required.

The implementation of the model development was the next challenge that arises. At this point, the result has a large RMSE, indicating that the model has not yet been fitted to the dataset. This was because outliers will influence the accuracy of the LR method. As a result, manually removing outlier videos were required for the model to match the dataset.

5.6 Concluding Remark

In the nutshell, it was critical to create a setup before machine learning begins to train. This was so that casual errors in system implementation could be prevented by a properly established setup. A concealed error might also be discovered. Also covered in this chapter was how to set up the hardware and software. To make it easy for other researchers to track or infer the development. The coding of the machine learning process had also been described here. Finally, the implementation issues and challenges might serve as a warning to other researchers to stay away from making the same error.

Chapter 6
System Evaluation and Discussion

6.1Results on Energy Consumption Pattern

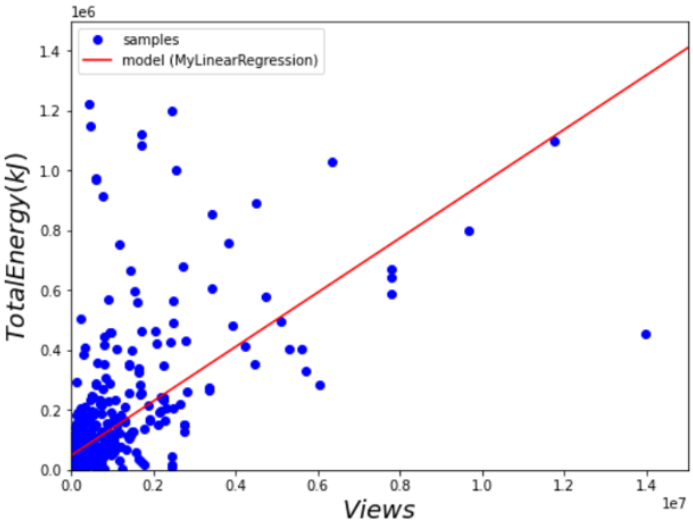


Figure 6-1-1 YouTube 2021 Energy Consumption Pattern

According to the previous finding, it gained that [4.7907e+04 9e-2] was the fitted theta, and the training error was 146498.28. By using the fitted theta as the parameter in equation (1), the equation for total energy consumption based on views was obtained. $y = 0.09x + 47907$ was the equation for Figure 6-1-1, which stated that each unit of total energy consumed was equal to 0.09 times the number of views times 47907 units.

The result of the energy consumption pattern has shown that the line was depicting a weak positive correlation form. The association between the views property and total energy attribute has responded, but it is not a strong one. Because the samples did not follow a line and adhere to it, the graph's absolute value was low. Training error would therefore be considerable. An observation in this figure demonstrates how the range of views will vary from one another. This is because many samples in the graph were converging at low views. The high number of views was lacking, hence extra training data that was added should have concentrated on the high number of views.

6.2 Project Challenges

Lack of experience was the main project obstacle overall. This was because before starting this project, there were few opportunities to get in touch with machine learning coding. There was only one subject in the course study dedicated to teaching artificial intelligence. As a result, conducting extensive research and investigations was necessary to solve this issue.

Aside from that, the project was lacking in human resources. This is because the mobile device must play the video 1 by 1 and record each video's energy consumption to obtain the data. Since this was a solo project, the dataset had fewer records. Thus, A helping hand from others has helped to overcome this obstacle. Different types of models for collecting video energy have increased the accuracy of each video energy consumption because not all people have the same mobile device model. This was accomplished by calculating and obtaining an average.

6.3 Objective Evaluation

Index	Objectives	States
1.	Collect and cluster the data set from YouTube.	Completed
2.	Work out the data set using Linear Regression.	Completed
3..	Determine the accuracy of the predicted energy consumption pattern.	Not good enough
4	Created an energy consumption pattern web page	Completed

Table 6-3-1 Objective Evaluation

Table 6-3-1 has shown the overall objective evaluation of this project. The dataset was collected and clustered successfully from YouTube. An energy consumption pattern has been effectively anticipated at the project's conclusion. If enough samples were uploaded, it should be possible to estimate the future YouTube energy usage pattern. By entering the chosen number of views, the general public could also forecast the overall amount of energy used on a video. Since this effort merely developed a prediction model for LR, it is also efficient for other researchers to conduct additional research.

Other than that, the design of the website's coding makes it simple for non-professional people to use. No technical terms were used, and several images were added to make the website more appealing and intriguing. The webpage does not contain any complex commands, and a sidebar allows users to switch at any time to another page. It is efficient and effective because *st. cache* command is added, preventing the model from loading the data each time the web page is refreshed.

After examining the entire project, it was determined that the training error was significant, the dataset had fewer records, and there were lots of outliers' data to be removed. Thus, the project needs to include more example data.

6.4 Concluding Remark

In summary, this chapter has discussed and analyzed the result of the final energy consumption pattern. It is crucial to analyze the outcome since occasionally, it may fall short of expectations. The end outcome may reveal whether there were any mistakes made during the implementation stage. The challenges of the project have then been discussed. When citing this study, other researchers might steer clear of making the same errors. Last but not least, an objective evaluation has been developed to demonstrate the project's progress and enable a fresh look at it.

Chapter 7
Conclusion and Recommendation

7.1 Conclusion

In a nutshell, the machine learning pipeline for estimating energy usage in 2021 based on the most popular YouTube videos has been developed successfully. Data collection, data exploration, data pre-processing, model selection and construction, model validation, model fine-tuning, model test performance, and finally, simulating the energy consumption pattern were all covered. The dataset became more intelligible, examined, and error-free as each phase of the machine learning pipeline was completed.

After all the steps have been completed, a result on estimating energy consumption has been gained. It discovered the equation for total energy consumption based on views was $y = 0.09x + 47907$ from the ideal model, which meant that every 1 total energy used was equivalent to 0.09 times views plus 47907. The next achievement was the creation of a successful graph using the MyLinearRegression class.

The coding implementation for the website was also successfully created. Numerous methods of data investigation were functional on the explore page. In addition, the predicted page displayed the outcome of the data prediction and the energy consumption pattern. The equation was implemented and utilized to forecast the energy consumed when several views were inserted.

As a result, it is critical to always remain aware of the surroundings. This is because there are many strategies to reduce carbon dioxide emissions, such as closing YouTube videos while individuals are not watching them. Although these efforts have a minor impact on the problem, a small shift from everyone can make life better. In conclusion, learning how to analyse an energy consumption pattern and forecast future energy consumption for a company or an individual is critical.

