****

**20403504**

**Supervisor: Damla Kilic**

**Module Code: COMP 3003**

****

**Second-Hand Car Price Prediction with Machine Learning**

Submitted April 2022, in partial fulfillment of the conditions for the awards of the degree **BSc (Hons) Computer Science with Artificial Intelligence**

**20403504**

School of Computer Science

University of Nottingham

I hereby declare that this dissertation is all my own work, except as indicated in the text:

**Signature\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Date\_\_\_/\_\_\_/\_\_\_**

I hereby declare that I have all the necessary rights and consents to publicly distribute this dissertation via the University of Nottingham’s e-dissertations archive

Abstract

This dissertation topic is about how the approaches of two different models can be used to predict the price of the second-hand car market. The support vector machine (SVR) and the linear regression will be implemented and compared in terms of scoring metrics which are the root mean squared error (RMSE) and the mean absolute error (MAE) to give insights into the second-hand car market.

Acknowledgment

I would like to thank my supervisor, Ms. Damla Kilic for her guidance, patience, and time to complete this dissertation project. I would like also want to thank my family, friends, and colleague in Nottingham for their never-ending mental support, advice, and time to hear the struggle of completing this project.

[**Introduction**](#_tsb8piwcso56) **1**

[Motivation](#_39iolaq5xgim) 1

[Aims and Objectives](#_dtemq2uonw7y) 1

[**Related Works**](#_mw5lu6bshp2b) **2**

[**Description of the Work**](#_4xul271dpvdg) **4**

[Methodology](#_6cmv0xbp001m) 4

[Preliminary Research](#_5mkw8v2wkag0) 4

[Support Vector Machine](#_v8s2e2wpj6ry) 4

[Linear Regression](#_otat0jhqade1) 6

[Evaluation Metrics](#_h61zxvljki4s) 7

[Design](#_8rywg5pie8xd) 7

[Support Vector Regression](#_ouvbp4svaqdk) 9

[Linear Regression](#_tlws03805jhs) 9

[**Implementation**](#_rillnee9brgi) **10**

[Support Vector Regression](#_wifzowz6dzbh) 10

[Linear Regression](#_uctkiy1hkgym) 13

[**Results**](#_r25lioegumai) **16**

[**Summary and Reflection**](#_nc46hoz23h46) **17**

[Project Management](#_hiuzntj08ruz) 17

[Contribution and Reflections](#_p819tbj9a4od) 19

[Limitation](#_yyl4kme3nxig) 19

[Future Works](#_nk2bopcjpms1) 19

[**Bibliography**](#_xcgpv8whpden) **20**

# Introduction

## Motivation

The main product in the automotive industry is the automobile. Around 70% of the production in the industrial country is dominated by car production to push the economy [1]. In the other case of the same automotive industry, second-hand car sales have been increasing in the number of sales compared to new car sales. As the result of an increased number of sales, the second-hand car market has created more business opportunities that are good for sellers and buyers. Although the sales are having a large number of sales, the second-hand market has problems to be dealt with to maintain its dominance one of them is the new car dealership that is able to offer a second-hand car to the customer if the price of a new car is expensive[2]. Secondly, the unpredicted and always changing price of the second-hand market[1].

Currently, the best solution for predicting the used car price is to find another variable that could affect the price[2]. The motivation behind the project is to predict the market for the buyer and seller in order to predict the car price. As a matter of fact, there are other variables that affect the used car price. An article mentioned about 10 factors that affect the price. For example, it is important to realize that stickers on the exterior of a car could affect the price because potential buyers have a different perception of it [3]. Secondly, the color choice of a car could affect the resale price. For instance, the colors black, gray, and white are the safe colors for a car to survive in the market for a few years [3]. As a result, these variables can help this project to predict the price of used cars using machine learning and prove the relevance between those factors and the car price.

## Aims and Objectives

The project aims to develop and design a prediction model for the second-hand car market from the dataset and visualize the predicted output. The quality of the prediction model of this project can be determined with the RMSE (Root Mean Square Error) metric and the MAE (Mean Absolute Error) score to see the performance of a regression model. In order to achieve that, a high percentage of prediction accuracy is also important in this project so it will be the main goal for this project. The objectives of the project are:

1. Investigate the current approach for the existing solution in the second-hand car market.
2. Design and develop a prediction model using an SVR(Support Vector Regression) to predict the price of the car.
3. Design and develop a prediction model using a Linear Regression to predict the price of the car.
4. Visualize the forecast to provide a better understanding and readable output from the project.
5. Comparing both models ( SVR and Linear Regression) using metrics.

# Related Works

There are many issues to be addressed in order to achieve a fully functional model in the state of developing and designing a model. The method to create a prediction model is vast and needs to be carefully considered. However, from a research standpoint, the project can be divided into a few major issues that must be addressed to achieve its goal.

Predicting the price of a second-hand car has been widely researched in multiple studies. The research done by Listiani for their thesis for a master's degree stated that the performance and precision of the prediction model made with support vector regression are higher compared to another multi-variable regression model in dealing with a dataset with more dimensions [4]. Hence, in their thesis, they also stated that the outcome from the kernel stability is affected by the form of the data. Moreover, Listiani also used the radial basis function, polynomial, and linear kernel to solve the problem in their project and compare the three kernels with statistical regression based on the generalized error as shown in the figure below [4].

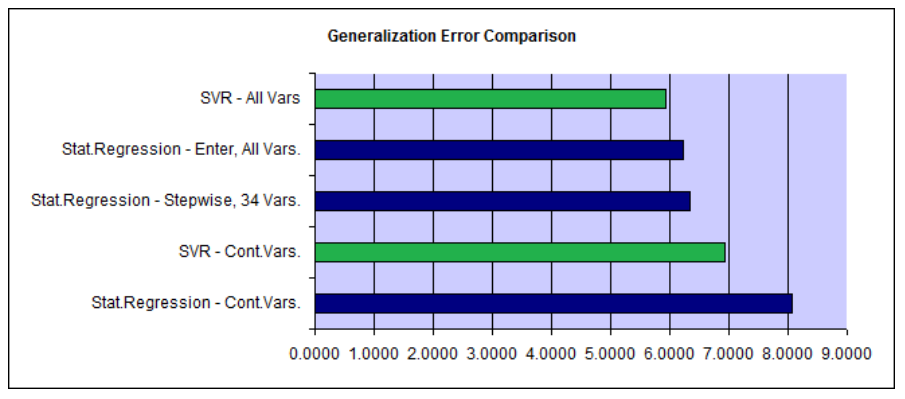


Figure 1: RMSE of Different Statistical Regression and SVR (Listiani, 2019)

As shown in the figure above, the root mean squared error (RMSE) score for the SVR is lower than the expectation besides the advantage of SVR in the benchmark. In conclusion from Listiani’s thesis, their experiment proved that SVR has a higher capability with the right kernel resulting in higher accuracy [4]. Before finding these results, Listiani analyzed the learning curve of the three kernels and also tuned the hyperparameter of the kernels with 10.000 data entries as shown in Figure 2.

