# AMS 325 Computing and Programming Fundamentals

# Final Project

# Car Price Prediction Report

**Team member**: Skyler Aliya

Hongcheng Qian

## Project Objectives

Introduction:

We chose the Car Price Prediction project as our final project. This is an open-resource dataset about used car information that was provided on Kaggle by www.cardekho.com. It contained 3 .csv files and we only used *car data.csv* as our dataset to do the further and deep study. The dataset has nine variables: *Car_Name, Year, Selling_Price, Present_Price, Kms_Driven, Fuel_Type, Seller_Type, Transmission, and Owner*. This data could be used for multiple purposes. In the last few lectures, we have been introduced to data analysis on different topics using Python libraries, such as pandas to structure and use data, Matplotlib for plotting, and Seaborn for drawing attractive and informative statistical graphics. As well as using Python to fit data in linear regression and machine learning of statistical models to predict. The project is perfectly matched by applying all the tools and techniques that we learned in class to a real-life problem.

Goal:

Our goal is to use the given data to do the data analysis visualization to find the correlation between variables and to exemplify the use of linear regression in Machine Learning for price prediction. We decided to divide the project into 4 parts: Exploratory data analysis

(EDA), Visualization, Preparing data for training and testing, and Linear regression. Hongcheng Qian, who has less coding experience, is responsible for Exploratory data analysis and Preparing data for training and testing which are the start of the entire project and provide us with an organized and solid data foundation. Skyler aliya, who has relatively more data analysis experience, is in charge of Visualization and Linear Regression which provide the data result for observation.

## Techniques and Tools

Python Library:

In this project, we used Python and Python libraries to implement. We use `pandas` to read the *csv.* file and acknowledge the basic information of the dataset, `numpy` to transform the dataset, `matplotlib` to plot the correlation between variables, `mpl_toolkits` for the 3D plotting, and `sklearn` for the regression models.

Methods:

**Read Dataset(By Hongcheng):**

We first downloaded the data from Kaggle and chose to use Python as the analysis tool. The dataset(picture 1) with a size of (301, 9) and no missing values. Considering the "*Year*" variable would not be the best choice for further analysis, we modified the "*Year*" to "*Car_age*" by calculating '*2022 - Year'* with .drop() (picture 2). Using `.info()` shows some variables have an object data type, which is not appropriate for analysis and needs to change.

**Visualization(By Skyler):**

Secondly, we want to see the visualization of the entire dataset. In order to know more about the relationship between variables, we find the pairwise correlation of all columns in the data frame. One choice was the heatmap (picture 3), a graphical representation of data where each value of a matrix is represented as a color. After we get the heatmap, we think the result it shows is not comprehensive enough, thus we made the pair plot as well. With the pair plot, it was pretty straightforward to see the linear relationship between *"Selling Price"* and *"Present Price"*. To show more details of it, we scatter the selling price - present price diagram (picture 4). 3D-plot is widely used in data visualization, hence we applied a 3D plot of the selling price, present price, and KMS-driven (picture 5).

**Prepare Data(By Hongcheng):**

Thirdly, we converted the string variable to *int* variable, so that the algorithms can understand the codes and can be used to make operations. In statistics and machine learning, we usually split our data into two subsets: training data and testing data, and fit our model on the train data, in order to make predictions on the test data. The training set contains a known output and the model learns on this data in order to be generalized to other data later on. Also, we have the test dataset in order to test our model's prediction on this subset. To implement that, we used `train_test_split(x ,y ,test_size=0.3, random_state=1)`, to set the test size to 30% of the model. `random_state=1` to make sure the random_state is fixed so that your train-test splits are always deterministic.