7.2 Recommendation

It is advised to select a less noisy dataset for this project because the LR method can get distracted easily by the noisy data. As the author [25] stated, LR is too sensitive to outliers compared to others learning techniques. After that, using more data for training is advised so that the machine may train on a larger volume of data and experience fewer overfitting problems.

For future projects, more improvements to the model and letting it produce a higher accuracy energy consumption pattern will be included. Next will fine-tune the model or add more collected data to this dataset to let the model become fit to the dataset because the RMSE observed from the data was still high. Techniques like Batch Gradient Descent, BatchNormalization for standardising the inputs, and self-paced fine-tuning network are also recommended to be analysed and discussed onwards[26], [27]. Aside from that, more reviews will be found and incorporated into this project to make it appear more professional and credible to the general public.

By reference to this research, the researcher can do more studies based on the use of LR based on this project. Further than that, the researcher can also use different machine learning languages to forecast an energy consumption trend based on videos.

REFERENCES

[1] IEA, “The carbon footprint of streaming video: fact-checking the headlines, IEA, Paris,” 2020. https://www.iea.org/commentaries/the-carbon-footprint-of-streaming-video-fact-checking-the-headlines (accessed Sep. 07, 2022).

[2] S. Z. Attari, M. L. DeKay, C. I. Davidson, and W. B. de Bruin, “Public perceptions of energy consumption and savings,” *Proc Natl Acad Sci U S A*, vol. 107, no. 37, pp. 16054–16059, Sep. 2010, doi: 10.1073/pnas.1001509107.

[3] United States Environmental Protection Agency, “Learn about Energy and its Impact on the Environment,” 2022. https://www.epa.gov/energy/learn-about-energy-and-its-impact-environment (accessed Sep. 07, 2022).

[4] R. Schmalensee, T. M. Stoker, and R. A. Judson, “WORLD CARBON DIOXIDE EMISSIONS: 1950-2050,” 1998. [Online]. Available: http://direct.mit.edu/rest/article-pdf/80/1/15/1612398/003465398557294.pdf

[5] K. Kumari and S. Yadav, “Linear regression analysis study,” *Journal of the Practice of Cardiovascular Sciences*, vol. 4, no. 1, p. 33, 2018, doi: 10.4103/jpcs.jpcs_8_18.

[6] C. H. K. Albert Kwok, “AccuBattery.” 2016.

[7] B. G. Fernando Pérez, “Jupyter Notebook.” 2014. Accessed: Aug. 26, 2022. [Online]. Available: https://jupyter.org/

[8] S. C. C. W. P. J. H. Tom Preston-Werner, “GitHub.” 2008. Accessed: Aug. 26, 2022. [Online]. Available: https://github.com/

[9] A. K. and T. Teixeira. Adrien Treuille, “Streamlit.io.” 2018. Accessed: Aug. 26, 2022. [Online]. Available: https://streamlit.io/

[10] Alfonso González-Briones, Guillermo Hernández, Juan M. Corchado, Sigeru Omatu, and Mohd Saberi Mohamad, “Machine Learning Models for Electricity Consumption Forecasting: A Review,” 2019, Accessed: Aug. 10, 2021. [Online]. Available: https://ieeexplore-ieee-org.libezp2.utar.edu.my/document/8769508

[11] P. Stavast, “Prediction of Energy Consumption Using Historical Data and Twitter,” 2014. Accessed: Aug. 10, 2021. [Online]. Available: https://fse.studenttheses.ub.rug.nl/11665/

[12] J. R. Quinlan Basser, “LEARNING WITH CONTINUOUS CLASSES,” World Scientiic, 1992. Accessed: Aug. 21, 2022. [Online]. Available: https://sci2s.ugr.es/keel/pdf/algorithm/congreso/1992-Quinlan-AI.pdf

[13] Hila Becker, Mor Naaman, and Luis Gravano, “Learning Similarity Metrics forEvent Identification in Social Media,” p. 450, 2010, Accessed: Aug. 10, 2021. [Online]. Available: https://dl.acm.org/doi/abs/10.1145/1718487.1718524

[14] Dong Liu, Jianyu Zhao, and Chenyang Yang", “Energy-Saving Predictive Video Streaming with Deep Reinforcement Learning,” 2019, Accessed: Aug. 10, 2021. [Online]. Available: https://ieeexplore-ieee-org.libezp2.utar.edu.my/document/9013784

[15] D. Maulud and A. M. Abdulazeez, “A Review on Linear Regression Comprehensive in Machine Learning,” *Journal of Applied Science and Technology Trends*, vol. 1, no. 4, pp. 140–147, Dec. 2020, doi: 10.38094/jastt1457.

[16] Y. C. Li, H. H. Li, H. L. Li, and C. L. Yang, “Content-aware energy prediction for video streaming in mobile devices,” in *2009 International Symposium on VLSI Design,*

Automation and Test, VLSI-DAT '09, 2009, pp. 239–242. doi: 10.1109/VDAT.2009.5158139.

[17] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, "Deep reinforcement learning: A brief survey," *IEEE Signal Processing Magazine*, vol. 34, no. 6. Institute of Electrical and Electronics Engineers Inc., pp. 26–38, Nov. 01, 2017. doi: 10.1109/MSP.2017.2743240.

[18] scikit-learn developers (BSD License), "sklearn.preprocessing.LabelBinarizer," 2022. https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelBinarizer.html (accessed Apr. 15, 2022).

[19] S. Murugan, B. M. Kumar, and S. Amudha, "Classification and Prediction of Breast Cancer using Linear Regression, Decision Tree and Random Forest." Accessed: Apr. 15, 2022. [Online]. Available: https://ieeexplore-ieee-org.libezp2.utar.edu.my/document/8455058

[20] Jason Brownlee, "A Gentle Introduction to k-fold Cross-Validation," May 23, 2018. https://machinelearningmastery.com/k-fold-cross-validation/ (accessed Apr. 15, 2022).

[21] scikit-learn developers (BSD License), "sklearn.model_selection.GridSearchCV," 2022. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed Apr. 15, 2022).

[22] SASKIA, "HOW TO NORMALIZE THE RMSE," Jan. 07, 2019. https://www.marinedatascience.co/blog/2019/01/07/normalizing-the-rmse/ (accessed Apr. 15, 2022).

[23] David Dalpiaz, "Chapter 4 Regression for Statistical Learning," in *R for Statistical Learning*, 2020. Accessed: Apr. 15, 2022. [Online]. Available: https://daviddalpiaz.github.io/r4sl/regression-for-statistical-learning.html

[24] scikit-learn developers (BSD License), "3.2. Tuning the hyper-parameters of an estimator," 2022. https://scikit-learn.org/stable/modules/grid_search.html#randomized-parameter-optimization (accessed Apr. 15, 2022).