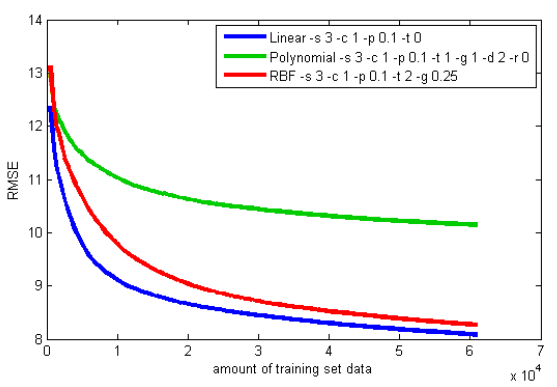


Figure 2. Learning curves of three kernels.

Chuyang Jin had a study titled *Price Prediction of Used Cars Using Machine Learning* with several algorithms and some of them are support vector regression and linear regression [5]. In their paper, Chuyang Jin stated that SVR can tolerate the error more superior than linear regression with a pre-set value. During the training process, the SVR algorithm can ignore the error outside the pre-set value margin. Chuyang Jin also stated that in linear regression, the model is vulnerable to outliers and can cause a disturbance in the regression [5].

Another work related to the prediction of car prices has been done by Satioglu. In the research, Satioglu and colleagues used the linear regression model with multi-features such as engine power, engine capacity, wheelbase, and maximum speed features to predict the price in Turkey [6]. In building the model, they split the 80% data used for training and 20% data used for testing and used the RMSE, mean absolute error (MAE), and r squared (R2) as the scoring metrics. Finally, the result of the actual price and the predicted price can be seen in Figure 2 below using the first 25 predicted records [6].

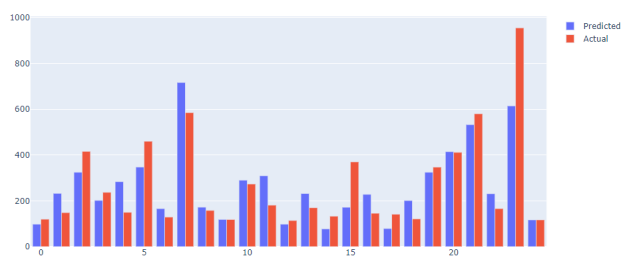


Figure 3. Satioglu Car Price Prediction in Turkey (Satioglu, 2021).

In conclusion, the Support Vector Regression and Linear Regression that have been done by Listiani, Jin, and Satiaglu can support this project in predicting the car price with both approaches and comparing both approaches based on what has been learned through their findings from the past work.

# Description of the Work

## Methodology

### Preliminary Research

#### Support Vector Machine

The idea of the support vector machine (SVM) was first created as a solution to a classification problem by Vladimir Vapnik in 1995 [7]. The SVM works in multi-dimensional space by separating data points in each class with the hyperplane and creating a line to separate the two classes [8]. This could solve a problem where a linear line could not separate the two classes. For example, the SVM classification method can be seen in *Figure 3* below.

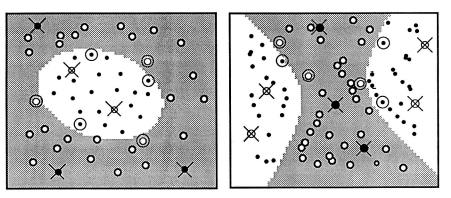


Figure 4. The SVM classification example by Vladimir Vapnik in 1995.

On the other hand, support vector regression (SVR) is a supervised machine learning that uses another form of support vector machine with the same theory to solve the regression problem [8]. The method of SVR is to perform a complex calculation of an input space to specified numbers inside the process of training data [9]. The main goal of the SVR is to find the closest data to the hyperplane and the closest data that exist between the hyperplane has the highest score and can be considered the best fit [8]. Furthermore, the SVM has 3 hyperparameters that help the model find the best fit, which are kernel function, hyperplane, and boundary lines. Three kernels are being used for this project which are radial basis function, polynomial function, and linear function. These 3 kernels are the common function to implement and perform a calculation with unknown knowledge of the data or the records inside of the data points [10].



Figure 5. Radial Basis Function kernel from *towardsdatascience.com*

RBF kernel as shown in the figure above is the most common kernel used in support vector machines [10]. The reason RBF is popular is that the ability to solve the space complexity problem and has the similarity to the K-Nearest Neighbour algorithm [11]. In mathematical terms, *y* is the free parameter that balances the number of impacts on two points upon each other. In addition, RBF can expand the dimensions into infinite numbers.



Figure 6. Linear kernel from *towardsdatascience.com*

The linear kernel can be known as the non-kernel because of its simplicity compared to other kernels [10]. To clarify, data that is projected in this kernel is the inner product of the *x* and *y* where *c* is constant and optional. The linear kernel is more practical for datasets with variant features but does not involve the inner product in higher dimension space.



Figure 7. Polynomial kernel from *towardsdatascience.com*

The polynomial kernel has the similarity with linear kernel although it does involve the inner product into the higher dimension [10]. The parameters in this kernel are *a, c,*  and *d* whereas *d* is commonly set into 2 to avoid overfitting. Notably, the polynomial kernel is also powerful in solving non-linear problems.

#### Linear Regression

Linear regression is a method to perform a calculation of the relationship between two variables [12]. Linear regression develops a statistical correlation between the two variables in a linear graph. In addition, If the dependent variable is known, it enables the calculation to predict it [13]. The main idea is to find the line that best fits the data. The best fit line is the one with the smallest total of prediction errors (across all data). The distance between the line and the data point is called an error [20].



Figure 8. Linear Regression Function.

As shown in Figure 7 above, the equation for linear regression takes the form of, *y* is the independent variable and *A* and *B*  are the coefficients that determine the descent and intercept the calculation [12]. As illustrated in Figure 8 on the visualization of linear regression output.