**Apply to Regression Model and predict(By Skyler):**

In this last part, we used three regression models to train the dataset and calculated the R-squared and Cross-validation to compare the models. We made a `models` function to compute each regression in terms of fixed training and testing datasets. The `models` function gives the output of R-squared, Cross-Validation, the mean of Cross-Validation, and the original vs. prediction graph. `LinearRegression()`, `DecisionTreeRegressor()`, and `RandomForestRegressor(n_estimators = 100, random_state = 42)` were done to apply the data (pictures 6, 7, 8). To make a better comparison, we added a synthesized model-score data frame (pictures 9) to check each one of the scores and make the choice of the regression to predict. Selecting Decision Tree Regression as our final regression, we can use the model to predict some used-car prices by some given values. Here, we made a data frame of the number of owners, the age of the car, and the kilometers driven by cars. Choice 1 car had 3 owners, age of 3 and 70000 KMS driven. Choice 2 car had 1 owner, age of 7 and 125500 KMS driven. Choice 3 car had 2 owners, age of 5, and 35000 KMS driven.

## Observations and Conclusions

Observations:

Through the visualizations part, we found the strongest correlation between the selling price and present price with a score of 0.88 (picture 3), and by separately scattering the selling price and present price, we can even see a more linear relationship between the two variables (picture 4). We can make a conclusion by observing the 3D plot that most of the cars accumulated around age 4 to 10, low present price, and low kilometer drove (picture 5). They are correlated variables.

In the regression analysis, the first model we used is the Simple Linear regression (picture 6). The result we got from it is:

- `r_2 score: 0.8466064262307118`
- `CV scores: [0.91513983 0.8949148  0.82525457 0.82190804 0.72193337]`
- `CV scores mean: 0.835830120801873.`

The second model we used is Decision Tree Regression(picture 7). It breaks down a dataset into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is:

- r_2 score: 0.9474462622726929
- CV scores: [0.82905078 0.87105829 0.90811801 0.90673507 0.4731159 ]
- CV scores mean: 0.7976156092113967.

The third model we used is Random Forest Regression (picture 8). It is a supervised learning algorithm that uses ensemble learning methods for regression. The ensemble learning method is a technique that combines predictions from multiple machine learning algorithms to make a more accurate prediction than a single model. The result is:

- r_2 score: 0.8871735211528714
- CV scores: [0.94153328 0.97022741 0.82608763 0.94484468 0.7348586 ]
- CV scores mean: 0.8835103210057318

After comparing the three generated original vs. prediction graphs of each regression, we can see the blue line of predictions all looks fitting to the orange line of originals, all those three regression models have a high R-squared value, which indicates that our training results are very close to reality. However, comparing the score of regressions (picture 9), we can tell that the Decision Tree Regression has the highest R-squared value and lowest Cross-Validation mean value. It can be concluded that the Decision Tree Regression is the best fit for our dataset and we could use it to predict the price of used cars.

## Applications:

Think about a scenario: you are comparing 3 cars on your wishlist and not sure if the price that the dealer provided is accurate. This is the time to use our Used-car Price Prediction tool to assist with the pricing. By putting the number of owners, age of the car, and kms_driven, it gives you a predicted price of those 3 cars. The predicted prices of each choice are 11.25, 4.90, and 7.75. Now you know the appropriate price for your dream car and be able to make a more reasonable choice as well.

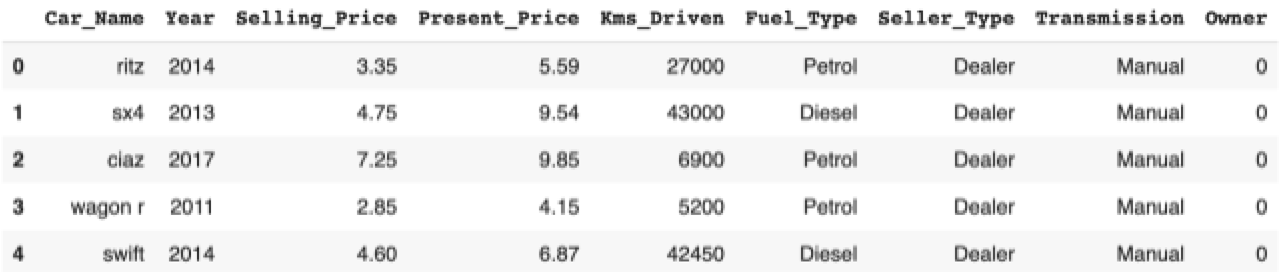
Conclusions:

In conclusion, the present price of a car plays an important role in predicting the Selling Price. One increases the other gradually increases. Car age is affecting negatively as older cars lower the Selling Price. Kms driven is the most correlated variable for the price prediction.

The code of this project is available on Github.

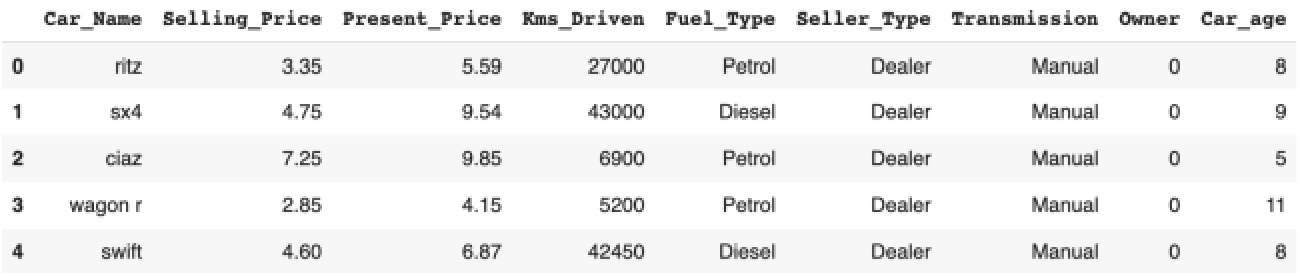
# Graphics and References

## Graphics:



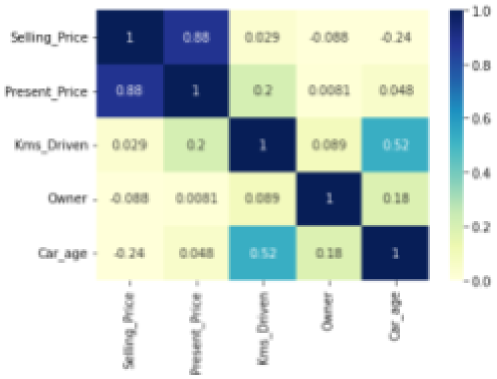
| | Car_Name | Year | Selling_Price | Present_Price | Kms_Driven | Fuel_Type | Seller_Type | Transmission | Owner |
|---|---|---|---|---|---|---|---|---|---|
| 0 | ritz | 2014 | 3.35 | 5.59 | 27000 | Petrol | Dealer | Manual | 0 |
| 1 | sx4 | 2013 | 4.75 | 9.54 | 43000 | Diesel | Dealer | Manual | 0 |
| 2 | ciaz | 2017 | 7.25 | 9.85 | 6900 | Petrol | Dealer | Manual | 0 |
| 3 | wagon r | 2011 | 2.85 | 4.15 | 5200 | Petrol | Dealer | Manual | 0 |
| 4 | swift | 2014 | 4.60 | 6.87 | 42450 | Diesel | Dealer | Manual | 0 |