[25] Peter Flom, "The Disadvantages of Linear Regression," 2018. https://sciencing.com/disadvantages-linear-regression-8562780.html (accessed Apr. 01, 2022).

[26] D. Zhang, L. Yang, D. Meng, D. Xu, and J. Han, "SPFTN: A self-paced fine-tuning network for segmenting objects in weakly labelled videos," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, Nov. 2017, vol. 2017-January, pp. 5340–5348. doi: 10.1109/CVPR.2017.567.

[27] Google Brain team, "Transfer learning and fine-tuning," 2015. https://www.tensorflow.org/tutorials/images/transfer_learning (accessed Apr. 11, 2022).

APPENDIX A - DATA SOURCE

[1] Silver Mouse, "Top trending YouTube videos in Malaysia 2021", 2021. https://blog.silvermouse.com.my/2021/12/top-trending-youtube-videos-in-malaysia.html

[2] MITCHELL J, "Trending YouTube Video Statistics", 2019. https://www.kaggle.com/datasets/datasnaek/youtube-new

APPENDIX B – DATA EXPLORATION CODING

```
#1.ML Pipeline - Data exploration

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt


youtube=pd.read_csv("YoutubeDataset.csv")

youtube.head()

youtube.tail()

youtube.info()

youtube.describe()


youtube.hist(bins=50,figsize=(20,15))

plt.show()

#no capped

#tail heavy


youtube.plot(kind="scatter",x="Duration(h)",y="Energy(J)",c='r

plt.show()

youtube.plot(kind="scatter",x="Duration(h)",y="Total_Energy(k

plt.show()


corr_matrix=youtube.corr()

corr_matrix

corr_matrix['Total_Energy(kJ)'].sort_values(ascending=False)

# Use Views to predict total energy used
```

APPENDIX C – DATA EXPLORATION RESULT

Top 5 Records

ID	Title	Channel_title	Country	Views	Duration(h)	VivoY20s_Usage(mAh)	Pocox3pro_Usage(mAh)	Avg_Usage(mAh)	Energy(J)	Total_Ener
1	Drama Spontan 31: Avengers versi Kelantan	Syahmi Sazli	MY	2463512	0.210833	55.0	72.0	63.5	228.6	5
2	RUMAH AM KENA MASUK PER0MP4K II - PRANK MARSYA...	Ammar Nazhan	MY	2066020	0.158056	48.0	66.0	57.0	205.2	4
3	Luísa Sonza - Boa Menina	LuisaSonzaVEVO	BR	1585731	0.040833	10.0	NaN	10.0	36.0	
4	MASTERCHEF PROFISSIONAIS (30/10/2018) | PARTE ...	MasterChef Brasil	BR	764890	0.301944	98.0	NaN	98.0	352.8	2
5	Jerry Smith - Vou Falar Pra Tu (Clipe Oficial)	Jerry Smith	BR	426563	0.045278	10.0	NaN	10.0	36.0	

Last 5 Records

ID	Title	Channel_title	Country	Views	Duration(h)	VivoY20s_Usage(mAh)	Pocox3pro_Usage(mAh)	Avg_Usage(mAh)	Energy(J)	Total_Energy(kJ)
652	Cô Gái Hàn Quốc Siêu Dễ Thương Trong Màn Đối Đ...	VieTalents	KR	878998	0.101389	NaN	43.0	43.0	154.8	136069
653	북한이 망할 싯점에 이르렀다-뱅모 박성현	고교연합-자유대한연대	KR	38286	0.153611	NaN	50.0	50.0	180.0	6891
654	Minh Dự Chửi Việt Hương Sờ Mờ Lờ Khiến Trấn Th...	VieShows	KR	2552896	0.255278	NaN	109.0	109.0	392.4	1001756
655	김이브님 ♥SNS에서 핫한 아이템을 착용했더니 전투력(?)이 상승해버렸다	김이브님	KR	155068	0.056111	NaN	21.0	21.0	75.6	11723
656	बिहे गर्ने लाग्दा महिनावारी भएपछि.. हसाउँदा श्...	Mazzako TV	KR	181400	0.424722	NaN	156.0	156.0	561.6	101874