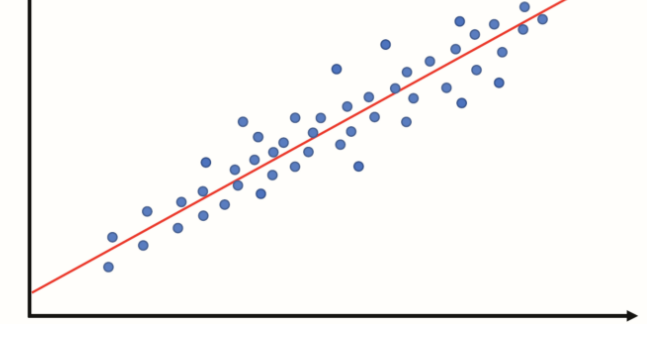
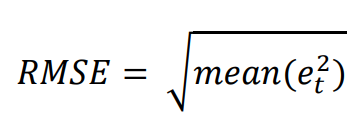
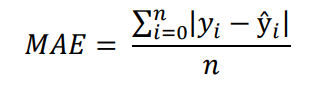


Figure 9. Linear Regression output.

### Evaluation Metrics



Root Mean Squared Error (RMSE) is a function to calculate the error in a model in predicting assessable data. The objective of RMSE is to find the distance between the vector of the predicted variable and the vector of the actual variable from the regression line [14]. In particular, it defines how focused the data is around the best-fitted regression line.



Mean Absolute Error (MAE) is a function to calculate the average of the absolute error formula. The main objective of this formula is to measure the error between the predicted value and the real value and display the expected error from the forecasting.

## Design

In order to achieve the expected output, this project will use Python language for the machine learning process. For this reason, Python is the most fitted language and the most popular language for machine learning [15]. This project will also use Python libraries such as NumPy, math, seaborn, matplotlib, and also pandas to solve the modeling and testing problems of this project. Furthermore, deep research for this project is a must in order to acquire the best possible outcome of a research-based dissertation.

* + 1. **Dataset**

The dataset retrieved from Kaggle provides 8 features that contain the components and brand of a car from different models.

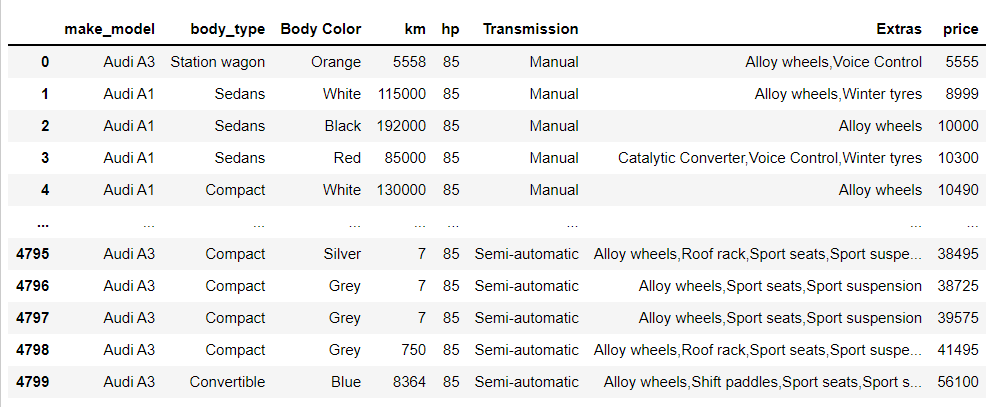


Figure 10. The snippet of the dataset.

The aspects that are applied in the feature selection for this project are by ranking the features based on the type of data in the features (numerical data). The objective of feature selection in this project is to reduce the complexity and also easier to understand the data. The method for this feature selection is by using the RFE (Recursive Feature Elimination) performs elimination to reduce the complexity by removing features one by one until the amount of selected features is left.



Figure 11. The result from RFE.

In addition to this project, to achieve a better understanding of the predictor, in this case, is the price feature. The distribution plot for the price is done as it is shown in Figure 11 by using a library called seaborn.

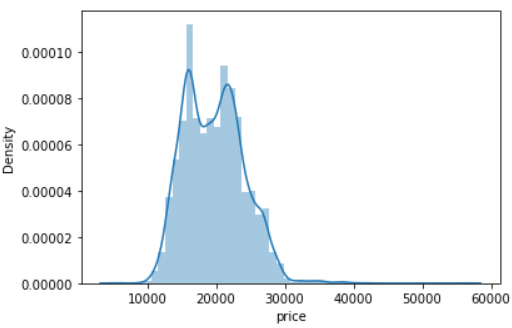


Figure 12. Price distribution in dataset.

### Support Vector Regression

For the objective of this project, the three functions of kernels were implemented for the comparison between the three kernels. The three kernels are radial basis function, polynomial, and linear kernel. Afterward, the features that were used for this project are the price feature and kilometer feature. The next step is feature scaling before fitting the features into the model to ensure the gradient descent avoids unnecessary iteration.

Finally, the predicted results from the three models were plotted to measure the performance between the three models.

### Linear Regression

For the linear regression model, the same features and feature scaling were applied for this model. In order to improve the performance of the linear regression model, cross-validation was applied to observe the performance of the model. After the cross-validation was implemented, the next step is to observe the accuracy of the 10 fold cross-validation using a library to perform a calculation for the accuracy. The final step was to visualize and observe the result from the linear regression model.

# Implementation

The programming language used for designing and developing this project in Python and was using the Jupyter Notebook in the Conda environment. The reason for Python being used was the simplicity of its syntax and the wide variety of libraries.

## Support Vector Regression

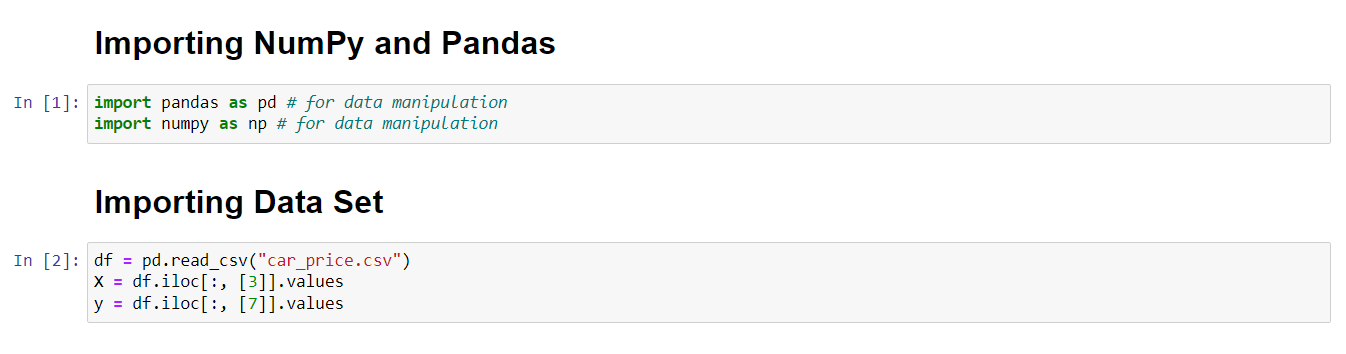
****

Figure 13. Importing Data Set and Dependencies (SVR).

