CAR RESALE VALUE PREDICTION USING MACHINE LEARNING ALGORITHMS

By

MADDUKURI NIVAS – 19BCE1010

KOVVURI UDAY SURYA DEVESWAR REDDY – 19BCE1253

A project report submitted to

Dr. SALEENA B.

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

in partial fulfilment of the requirements for the course of

CSE2003 – DATA STRUCTURES AND ALGORITHMS

In

B.Tech. COMPUTER SCIENCE ENGINEERING

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

VIT CHENNAI

Vandalur – Kelambakkam Road

Chennai – 600127

NOVEMBER 2020

INDEX

1. BONAFIDE CERTIFICATE

Certified that this project report entitled "**CAR RESALE VALUE PREDICTION USING MACHINE LEARNING ALGORITHMS**" is a bonafide work of **MADDUKURI NIVAS (19BCE1010)** and **KOVVURI UDAY SURYA DEVESWAR REDDY (19BCE1253)** who carried out the Project work under my supervision and guidance.

Dr. SALEENA B

Associate Professor

School of Computing Science and Engineering (SCSE),

VIT University, Chennai

Chennai – 600127.

2. ABSTRACT

The main purpose of this project is to predict the car price by using different regression algorithms. In our daily process at some time if we want to buy a car in a second hand for learning purpose this is one of the way to predict the price for the car by using different algorithms. In this project the algorithms we are using to predict the values are support vector regression algorithm, random forest regression algorithm. So, as we are using different algorithms, in practical terms they have their advantages and disadvantages. So by implementing the algorithms the prediction values will be shown in graph format and we will find the accuracy rate of the algorithms so by using these accuracy rate we can find out which algorithm is best among all for prediction.

We will conclude with our final outcome that is to predict the car resale values and the which algorithm is best for prediction of car resale value. Hence we will use random forest regression algorithm and support vector regression algorithm to get the accurate price of the resale cars. These algorithms are machine learning based algorithms.

3. ACKNOWLEDGEMENT

We are thankful to **Dr. Jagadeesh Kannan R**, Dean of the School of Computer Science Engineering, VIT Chennai, for extending the facilities of the School towards our project and for his beneficent support.

We express our thanks to our Programme Chair **Dr. Justus S** for his support throughout the course of this project.

I respect and thank **Dr. Saleena B,** School of Computing Science and Engineering for providing us an opportunity to do the project work and giving us all support and guidance which made me complete the project duly. I am extremely thankful for providing such a nice support and guidance, although he had busy schedule managing the corporate affairs.

I am thankful to and fortunate enough to get constant encouragement, support and guidance from all Teaching staffs of School of Computer Science and Engineering which helped us in successfully completing our project work. Also, I would like to extend our sincere esteems to all staff in laboratory for their timely support.

Secondly, I would also like to thank my parents and friends who helped me a lot in finalizing this project within the limited time frame.

MADDUKURI NIVAS KOVVURI UDAY SURYA DEVESWAR REDDY

4. INTRODUCTION

What is machine learning? How it is used in this project?

Machine Learning is a core sub-area of Artificial Intelligence (AI). ML applications learn from experience (well data) like humans without direct programming. When exposed to new data, these applications learn, grow, change, and develop by themselves. In other words, with Machine Learning, computers find insightful information without being told where to look. Instead, they do this by leveraging algorithms that learn from data in an iterative process.

We used ML in this project by using Support Vector Regression and Random Forest Regression Algorithm.