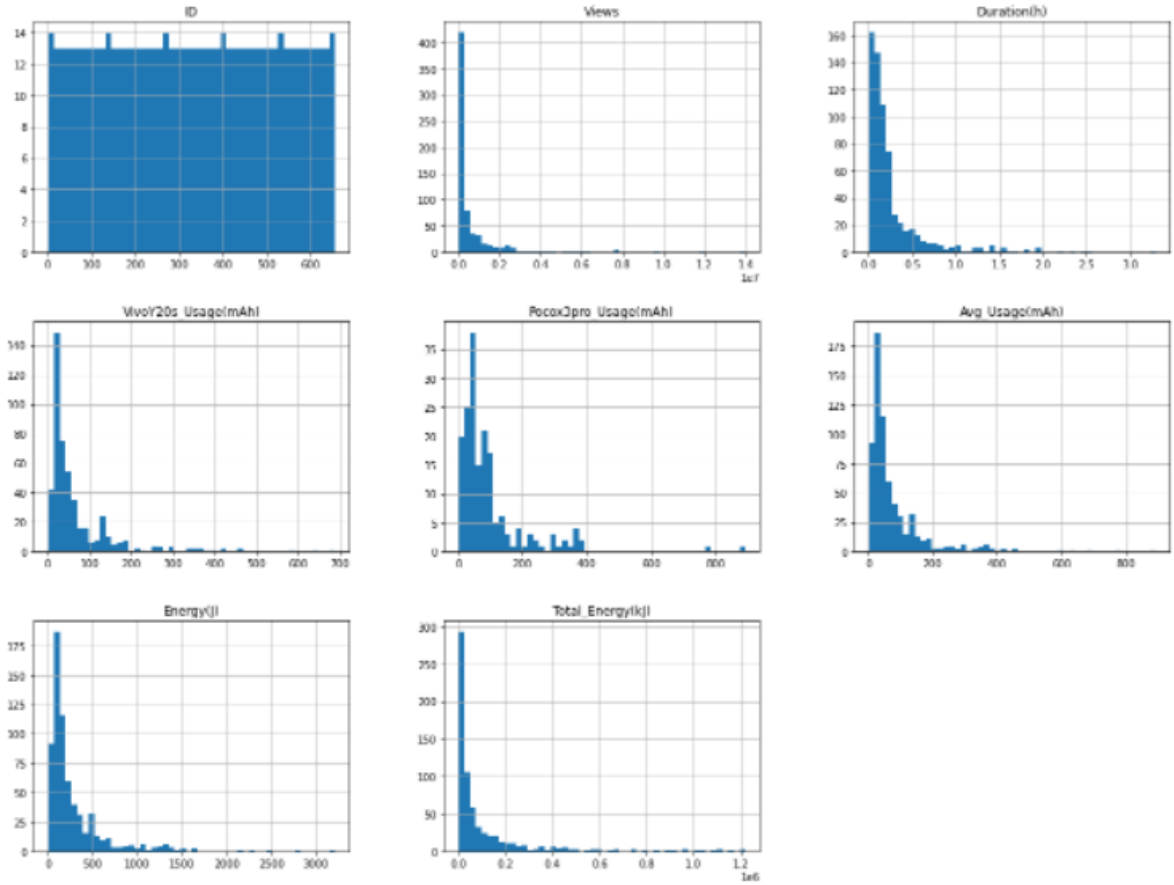
Data Properties

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 656 entries, 0 to 655
Data columns (total 11 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   ID                    656 non-null    int64
 1   Title                 656 non-null    object
 2   Channel_title         656 non-null    object
 3   Country               656 non-null    object
 4   Views                 656 non-null    int64
 5   Duration(h)           656 non-null    float64
 6   VivoY20s_Usage(mAh)   481 non-null    float64
 7   Pocox3pro_Usage(mAh)  177 non-null    float64
 8   Avg_Usage(mAh)        656 non-null    float64
 9   Energy(J)             656 non-null    float64
 10  Total_Energy(kJ)      656 non-null    int64
dtypes: float64(5), int64(3), object(3)
memory usage: 56.5+ KB
```

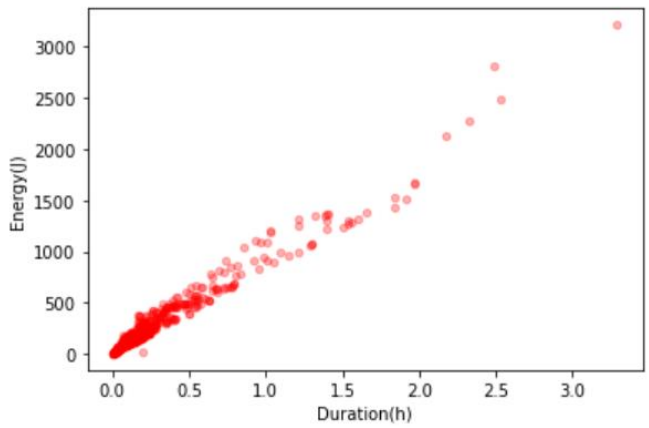
Data Description

	ID	Views	Duration(h)	VivoY20s_Usage(mAh)	Pocox3pro_Usage(mAh)	Avg_Usage(mAh)	Energy(J)	Total_Energy(kJ)
count	656.000000	6.560000e+02	656.000000	481.000000	177.000000	656.000000	656.000000	6.560000e+02
mean	328.500000	5.950644e+05	0.266347	68.577963	98.531073	76.685213	276.066768	1.019279e+05
std	189.515171	1.267564e+06	0.376480	87.930196	118.365480	98.012869	352.846329	1.863750e+05
min	1.000000	1.113600e+04	0.002500	1.000000	1.000000	1.000000	3.600000	4.000000e+01
25%	164.750000	5.829775e+04	0.068889	22.000000	36.000000	24.000000	86.400000	9.995000e+03
50%	328.500000	1.646450e+05	0.149444	38.000000	60.000000	43.000000	154.800000	2.854100e+04
75%	492.250000	5.505530e+05	0.263264	74.000000	103.000000	87.000000	313.200000	1.025822e+05
max	656.000000	1.397810e+07	3.291389	689.000000	893.000000	893.000000	3214.800000	1.220479e+06

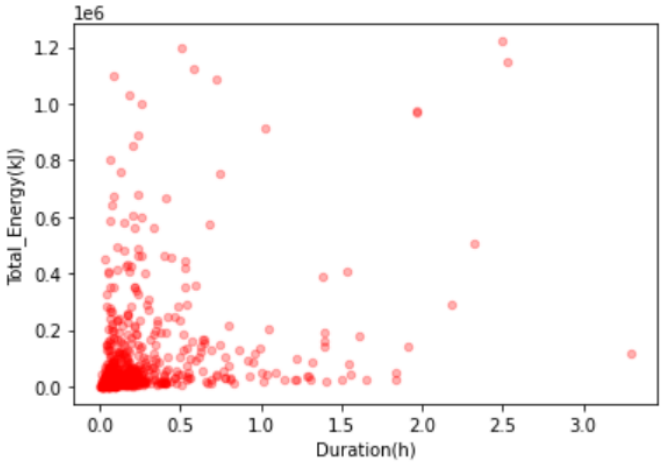
Data plot in histogram



Relationship between duration with energy attribute



relationship between duration with total energy attribute



Correlation matrix table

	ID	Views	Duration(h)	VivoY20s_Usage(mAh)	Pocox3pro_Usage(mAh)	Avg_Usage(mAh)	Energy(J)	Total_Energy(kJ)
ID	1.000000	-0.122456	0.057871	0.053403	-0.130754	0.120998	0.120998	-0.059974
Views	-0.122456	1.000000	-0.044949	-0.051191	-0.047453	-0.042981	-0.042981	0.623253
Duration(h)	0.057871	-0.044949	1.000000	0.992153	0.993584	0.986944	0.986944	0.191438
VivoY20s_Usage(mAh)	0.053403	-0.051191	0.992153	1.000000	0.999381	0.999516	0.999516	0.204801
Pocox3pro_Usage(mAh)	-0.130754	-0.047453	0.993584	0.999381	1.000000	0.999370	0.999370	0.196414
Avg_Usage(mAh)	0.120998	-0.042981	0.986944	0.999516	0.999370	1.000000	1.000000	0.212063
Energy(J)	0.120998	-0.042981	0.986944	0.999516	0.999370	1.000000	1.000000	0.212063
Total_Energy(kJ)	-0.059974	0.623253	0.191438	0.204801	0.196414	0.212063	0.212063	1.000000

Correlation matrix on total energy

```
Total_Energy(kJ)        1.000000
Views                   0.617412
Pocox3pro_Usage(mAh)    0.369911
Avg_Usage(mAh)          0.313847
Energy(J)               0.313847
Duration(h)             0.302592
VivoY20s_Usage(mAh)     0.301210
ID                     -0.024315
Name: Total_Energy(kJ), dtype: float64
```

APPENDIX D – WEB PAGE CODING

Main Page