As shown in Figure 13 above, this project is using NumPy (Numerical Python) and Pandas for the dependencies. The NumPy library is known for its wide variety of mathematical operations such as linear algebra calculation and the ability to manipulate arrays as shown in Figure 13. Identically to NumPy, the Pandas library is also useful in the data science field due to its performance in analytics and ease to use. After importing the required dependencies, the Pandas library has a function to store data in a built-in 2D-object memory called DF (DataFrame). In this case, the function is used to store the dataset in this project and to select the features using *df.iloc* by choosing the features based on index location as the selected features are stored in the X and y.

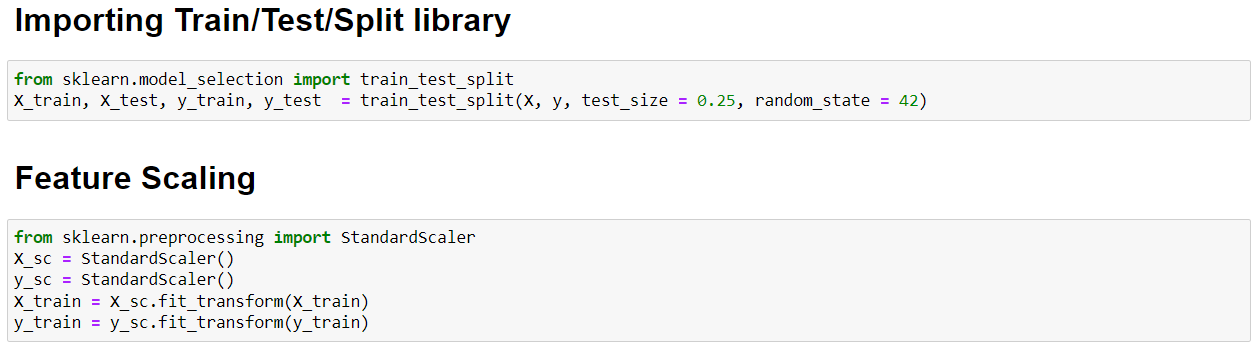


Figure 14. Train test split and feature scaling.

After importing the dataset and main dependencies, the next step is to split the data for the purpose of training and testing. For this project, the dataset is split into 75% data for training and 25% data for testing. Also, the random state parameter in the *train\_test\_split* has been set to 42 by reason of avoiding the code generating a random test set as shown in Figure 14. The next step is to implement feature scaling and use the training samples using the *.fit\_transform()* to learn the scaling of the parameters on that data.

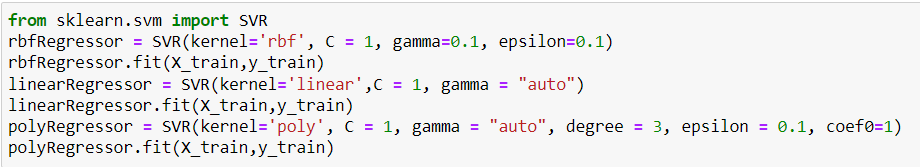


Figure 15. Fitting the kernels into training data.

Before predicting the result, the SVR was imported and the three kernels were created by implementing the *SVR()* method. Before fitting the kernels, the parameter for each kernel must be set to improve the performance of the model. The parameters that were used in the kernel are *C, epsilon, gamma, degree, and coef0* for the polynomial kernel. Then, fitting the three kernels into the training data and the SVR model was created as shown in Figure 15.

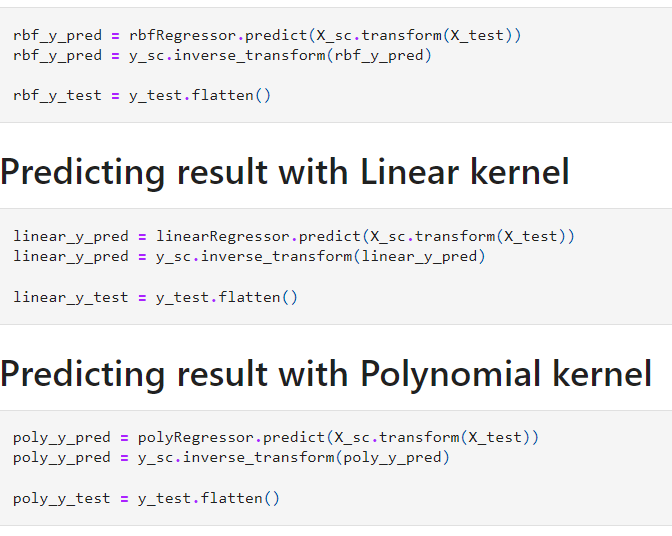


Figure 16. Predict new results for each kernel.

After creating the model, the results from the three kernels are predicted with the *predict()* function, and the scaled data were transformed into the original data using the *.inverse\_transform()* function. Hence, the *.flatten()* function was applied to transform the data into one-dimensional data as shown in Figure 16.