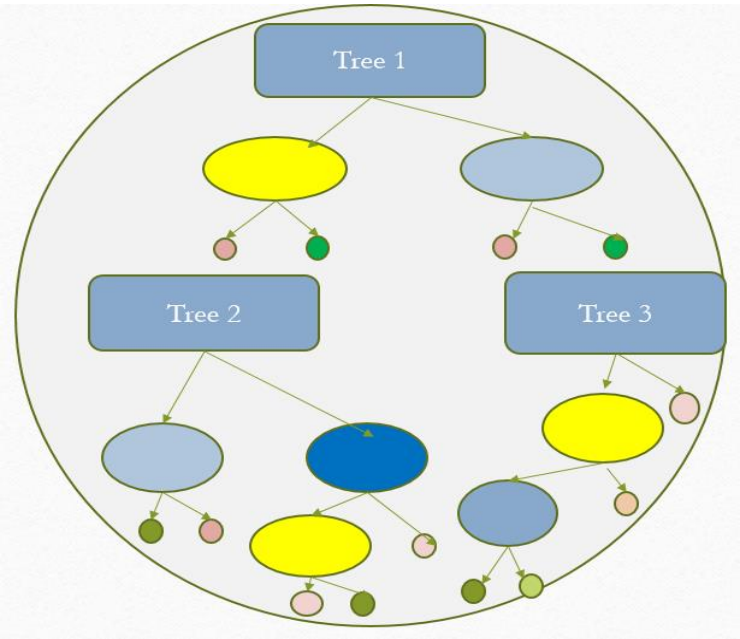
Random forest Regression

The Random Forest is one of the most effective machine learning models for predictive analytics, making it an industrial workhorse for machine learning.

The random forest model is a type of additive model that makes predictions by combining decisions from a sequence of base models. More formally we can write this class of models as:

$$g(x) = f_0(x) + f_1(x) + f_2(x) + \ldots$$

Where the final model g is the sum of simple base models f_i. Here, each base classifier is a simple decision tree. This broad technique of using multiple models to obtain better predictive performance is called model ensembling. In random forests, all the base models are constructed independently using a different subsample of the data.



For Example:

Prediction value of Tree 1=1.2

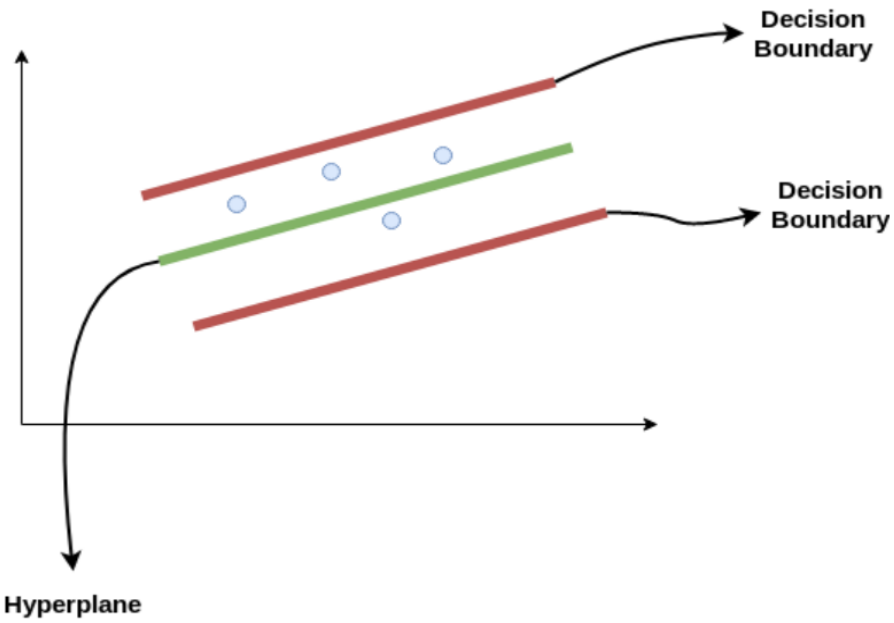
Prediction value of Tree 2=1.3

Prediction value of Tree 3 =1.4

Predicted value of random forest=1.2+1.3+1.4/3=1.3

Support vector Regression

The problem of regression is to find a function that approximates mapping from an input domain to real numbers on the basis of a training sample. Then let us see what support vector regression is:



Consider these two red lines as the decision boundary and the green line as the hyperplane. Our objective, when we are moving on with SVR, is to basically consider the points that are within the decision boundary line. Our best fit line is the hyperplane that has a maximum number of points.