```
import streamlit as st
from explore_page import show_explore_page
from predict_page import show_predict_page

def sidebar():
  page = st.sidebar.selectbox("Explore or Predict the energy.", ("Explore","Predict"))
  st.sidebar.image("bongo-cat.gif")
  return page

page = sidebar()
if page == "Explore":
  show_explore_page()
elif page == "Predict":
  show_predict_page()
```

Explore Page

```
import streamlit as st
import numpy as np
import pandas as pd
import io
from PIL import Image
import matplotlib.pyplot as plt
import plotly.figure_factory as ff

# CSS for table
hide_table_row_index = """
        <style>
        thead tr th:first-child {display:none}
        tbody th {display:none}
        </style>
        """
```

```
#define explore coding
def show_explore_page():
    st.title("Youtube Energy Estimate Page")

    st.write("""### Explore the *Data!* :sunglasses:""")

    #display image
    image = Image.open('Excited_Hedgehog.jpg')
    new_image = image.resize((125, 190))
    st.image(new_image, caption='Let’s Go!')

    #load data
    youtube=pd.read_csv("YoutubeDataset.csv")

    #radio
    choice = st.radio("Select a data explore way!",(
            "Top 5 records",
            "Last 5 records",
            "Data's information",
            "Data's description",
            "Target data's correlationship",
            "Plot Relation graph"))

    # Inject CSS with Markdown
    st.markdown(hide_table_row_index, unsafe_allow_html=True)

    if choice=="Top 5 records":
        st.write("Top 5 records of data is")
        st.table(youtube.head())
    elif choice=="Last 5 records":
        st.write("Last 5 records of data is")
        st.table(youtube.tail())
    elif choice=="Data's information":
```

```
        buffer = io.StringIO() #to store as memory file-like object
        youtube.info(buf=buffer) #info is a standard file object
        st.write("""Data's information is""",buffer.getvalue())
    elif choice=="Data's description":
        st.write("""Data's description is""",youtube.describe())
    elif choice=="Target data's correlationship":
        corr_matrix=youtube.corr()
        st.write("""Target data's correlationship
is""",corr_matrix['Total_Energy(kJ)'].sort_values(ascending=False))
    else:
        X = youtube['Total_Energy(kJ)'].values
        y = youtube['Views'].values
        fig = plt.figure(figsize=(5, 4))
        plt.plot (X, y, 'b.', markersize=6,label='samples')
        plt.xlabel ('TotalEnergy(kJ)', fontsize=18)
        plt.ylabel('Views', fontsize=18)
        st.plotly_chart(fig) #to create unique chart types

show_explore_page()
```

Predict Page

```
import streamlit as st
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.figure_factory as ff
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
```

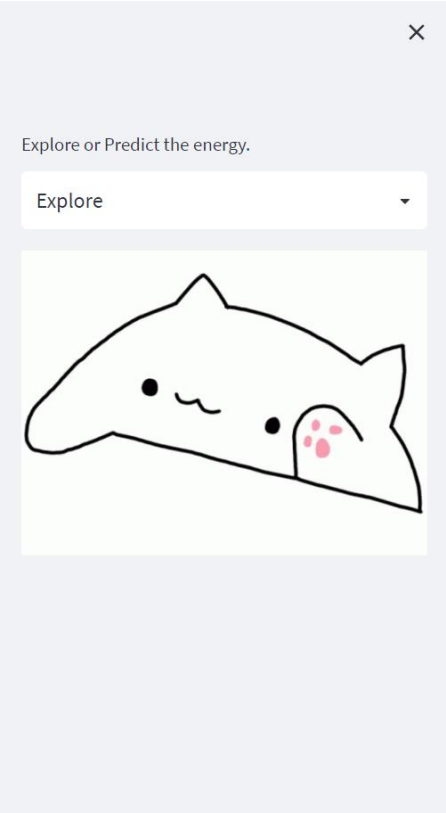
```
#define predict coding
class MyLinearRegression :

    def __init__(self, theta = None):
        self.theta = np.array(theta)
    def fit(self, X, y):
        m, n = X.shape
        Xo= np.hstack((np.ones((m,1)),X))
        self.theta =np.linalg.pinv(Xo.T.dot(Xo)).dot(Xo.T.dot(y))
    def predict(self, X):
        m,n=X.shape
        Xo=np.hstack((np.ones((m,1)),X))
        h=Xo.dot(self.theta)
        return h

def RMSE(y_true, y_pred):
    rmse=np.sqrt(np.mean((y_true-y_pred)**2))
    return rmse

@st.cache #cache the data, no need execute everytime
def load_data():
    youtube=pd.read_csv("YoutubeDataset.csv")
    X = youtube[['Views']].values
    y = youtube ['Total_Energy(kJ)'].values
    return X,y

#define calculation
def calculation(X):
    #y=mx+c
    y = 0.09*X+47907
    return y

#define page
```

```
def show_predict_page():
    st.title("Youtube Energy Estimate Page")

    st.write("""### Predict the *Data!* :smile:""")

    # Load data
    X,y = load_data()

    mylinreg = MyLinearRegression()
    mylinreg.fit(X, y)
    st.write("Fitted theta:",mylinreg.theta)

    # Show the learnt model
    h_pred = mylinreg.predict(X)
    st.markdown(f"""Linear Regression model trained : RMSE={RMSE(y, h_pred):.2f}""")
#Display string formatted
    st.success('Model trained successfully') #display success message

    # Visualize the result
    Xtest = np.linspace(0, 15000000, 100).reshape(-1, 1)
    h_pred = mylinreg.predict(Xtest)
    fig = plt.figure(figsize=(7, 5))
    plt.plot (X, y, 'b.', markersize=12,label='samples')
    plt.plot (Xtest, h_pred, 'r-', label = 'model (MyLinearRegression)')
    plt.xlabel ('Views', fontsize=18)
    plt.ylabel('TotalEnergy(kJ)', fontsize=18)
    plt.axis([0,15000000,0,1500000])
    plt.legend(loc='best')
    #plt.show()
    st.plotly_chart(fig) #to create unique chart types