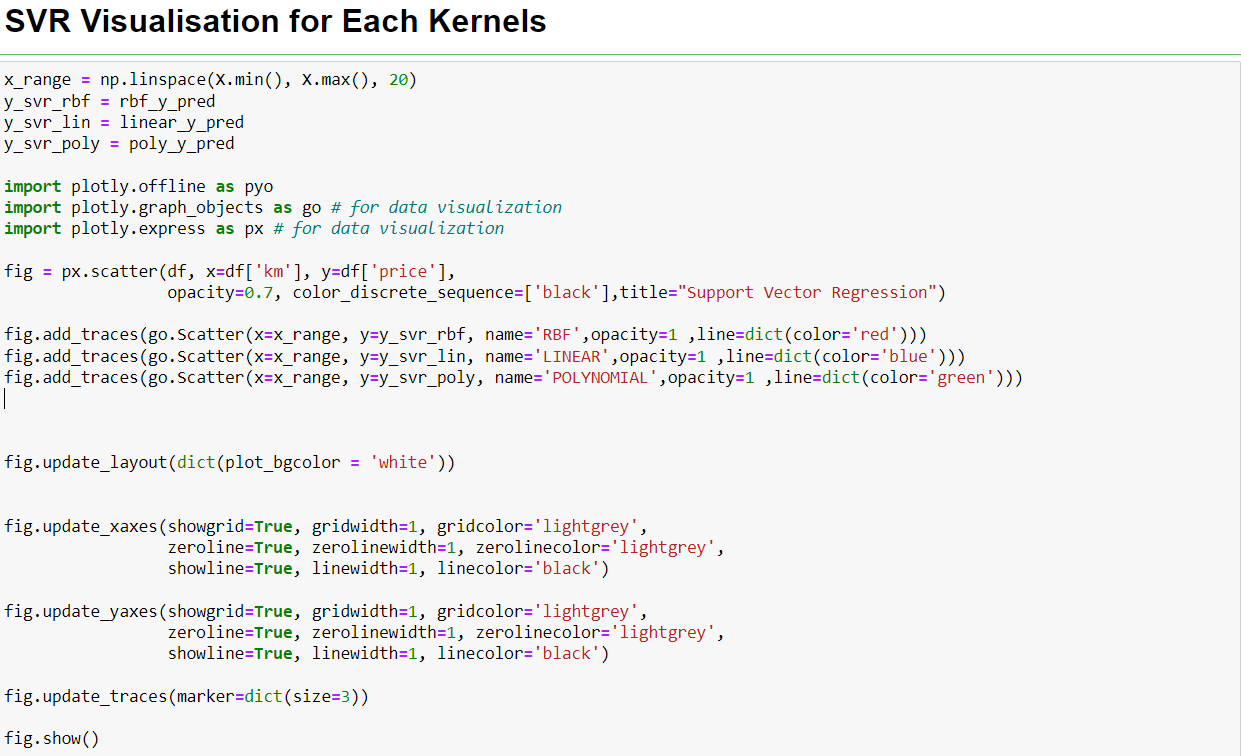


Figure 17. Plotting the SVR using Plotly.

In Figure 17, the visualisation for the SVR model is using the plotly library due to its comprehensive features using JavaScript makes the user analyze the output more clearly.

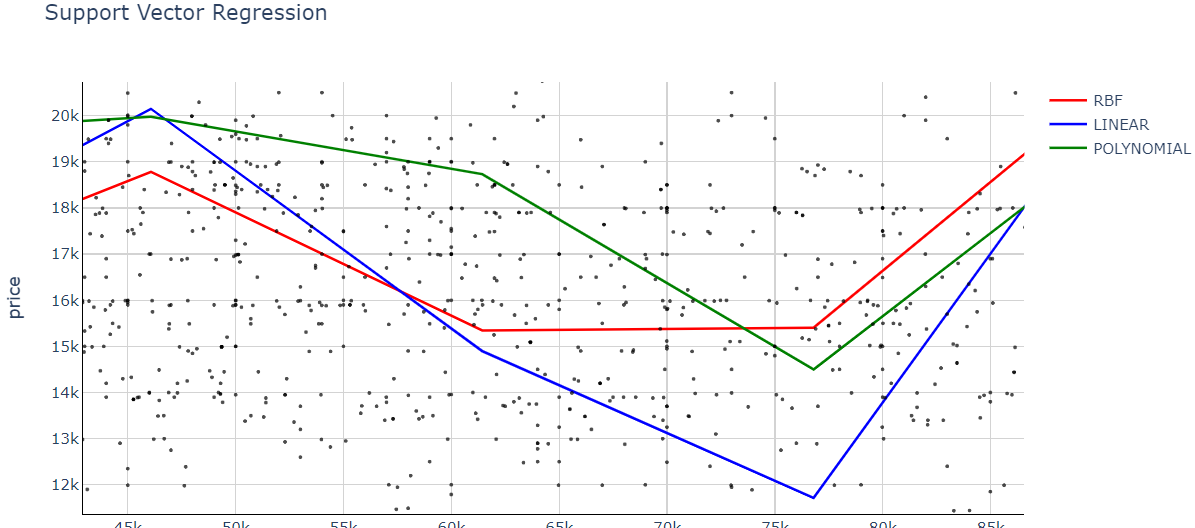


Figure 18. Visualization of SVR from three kernels.

The regression line for the SVR model is presented as shown in Figure 18. The red-colored line represents the RBF kernel, the blue-colored line represents the linear kernel, and the green-colored line represents the polynomial kernel. In addition, the scatter plot is representing the correlation of the data in *price* and *km* features.

## Linear Regression

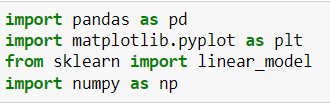


Figure 19. Importing dependencies for linear regression.

The first step for the linear regression model is to import the important dependencies as shown in Figure 19. In this model, Pandas library and NumPy library were implemented due to the function that serves this data-related project.

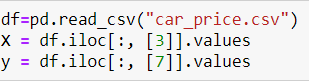


Figure 20. Importing dataset and storing features into variables.

In Figure 20, after importing the important dependencies, the data is imported and stored in the *df* variable. Since the objective of this project is to compare both models, the same features were implemented to achieve the objective which are *km* and *price* features.

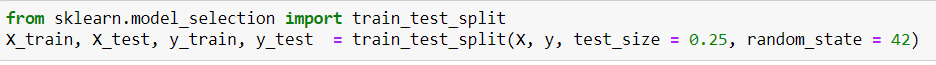


Figure 21. Importing *train\_test\_split.*

The next step is to import the *train\_test\_split* library and implement it. The *test\_size* is 0.25 signify that the data used for testing is 25% of the data and the rest of the data is used for training. In addition, the *random\_state* parameter was set to 42 to avoid random sampling each time the code is running as shown in Figure 21.

After splitting the data, feature scaling was implemented in this model. The *inverse\_transform()* function was also implemented to revert the scaled data into the actual data as shown in the Figure below.

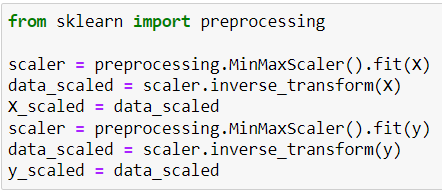


Figure 22. Feature Scaling using *StandardScaler()*.