The first thing that we'll understand is what is the decision boundary (the danger red line above!). Consider these lines as being at any distance, say 'a', from the hyperplane. So, these are the lines that we draw at distance '+a' and '-a' from the hyperplane. This 'a' in the text is basically referred to as epsilon.

Assuming that the equation of the hyperplane is as follows:

$y = nx + b$ (Equation of hyper plane)

Then the equations of decision boundary become:

$nx + b = +a$

$nx + b = -a$

Thus, any hyperplane that satisfies our SVR should satisfy:

$-a < y - nx + b < +a$

OBJECTIVES AND GOALS

To implement the algorithm to predict the car resale value from the history of the dataset.

- Fitting the regressor to the train set using random forest regression algorithm.
- Fitting the regressor to the train set using Support vector regression algorithm.
- Calculating the accuracy scores of the algorithms.
- Predicting the test set values using the algorithms and comparing them with the actual values in the dataset.
- Plotting the graph between actual and predicted result of the test set.

5. ALGORITHM USED

RANDOM FOREST REGRESSION BASED ALGORITHM

1. Pick at random K data points from the training set.

2. Build the decision tree associated with those k data points.

3. Choose the number N tree of trees you want to build and repeat step 1 & 2.

4. For a new data point, make each one of your N tree trees predict the value of Y for the data point, and assign the new data point the average across all of the predicted Y values.

SUPPORT VECTOR REGRESSION BASED ALGORITHM

1. Identify the right hyperplane.

2. Find the hyperplane to segregate the classes.

3. Here, we will add a new feature z=x^2+y^2. Now, plot the data points on axis x and z.

4. All values for z would be positive always because z is the squared sum of both x and y.

6. ALGORITHM IMPLEMENTATION

RANDOM FOREST REGRESSION ALGORITHM

#importing libraries

```
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
import collections
from sklearn.metrics import r2_score
import re
from sklearn import ensemble
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
%matplotlib inline
```

#importing dataset

```
df_train = pd.read_excel("Data_Train.xlsx")
```

 # splitting into test and train set

#default it will take test size as 25%

```
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

 # fitting the model and calculating accuracy score

```
regr = RandomForestRegressor(max_depth=20, random_state=0, n_estimators=10)
```

```
regr.fit(X_train, y_train)
print('R-squared score (training): {:.3f}'
    .format(regr.score(X_train, y_train)))
print('R-squared score (testing): {:.3f}'
    .format(regr.score(X_test, y_test)))
```

 # predicting the test set results using the random forest regression model

```
y_pred = regr.predict(X_test)
print(y_pred);
```

 # visualizing the actual and predicting results of the test set using random forest regression model.

```
plt.scatter(y_test, y_pred, color='red')
m, b = np.polyfit(y_test, y_pred, 1)
plt.plot(y_test, m*y_test + b)
plt.title('Actual values vs predicted values using random forest')
plt.show()’
```

SUPPORT VECTOR REGRESSION ALGORITHM

#importing libraries

```
import warnings
warnings.filterwarnings("ignore")
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
import collections
from sklearn.metrics import r2_score
import re
```

```
from sklearn import ensemble

from sklearn.linear_model import LinearRegression

from sklearn.model_selection import train_test_split

from sklearn.tree import DecisionTreeRegressor

from sklearn.ensemble import RandomForestRegressor

%matplotlib inline

#importing dataset

df_train = pd.read_excel("Data_Train.xlsx")

# splitting into test and train set

#default it will take test size as 25%

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)

# fitting the model

from sklearn.svm import SVR

regressor = SVR(kernel = 'rbf')

regressor.fit(X_train, y_train)

# predicting the test set results using the support vector regression model

y_pred = regressor.predict(X_test)

print(y_pred);

# visualizing the actual and predicting results of the test set using random forest
regression model.

plt.scatter(y_test, y_pred, color='red')

m, b = np.polyfit(y_test, y_pred, 1)

plt.plot(y_test, m*y_test + b)

plt.title('Actual values vs predicted values using support vector')

plt.show()