Picture 1: Read file data

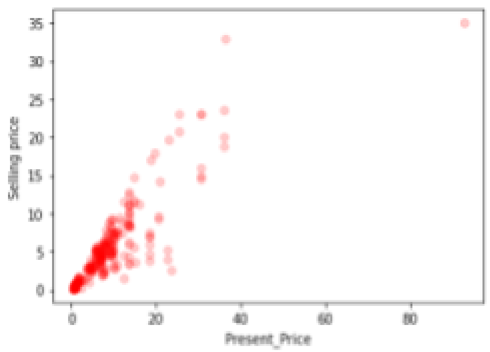


| | Car_Name | Selling_Price | Present_Price | Kms_Driven | Fuel_Type | Seller_Type | Transmission | Owner | Car_age |
|---|---|---|---|---|---|---|---|---|---|
| 0 | ritz | 3.35 | 5.59 | 27000 | Petrol | Dealer | Manual | 0 | 8 |
| 1 | sx4 | 4.75 | 9.54 | 43000 | Diesel | Dealer | Manual | 0 | 9 |
| 2 | ciaz | 7.25 | 9.85 | 6900 | Petrol | Dealer | Manual | 0 | 5 |
| 3 | wagon r | 2.85 | 4.15 | 5200 | Petrol | Dealer | Manual | 0 | 11 |
| 4 | swift | 4.60 | 6.87 | 42450 | Diesel | Dealer | Manual | 0 | 8 |