    # Calculation
```

```
    pred_view = st.slider('Enter the number of views', min_value=0, max_value=15000000,
value=7500000, step=10000)
    pred_energy = calculation(pred_view)
    st.markdown(f"""Estimated energy consume for the selected view is : {pred_energy:.0f}
kJ""")
show_predict_page()
```

APPENDIX E – WEB PAGE SIMULATION

Main Page



Explore Page

Youtube Energy Estimate Page

Explore the *Data!* 😎

Lets Go!

Select a data explore way!

- (●) Top 5 records
- ( ) Last 5 records
- ( ) Data's information
- ( ) Data's description
- ( ) Target data's correlationship
- ( ) Plot Relation graph

Top 5 records of data is

ID	Title	Channel_title	Country	Views	Duration(h)	VivoY20s_Usage(mAh)	Pocox
1	Drama Spontan 31: Avengers versi Kelantan	Syahmi Sazli	MY	2463512	0.2108	55.0000	
2	RUMAH AM KENA MASUK PER0MP4K !! - PRANK MARSYA , AIRA DAN TEAH T4KUT !!	Ammar Nazhan	MY	2066020	0.1581	48.0000	
3	Luísa Sonza - Boa Menina	LuisaSonzaVEVO	BR	1585731	0.0408	10.0000	
4	MASTERCHEF PROFISSIONAIS (30/10/2018) | PARTE 2 | EP 11 | TEMP 03	MasterChef Brasil	BR	764890	0.3019	98.0000	
5	Jerry Smith - Vou Falar Pra Tu (Clipe Oficial)	Jerry Smith	BR	426563	0.0453	10.0000	

Predict Page

Youtube Energy Estimate Page

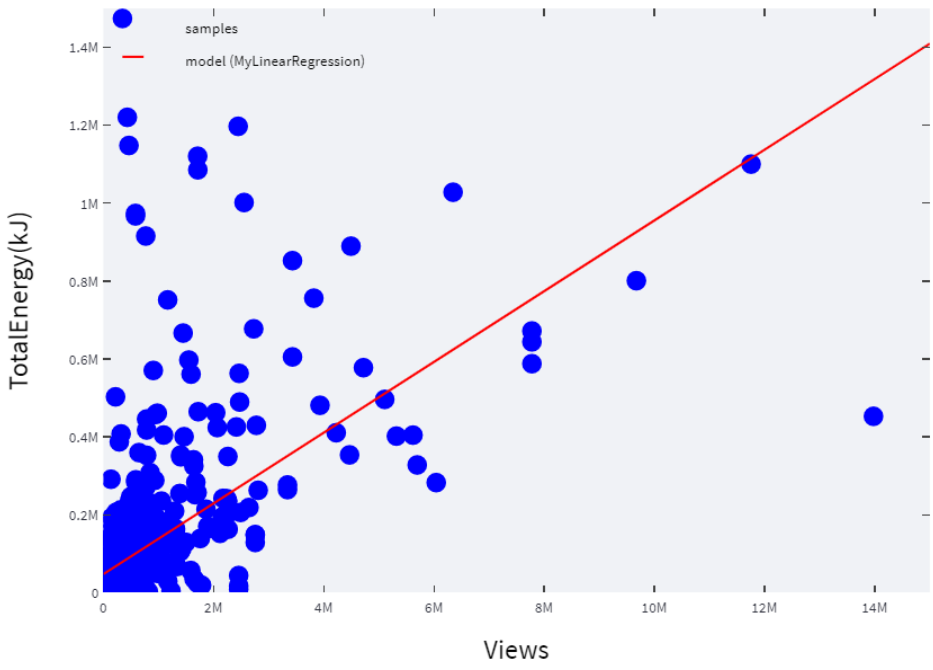
Predict the *Data!* 😄

Fitted theta:

	0
0	47,907.6264
1	0.0908

Linear Regression model trained : RMSE=146498.28

Model trained successfully



Enter the number of views

7500000

0 15000000

Estimated energy consume for the selected view is : 722907 kJ

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.:1**
Student Name & ID: Ng Jiun Shen, 2000362	
Supervisor: Ms Tseu Kwan Lee	
Project Title: Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning techniques	

1. WORK DONE • Fine Tuning methods have been studied.
2. WORK TO BE DONE • Fine Tuning methods will be applied next.
3. PROBLEMS ENCOUNTERED No.
4. SELF EVALUATION OF THE PROGRESS On track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 2 - 5**
Student Name & ID: Ng Jiun Shen, 2000362	
Supervisor: Ms Tseu Kwan Lee	
Project Title: Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning techniques	

1. WORK DONE

- A new method on fine tuning coding on the dataset has been applied.

2. WORK TO BE DONE

- Website development techniques will be studied next.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

On track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 6**
Student Name & ID: Ng Jiun Shen, 2000362	
Supervisor: Ms Tseu Kwan Lee	
Project Title: Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning techniques	

1. WORK DONE

- Website Development tools and algorithms have been studied.
- GitHub software has been chosen for storing the coding files.
- Streamlit.io has been chosen for the platform for building machine learning web page.

2. WORK TO BE DONE

- Coding for machine learning web page will be done next.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

On track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 7-10**
Student Name & ID: Ng Jiun Shen, 2000362	
Supervisor: Ms Tseu Kwan Lee	
Project Title: Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning techniques	

1. WORK DONE

- The machine learning web page has done created.
- Python and CSS language are applied.

2. WORK TO BE DONE

- The report will start next week.

3. PROBLEMS ENCOUNTERED

Streamlit.io library has lack of some library. Requirement text file has included to let the streamlit.io download library.

4. SELF EVALUATION OF THE PROGRESS

On track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 11**
Student Name & ID: Ng Jiun Shen, 2000362	
Supervisor: Ms Tseu Kwan Lee	
Project Title: Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning techniques	

1. WORK DONE

- Chapter 1,2,3,7 have done.

2. WORK TO BE DONE

- The first draft of FYP2 will be done next.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

On track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

Trimester, Year: Y3S2	**Study week no.: 12**
Student Name & ID: Ng Jiun Shen, 2000362	
Supervisor: Ms Tseu Kwan Lee	
Project Title: Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning techniques	

1. WORK DONE

- The first draft of the FYP2 has been done and submitted to the supervisor to check.

2. WORK TO BE DONE

- The second draft of FYP2 will be done next.
- Chapters 4 and 5 of the report will start next week.
- The first draft of the report will be submitted to the supervisor to check on next week.

3. PROBLEMS ENCOUNTERED

No

4. SELF EVALUATION OF THE PROGRESS

On track

Supervisor's signature

Student's signature

FINAL YEAR PROJECT WEEKLY REPORT

(Project II)

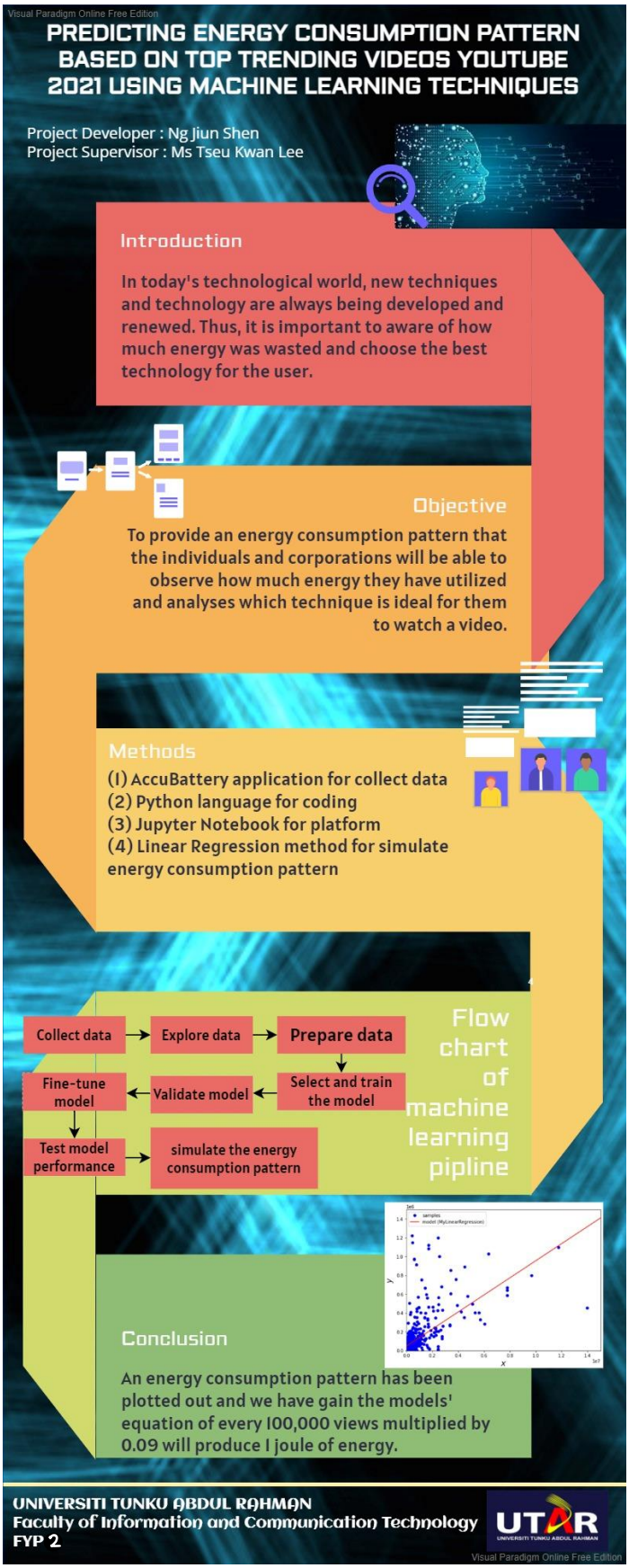