# calculating the accuracy score of the support vector regression model

from sklearn.metrics import classification_report

regressor.score(X_test,y_test)
```

7. OUTPUT AND RESULT

FITTING AND CALCULATING THE ACCURACY SCORES USING RANDOM FOREST REGRESSION MODEL.

Random Forest Regressor Model

```
regr = RandomForestRegressor(max_depth=20, random_state=0, n_estimators=10)
regr.fit(X_train, y_train)
print('R-squared score (training): {:.3f}'
     .format(regr.score(X_train, y_train)))
print('R-squared score (testing): {:.3f}'
     .format(regr.score(X_test, y_test)))
```

```
R-squared score (training): 0.986
R-squared score (testing): 0.932
```

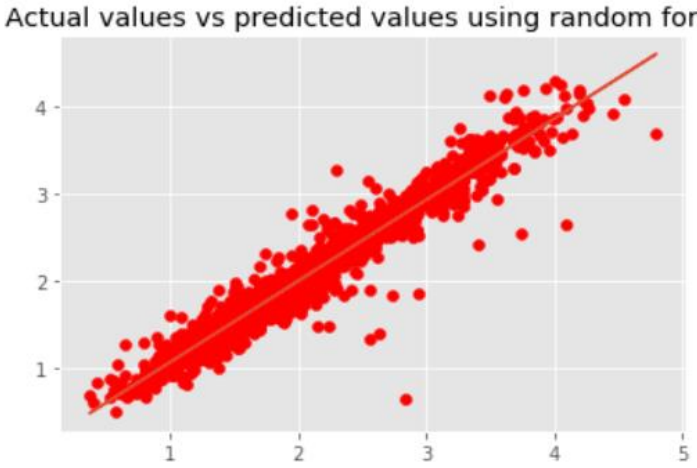
predicting test set values using random forest

```
y_pred = regr.predict(X_test)
print(y_pred);
```

```
[2.21700654 1.47785114 1.62545259 ... 2.50062231 1.90864515 1.7904186 ]
```

Visualizing actual results vs predicted results using Random forest

```
plt.scatter(y_test, y_pred, color='red')
m, b = np.polyfit(y_test, y_pred, 1)
plt.plot(y_test, m*y_test + b)
plt.title('Actual values vs predicted values using random forest')
plt.show()
```



FITTING AND CALCULATING THE ACCURACY SCORES USING SUPPORT VECTOR REGRESSION MODEL.

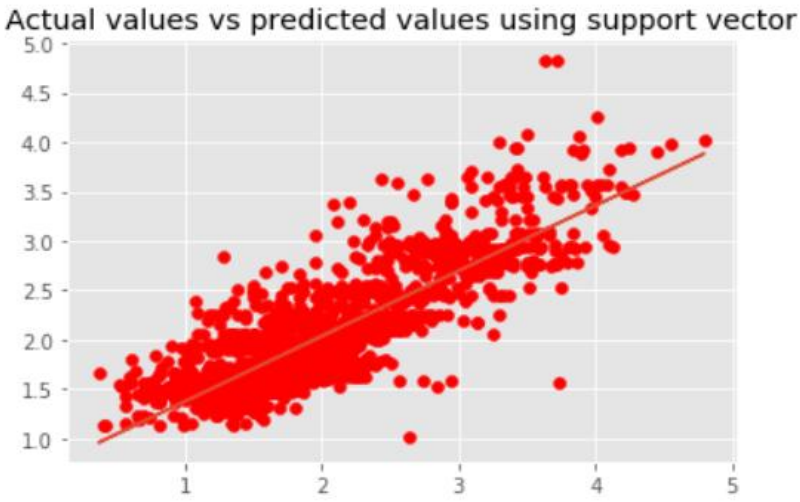
Predicting test set values using support vector

```
y_pred = regressor.predict(X_test)
print(y_pred);
```

```
[2.48783871 1.58652453 1.96866888 ... 2.2411862  1.66746063 1.67121376]
```

Visualizing actual results vs predicted results