Picture 2: Car_age added result

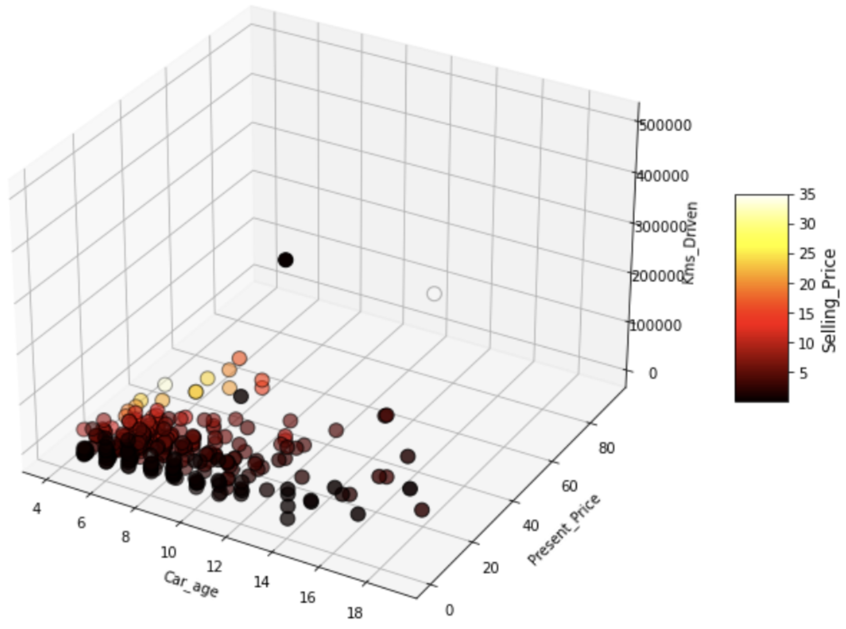


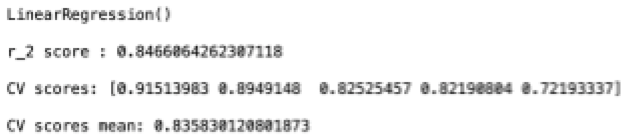
Picture 3: heatmap



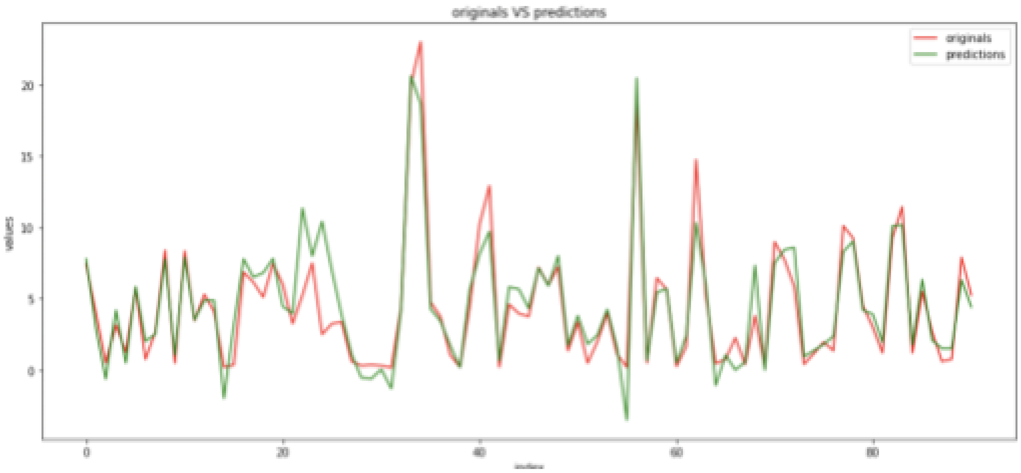
Picture 4:Selling Price Vs. Present Price

5. 3D plot for Car age, Present price and Kms driven

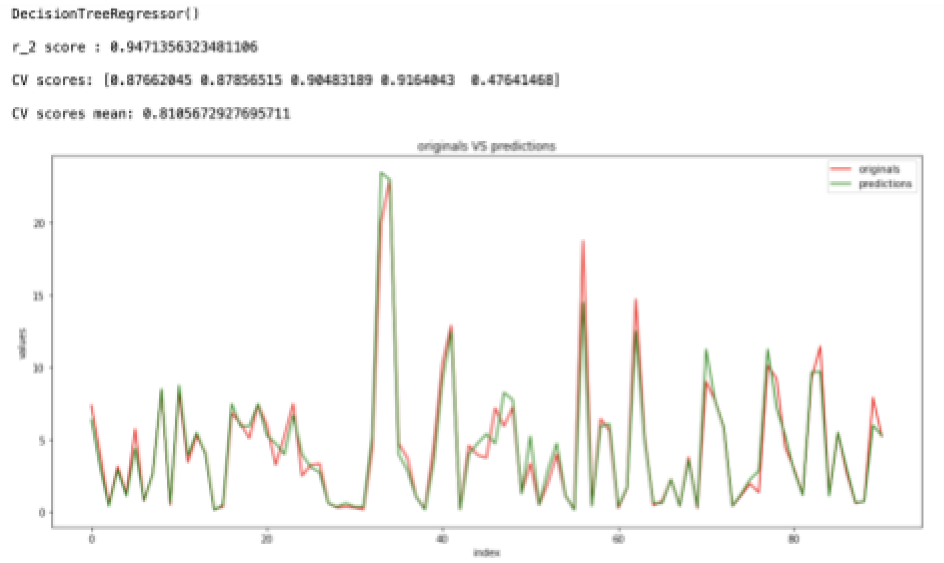




```
LinearRegression()

r_2 score : 0.8466064262307118

CV scores: [0.91513983 0.8949148  0.82525457 0.82190804 0.72193337]

CV scores mean: 0.835830120801873
```