Trimester, Year: Y3S2	**Study week no.: 13**
Student Name & ID: Ng Jiun Shen, 2000362	
Supervisor: Ms Tseu Kwan Lee	
Project Title: Predicting energy consumption pattern based on top trending videos YouTube 2021 using machine learning techniques	

1. WORK DONE • The final draft of FYP2 has been done.
2. WORK TO BE DONE No
3. PROBLEMS ENCOUNTERED No
4. SELF EVALUATION OF THE PROGRESS On track

Supervisor's signature

Student's signature

POSTER


Visual Paradigm Online Free Edition
PREDICTING ENERGY CONSUMPTION PATTERN BASED ON TOP TRENDING VIDEOS YOUTUBE 2021 USING MACHINE LEARNING TECHNIQUES
Project Developer : Ng Jiun Shen
Project Supervisor : Ms Tseu Kwan Lee
Introduction
In today's technological world, new techniques and technology are always being developed and renewed. Thus, it is important to aware of how much energy was wasted and choose the best technology for the user.
Objective
To provide an energy consumption pattern that the individuals and corporations will be able to observe how much energy they have utilized and analyses which technique is ideal for them to watch a video.
Methods
(1) AccuBattery application for collect data
(2) Python language for coding
(3) Jupyter Notebook for platform
(4) Linear Regression method for simulate energy consumption pattern
Flow chart of machine learning pipline
Collect data
Explore data
Prepare data
Select and train the model
Validate model
Fine-tune model
Test model performance
simulate the energy consumption pattern
Conclusion
An energy consumption pattern has been plotted out and we have gain the models' equation of every 100,000 views multiplied by 0.09 will produce 1 joule of energy.
UNIVERSITI TUNKU ABDUL RAHMAN
Faculty of Information and Communication Technology
FYP 2
UTAR
Visual Paradigm Online Free Edition

PLAGIARISM CHECK RESULT

ORIGINALITY REPORT

10%	9%	4%	%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	fict.utar.edu.my Internet Source	2%
2	riyan-aditya.github.io Internet Source	1%
3	towardsdatascience.com Internet Source	1%
4	datascienceplus.com Internet Source	<1%
5	www.danli.org Internet Source	<1%
6	dspace.daffodilvarsity.edu.bd:8080 Internet Source	<1%
7	eprints.utar.edu.my Internet Source	<1%
8	github.com Internet Source	<1%
9	www.section.io Internet Source	<1%

10	www.itdaan.com Internet Source	<1 %
11	index-of.es Internet Source	<1 %
12	Alfonso Gonzalez-Briones, Guillermo Hernandez, Juan M. Corchado, Sigeru Omatu, Mohd Saberi Mohamad. "Machine Learning Models for Electricity Consumption Forecasting: A Review", 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), 2019 Publication	<1 %
13	Yogendra Narayan Pandey, Ayush Rastogi, Sribharath Kainkaryam, Srimoyee Bhattacharya, Luigi Saputelli. "Machine Learning in the Oil and Gas Industry", Springer Science and Business Media LLC, 2020 Publication	<1 %
14	Samiksha Marne, Shweta Churi, Maheshwari Marne. "Predicting Breast Cancer using effective Classification with Decision Tree and K Means Clustering technique", 2020 International Conference on Emerging Smart Computing and Informatics (ESCI), 2020 Publication	<1 %

15	pages.github.rpi.edu Internet Source	<1 %
16	docplayer.net Internet Source	<1 %
17	praison.com Internet Source	<1 %
18	www.scribd.com Internet Source	<1 %
19	Poornachandra Sarang. "Artificial Neural Networks with TensorFlow 2", Springer Science and Business Media LLC, 2021 Publication	<1 %
20	arun-aiml.blogspot.com Internet Source	<1 %
21	Vera Rusakova, Alexander Rusakov, Ekaterina Savateeva. "Methodological Aspects of Propaedeutics of Studying Artificial Intelligence by Medical Students", 2022 2nd International Conference on Technology Enhanced Learning in Higher Education (TELE), 2022 Publication	<1 %
22	oapub.org Internet Source	<1 %
23	www.coursehero.com Internet Source	<1 %

24	www.iea.org Internet Source	<1 %
25	www.youtube.com Internet Source	<1 %
26	pythonhealthcare.org Internet Source	<1 %
27	djvu.online Internet Source	<1 %
28	www.ncbi.nlm.nih.gov Internet Source	<1 %
29	cag.gov.in Internet Source	<1 %
30	machinelearningmastery.com Internet Source	<1 %
31	www.doria.fi Internet Source	<1 %
32	www.ieomsociety.org Internet Source	<1 %
33	medium.com Internet Source	<1 %
34	mubaris.com Internet Source	<1 %
35	www.delftstack.com Internet Source	<1 %

36	dokumen.pub Internet Source	<1 %
37	hdl.handle.net Internet Source	<1 %
38	umpir.ump.edu.my Internet Source	<1 %
39	Bo Chen. "A Design Model of Music Teaching and Auxiliary System Based on Artificial Neural Networks", Computational Intelligence and Neuroscience, 2022 Publication	<1 %