```
plt.scatter(y_test, y_pred, color='red')
m, b = np.polyfit(y_test, y_pred, 1)
plt.plot(y_test, m*y_test + b)
plt.title('Actual values vs predicted values using support vector')
plt.show()
```



ACCURACY SCORES OF THE SUPPORT VECTOR REGRESSION

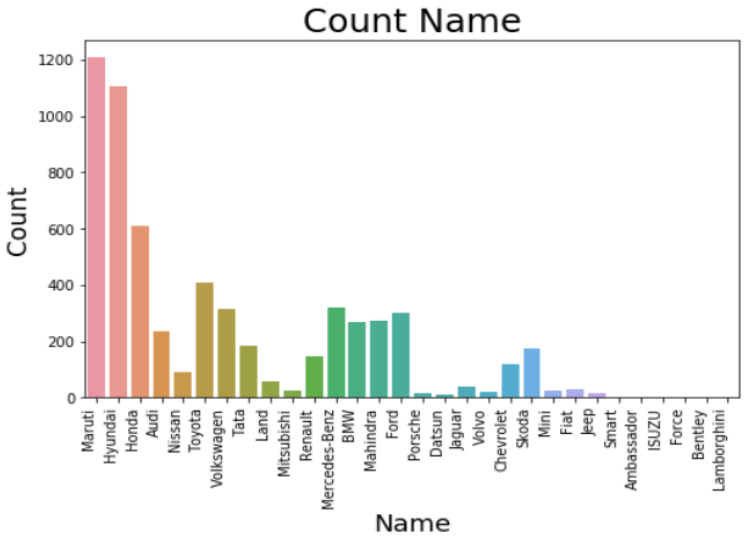
```
from sklearn.metrics import classification_report
regressor.score(X_test,y_test)
```