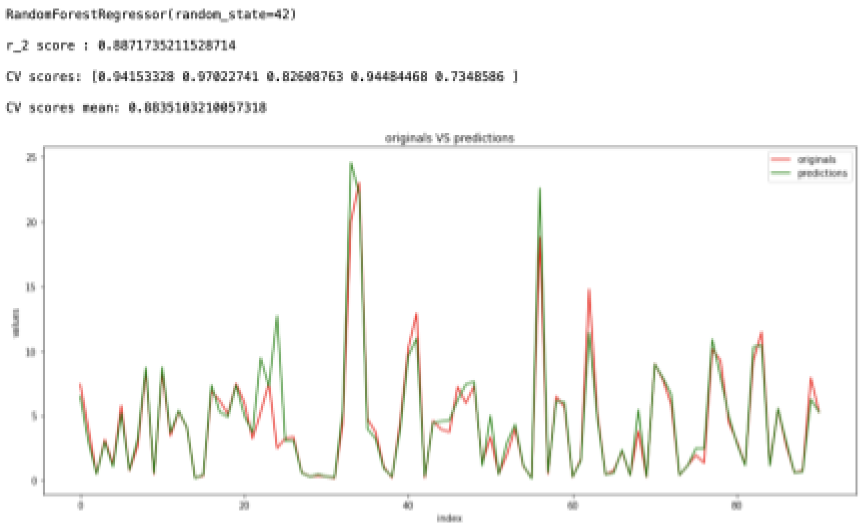


Picture 6: Linear Regression



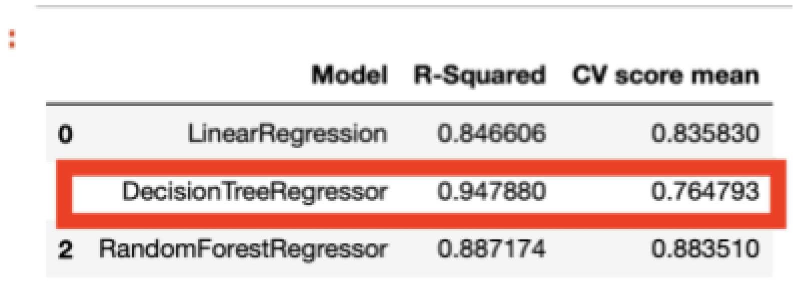
```
DecisionTreeRegressor()

r_2 score : 0.9471356323481106

CV scores: [0.87662045 0.87856515 0.90483189 0.9164043  0.47641468]

CV scores mean: 0.8105672927695711
```

Picture 7: Decision Tree Regression



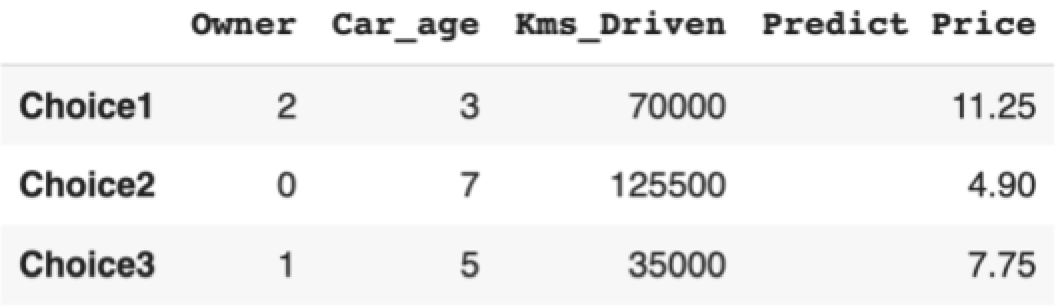
```
RandomForestRegressor(random_state=42)

r_2 score : 0.8871735211528714

CV scores: [0.94153328 0.97022741 0.82608763 0.94484468 0.7348586 ]

CV scores mean: 0.8835103210057318
```

8. Random Forest Regression



| | Model | R-Squared | CV score mean |
|---|---|---|---|
| 0 | LinearRegression | 0.846606 | 0.835830 |
| | DecisionTreeRegressor | 0.947880 | 0.764793 |
| 2 | RandomForestRegressor | 0.887174 | 0.883510 |

Picture 9: Scores of Models



| | Owner | Car_age | Kms_Driven | Predict Price |
|---|---|---|---|---|
| Choice1 | 2 | 3 | 70000 | 11.25 |
| Choice2 | 0 | 7 | 125500 | 4.90 |
| Choice3 | 1 | 5 | 35000 | 7.75 |

Picture 10: Prediction

**References:**

1. https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho
2. https://www.kaggle.com/code/kanncaa1/data-sciencetutorial-for-beginners/notebook
3. https://seaborn.pydata.org/generated/seaborn.heatmap.html
4. https://scikit-learn.org/0.17/modules/generated/sklearn.tree.DecisionTreeRegressor.html