40	David O'Connor, Evelyn M.R. Lake, Dustin Scheinost, R. Todd Constable. "Bootstrap aggregating improves the generalizability of Connectome Predictive Modelling", Cold Spring Harbor Laboratory, 2020 Publication	<1 %
41	Dong Liu, Jianyu Zhao, Chenyang Yang. "Energy-Saving Predictive Video Streaming with Deep Reinforcement Learning", 2019 IEEE Global Communications Conference (GLOBECOM), 2019 Publication	<1 %
42	Juliano Marçal Lopes. "Stochastic optimization and machine learning applied in the demand forecast, allocation and distribution of vaccines between Brazilian states.",	<1 %

	Universidade de Sao Paulo, Agencia USP de Gestao da Informacao Academica (AGUIA), 2022 Publication	
43	Susan B. Rodriguez, Mark A. Thornton, Roy J. Thornton. "Raman Spectroscopy and Chemometrics for Identification and Strain Discrimination of the Wine Spoilage Yeasts Saccharomyces cerevisiae, Zygosaccharomyces bailii, and Brettanomyces bruxellensis", Applied and Environmental Microbiology, 2013 Publication	<1 %
44	Thanasis Vafeiadis, Nikolaos Kolokas, Nikolaos Dimitriou, Angeliki Zacharaki et al. "A comparison of 2DCNN network architectures and boosting techniques for regression-based textile whiteness estimation", Simulation Modelling Practice and Theory, 2021 Publication	<1 %
45	irep.ntu.ac.uk Internet Source	<1 %
46	jesit.springeropen.com Internet Source	<1 %
47	staff.utar.edu.my Internet Source	<1 %
48	www.freecodecamp.org Internet Source	

		<1 %
49	www.mdpi.com Internet Source	<1 %
50	Vaibhav Verdhan. "Supervised Learning with Python", Springer Science and Business Media LLC, 2020 Publication	<1 %

PLAGIARISM CHECK RESULT

Universiti Tunku Abdul Rahman			
Form Title: Supervisor's Comments on Originality Report Generated by Turnitin for Submission of Final Year Project Report (for Undergraduate Programmes)			
Form Number: FM-IAD-005	Rev No.: 0	Effective Date: 01/10/2013	Page No.: 1of 1

FACULTY OF INFORMATION AND COMMUNICATION TECHNOLOGY

Full Name(s) of Candidate(s)	Ng Jiun Shen
ID Number(s)	20ACB00362
Programme / Course	FICT/CS
Title of Final Year Project	Predicting energy consumption pattern based on top trending videos youtube 2021 using machine learning techniques

Similarity	**Supervisor's Comments (Compulsory if parameters of originality exceeds the limits approved by UTAR)**
Overall similarity index: 10 % **Similarity by source** Internet Sources: 9 % Publications: 4 % Student Papers: 0 %	ok
Number of individual sources listed of more than 3% similarity: 0	ok
Parameters of originality required and limits approved by UTAR are as Follows: **(i) Overall similarity index is 20% and below, and** **(ii) Matching of individual sources listed must be less than 3% each, and** **(iii) Matching texts in continuous block must not exceed 8 words** ***Note: Parameters (i) – (ii) shall exclude quotes, bibliography and text matches which are less than 8 words.***	

Note Supervisor/Candidate(s) is/are required to provide softcopy of full set of the originality report to Faculty/Institute

Based on the above results, I hereby declare that I am satisfied with the originality of the Final Year Project Report submitted by my student(s) as named above.

Signature of Supervisor

Name: Ms Tseu Kwan Lee

Date: 9 SEPTEMBER 2022

Signature of Co-Supervisor

Name: _____

Date: _____

FYP2 CHECKLIST

UNIVERSITI TUNKU ABDUL RAHMAN

FACULTY OF INFORMATION & COMMUNICATION TECHNOLOGY
(KAMPAR CAMPUS)

CHECKLIST FOR FYP2 THESIS SUBMISSION

Student Id	2000362
Student Name	Ng Jiun Shen
Supervisor Name	Ms Tseu Kwan Lee

TICK (√)	DOCUMENT ITEMS Your report must include all the items below. Put a tick on the left column after you have checked your report with respect to the corresponding item.
	Front Plastic Cover (for hardcopy)
√	Title Page
√	Signed Report Status Declaration Form
√	Signed FYP Thesis Submission Form
√	Signed form of the Declaration of Originality
√	Acknowledgement
√	Abstract
√	Table of Contents
√	List of Figures (if applicable)
√	List of Tables (if applicable)
√	List of Symbols (if applicable)
√	List of Abbreviations (if applicable)
√	Chapters / Content
√	Bibliography (or References)
√	All references in bibliography are cited in the thesis, especially in the chapter of literature review
√	Appendices (if applicable)
√	Weekly Log
√	Poster
√	Signed Turnitin Report (Plagiarism Check Result - Form Number: FM-IAD-005)
√	I agree 5 marks will be deducted due to incorrect format, declare wrongly the ticked of these items, and/or any dispute happening for these items in this report.

*Include this form (checklist) in the thesis (Bind together as the last page)

I, the author, have checked and confirmed all the items listed in the table are included in my report.

_____

(Signature of Student)
Date: 9 SEPTEMBER 2022