```
]: 0.683128821493263
```

EXPLORATORY DATA ANALYSIS

Name Type wise count

```
plt.figure(figsize = (8,5))
bar1 = sns.countplot(dataset['Name'])
bar1.set_xticklabels(bar1.get_xticklabels(), rotation = 90, ha = 'right')
plt.title('Count Name', size = 24)
plt.xlabel('Name', size = 18)
plt.ylabel('Count', size = 18)
plt.show()
```

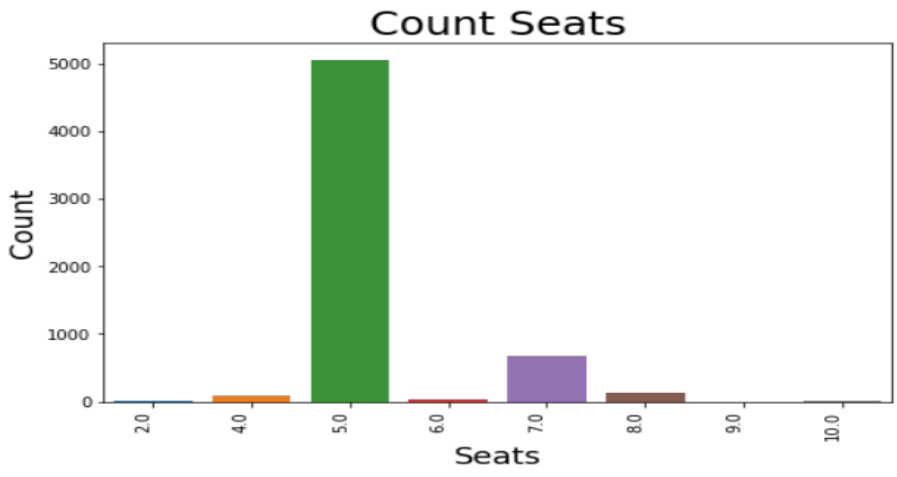


From above picture we can conclude that maruti and Hyundai are highly re saled.

Seats Type wise count

In [31]:

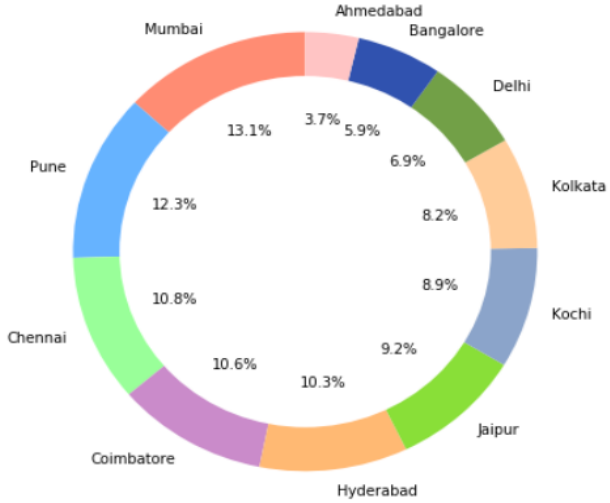
```
plt.figure(figsize = (8,5))
bar1 = sns.countplot(dataset['Seats'])
bar1.set_xticklabels(bar1.get_xticklabels(), rotation = 90, ha = 'right')
plt.title('Count Seats', size = 24)
plt.xlabel('Seats', size = 18)
plt.ylabel('Count', size = 18)
plt.show()
```



From the above graph we can conclude that cars with 5 seats are re saled more.

Loction graph and percentage

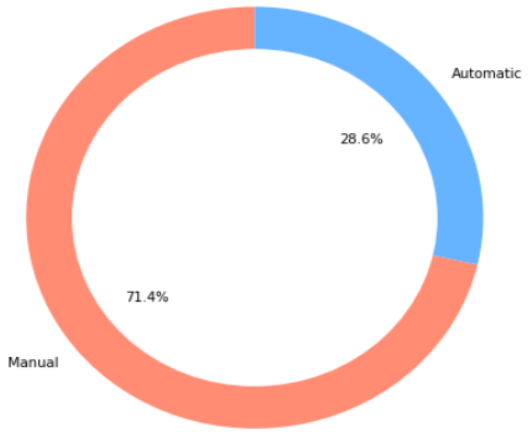
In [37]:
```
plt.figure(figsize = (6,6))
plt.pie(dataset['Location'].value_counts(), startangle = 90, autopct = '%1.1f%%', colors = colors,
        labels = dataset['Location'].unique())
centre_circle = plt.Circle((0,0),0.80,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.tight_layout()
plt.show()
```