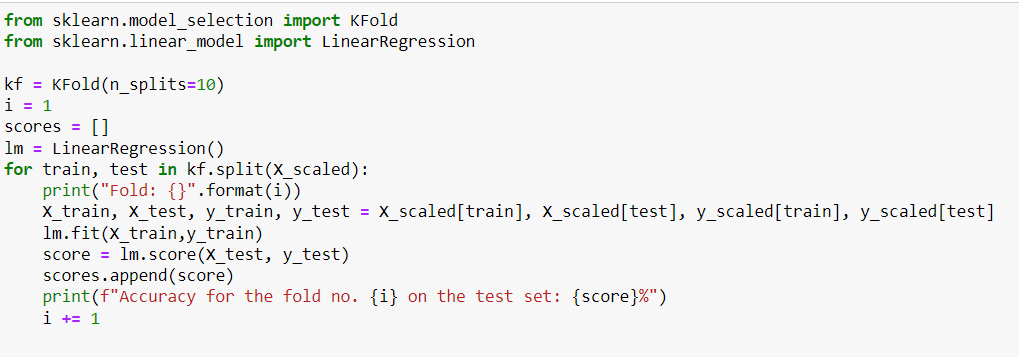


Figure 23. Implementing K-Fold cross-validation.

In this model, cross-validation was implemented to calculate the ability of this model toward unseen data. The first step is to import the K-Fold and import the linear regression model from the *sklearn* library. After importing the important dependencies, the *kf* variable was created to define the number of folds in the cross-validation ( *n\_splits = 10)* as shown in Figure 23, the number of folds is set to 10. In order to store the scores from the cross-validation, an array called *scores* was created. In the wake of implementing the necessary process for the cross-validation, an iteration was also implemented to print the accuracy from the training process in the cross-validation and fit the model into the data.

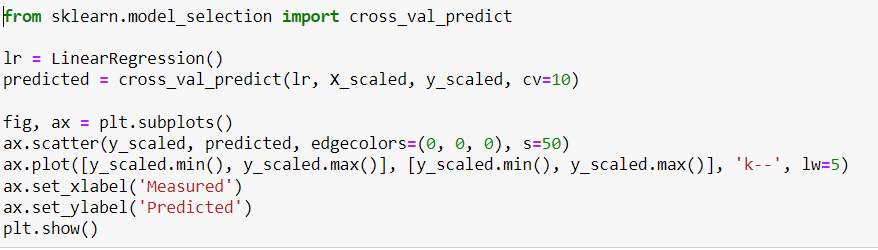


Figure 24. Importing and implementing the *cross\_val\_predict* package.

In order to predict the result from the cross-validation, this model used the *cross\_val\_predict* from *sklearn* to perform a prediction. Furthermore, the visualisation of linear regression was also performed in this section using the *matplotlib* library as shown in Figure 24.

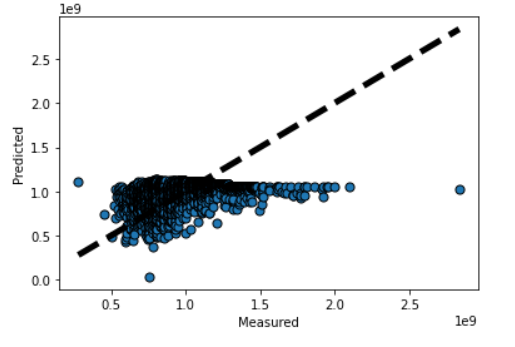


Figure 25. The linear regression visualisation.

Figure 25 shows the visualisation of the linear regression. The blue-colored scattered plot represents the dataset and the black-colored line is the representation of the regression line. To clarify, the dataset is not moving towards the regression line although some data points fit the linear regression line.

# Results

The visual presentation of linear regression and SVR models projected a relatively smooth prediction. Hence, the measurement of the performance of both approaches is represented by metrics scores such as RMSE and MSE as discussed in the methodology section. The metrics were imported from the *sklearn* library that contains the *metrics* package for the necessary scoring metrics. The code to perform the calculation for scoring metrics can be seen in Figure 26 below.

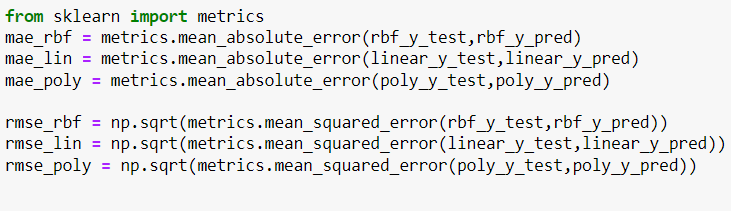


Figure 26. Import metrics for SVR scoring.

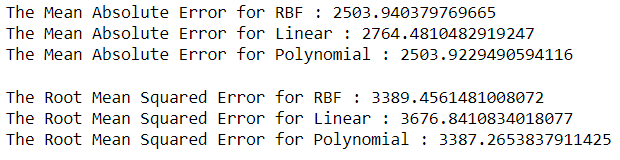


Figure 27. SVR performance measurements.

The result seen in Figure 27 for the support vector model presents an assumption that the polynomial kernel performs marginally close to the RBF kernel with a significant difference. Meanwhile, the linear kernel performs the worst amongst the three kernels in mean absolute error metrics (MAE). As for the root mean squared error (RMSE) metrics, the scores provide an assumption that the polynomial kernel performs better than the other two kernels.

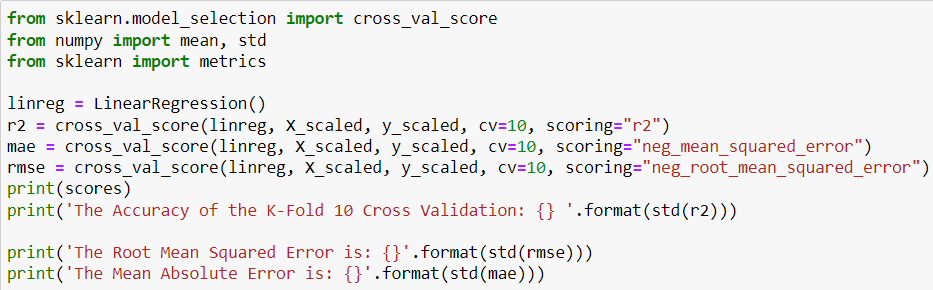


Figure 28. Import the *cross\_val\_score* package and calculate the linear regression score.

Since the linear regression was using the cross-validation method, the scoring metrics from *cross\_val\_score* were imported from a different package in the *sklearn* library. The *cross\_val\_score()* function has a parameter called scoring to determine the metrics score. For each scoring metric, the variable *mae* and *rmse* represent the scoring metrics inside the function. Meanwhile, the *cv* parameter determines the number of splits in the cross-validation which means 10 splits (cv= 10) as shown in Figure 28.

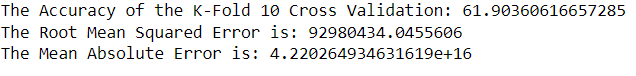


Figure 29. Linear regression performance measurements.

The scoring metrics for the linear regression model that were implemented in this project are root mean squared error (RMSE) and mean absolute error (MAE). The RMSE for this model is inferior compared to the SVR model considering the linear regression model implemented the cross-validation. Although the RMSE is inferior, the MAE of this model is superior to the SVR. To conclude this project based on hypothesis, the SVR and linear regression have their own advantages in this project as seen in the Figure.

# Summary and Reflection

## Project Management

This project was planned to be completed in a sequential manner, as the starting point of this project is data collection and refinements completed before designing and implementing the model. Since the original chart was made at the early stage, there are some changes to the schedule that need to be done for this project. In order to proceed with the machine learning stage, the objective that needs to be accomplished before applying the machine learning was to do all the literature reviews to find different insights into the models that were implemented in this project. After the data collection, data refinements, and literature review were done, the next process was to write the interim report and design both models. There were some changes in model selection and writing the dissertation took longer than the expected time as shown in the original chart (Figure 30) and the updated chart (Figure 31). In order to maximize the dissertation, the were some addition for reviewing and implementing given feedback from the supervisor.

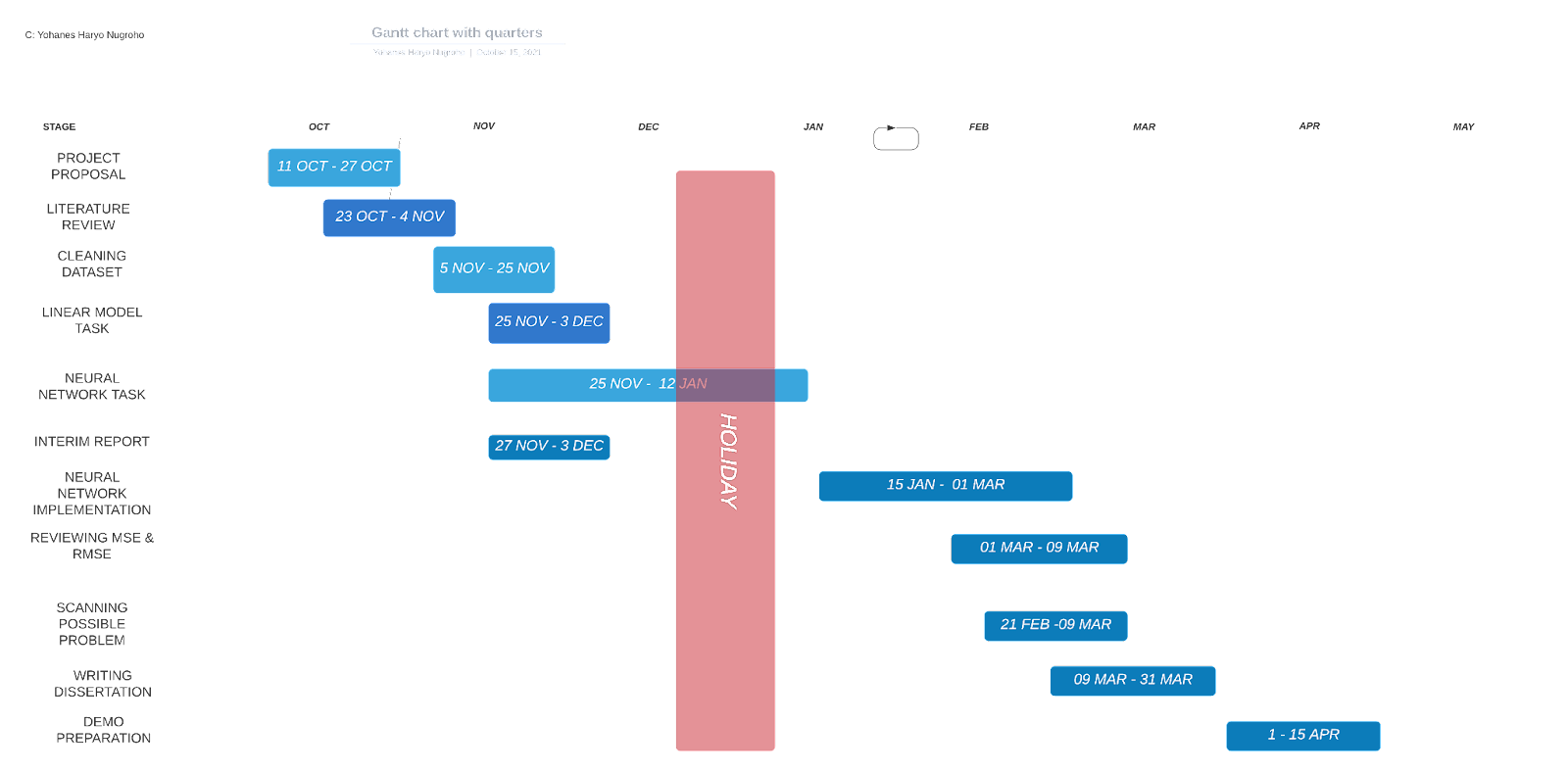


Figure 30. Original Gantt Chart.

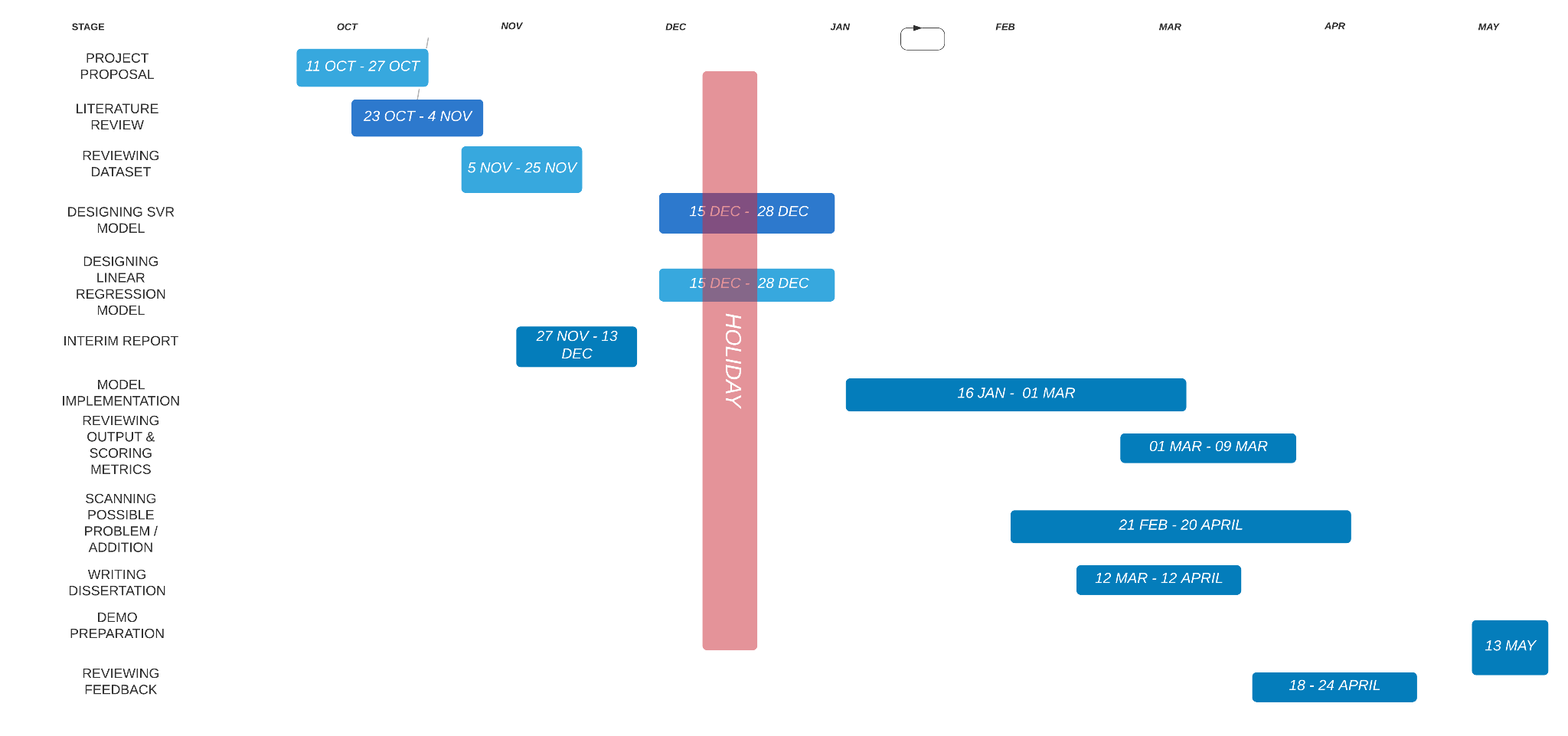


Figure 31. Updated Gantt chart.

## Contribution and Reflections

During the process of designing and implementing the models that are used in this project, there were some obstacles that need more attention to solve such as selecting the models to compare and feature selection. Moreover, as stated in the related work of Listiani that in their SVR model, the best performance kernel is the RBF kernel. Meanwhile, in this project, the best performance kernel is the polynomial kernel. The output of cross-validation in linear regression results was unexpected, assuming the cross-validation can improve the model toward marginal error in the linear regression. Although, one metric in the linear regression performs better than SVR as shown in the results section. Overall, the satisfaction level for this project is good, considering the length of time given to complete this project is long. Also, some changes that were made during the process of designing the model took a while to be decided and implemented as the right model. In addition, there were changes in the methodology section and related work section as the guidance from the supervisor’s feedback.

## Limitation

There are some limitations during the work of this project besides the author's satisfaction level. The project was held back for 2 weeks due to a covid-19 situation that affected the author to completion of the project. The second limitation of this project is the difficulty of the machine learning field since it is a whole new field for the author with high computational ability in order to perform in this field as one of the requirements. The last obstacle for this project was finding the right literature that fits the most with this project considering the vast variety of options on the internet and implementing the findings from previous work that has been done.

## Future Works

There are things that can be improved for this project in the future. It will be more challenging if there is a specific dataset with the time stamps for each model sales from a specific manufacturer to change the course of this project into a time-series prediction. In addition, the new model can also be implemented and the sample size can be improved.

# Bibliography

[1] Ö. Çelik and U. Osmanoğlu, "Prediction of The Prices of Second-Hand Cars", *European Journal of Science and Technology*, pp. 77-83, 2019. Available: 10.31590/ejosat.542884.

[2] D. Genesove, "Adverse Selection in The Wholesale Used Car Market", *Journal of Political Economy*, vol. 101, no. 4, pp. 644-665, 1993.

[3] C. Threewitt, "10 Factors That Affect Your Car's Resale Value", *HowStuffWorks*, 2022. [Online]. Available: https://auto.howstuffworks.com/buying-selling/car-resale-value.htm.

[4] M. Listiani, "Support Vector Regression Analysis for Price Prediction in a Car Leasing Application", 2009.

[5] C. Jin, "Price Prediction of Used Cars Using Machine Learning," 2021 IEEE International Conference on Emergency Science and Information Technology (ICESIT), 2021. Available: 10.1109/ICESIT53460.2021.9696839.

[6] M. Satioglu, Y. Ar and B. Tugrul, "Automobile Price Prediction in Turkey Marketplace with Linear Regression", *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2021. Available: 10.1109/ismsit52890.2021.9604688.

[7] C. Cortes and V. Vapnik, "Support-vector networks", *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995. Available: 10.1007/bf00994018

[8] A. Raj, "Unlocking the True Power of Support Vector Regression", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/unlocking-the-true-power-of-support-vector-regression-847fd123a4a0.

[9] D. K, "Support Vector Regression in Python Using Scikit-Learn", *Medium*, 2020. [Online]. Available: https://heartbeat.comet.ml/support-vector-regression-in-python-using-scikit-learn-89cc18e933b7.

[10] S. Dye, "An Intro to Kernels", *Medium*, 2019. [Online]. Available: https://towardsdatascience.com/an-intro-to-kernels-9ff6c6a6a8dc.

[11] S. Sreenivasa, "Radial Basis Function (RBF) Kernel: The Go-To Kernel", *Medium*, 2020. [Online]. Available: https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a.

[12] P. Grant, "Understanding the Fundamentals of Linear Regression", *Medium*, 2019. [Online]. Available: https://towardsdatascience.com/understanding-the-fundamentals-of-linear-regression-7e64afd614e1.

[13] S. Swaminathan, "Linear Regression — Detailed View", *Medium*, 2018. [Online]. Available: https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86.

[14] J. Moody, "What does RMSE really mean?", *Medium*, 2019. [Online]. Available: https://towardsdatascience.com/what-does-rmse-really-mean-806b65f2e48e.

[15] T. Elliot, "The State of the Octoverse: machine learning", *GitHub*, 2019. [Online]. Available: https://github blog/2019-01-24-the-state-of-the-octoverse-machine-learning/, 2019.