From the above pie chart we can conclude that location Mumbai has high percentage of re sale.

Transmission graph and percentage

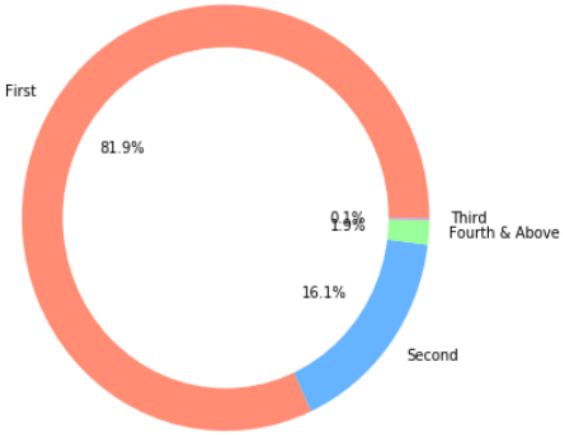
In [42]:
```
plt.figure(figsize = (6,6))
plt.pie(dataset['Transmission'].value_counts(), startangle = 90, autopct = '%1.1f%%', colors = colors,
        labels = dataset['Transmission'].unique())
centre_circle = plt.Circle((0,0),0.80,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.tight_layout()
plt.show()
```



From the above graph we can conclude that cars are re saled more by manually.

Owner_Type graph and percentage

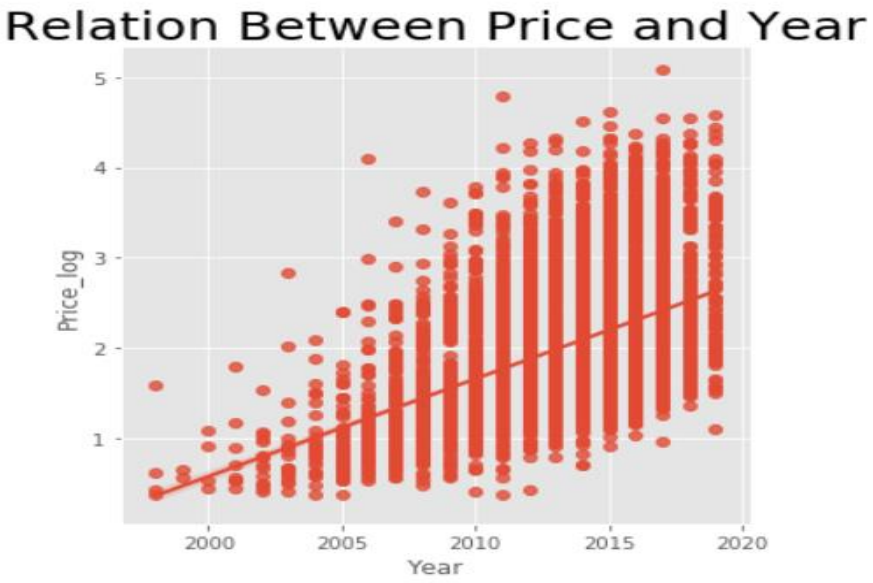
In [43]:
```
plt.figure(figsize = (6,6))
plt.pie(dataset['Owner_Type'].value_counts(), startangle = 0, autopct = '%1.1f%%', colors = colors,
        labels = dataset['Owner_Type'].unique())
centre_circle = plt.Circle((0,0),0.80,fc='white')
fig = plt.gcf()
fig.gca().add_artist(centre_circle)
plt.tight_layout()
plt.show()
```



From the above pie chart we can conclude that owner type first were re saled more.

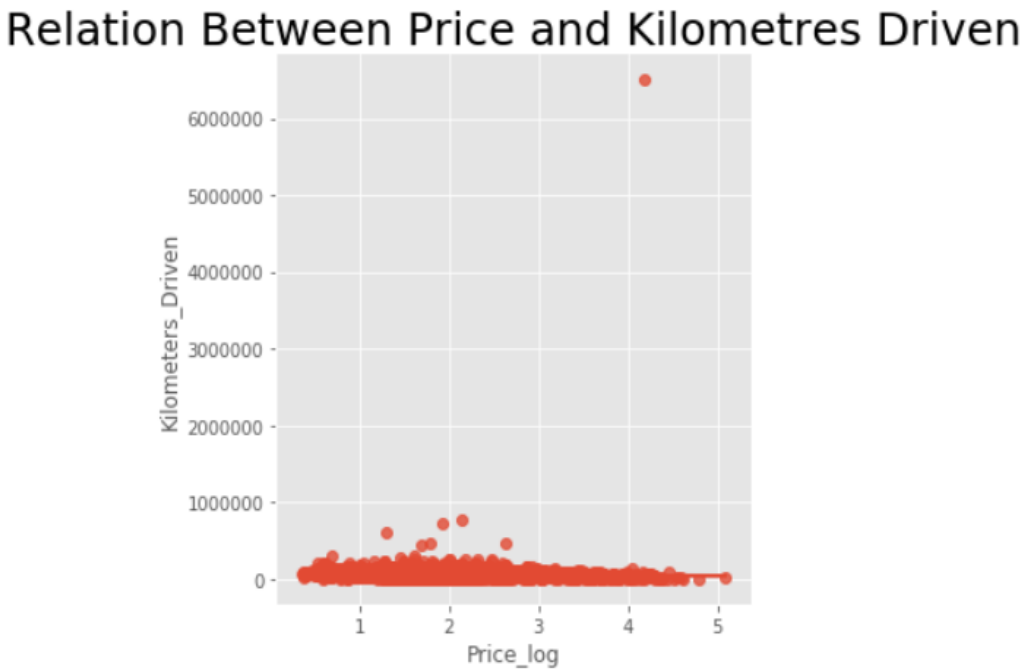
In [51]:
```
sns.lmplot('Year','Price_log',data=df_train)
plt.title('Relation Between Price and Year',fontsize=25)
```

Out[51]: Text(0.5, 1, 'Relation Between Price and Year')



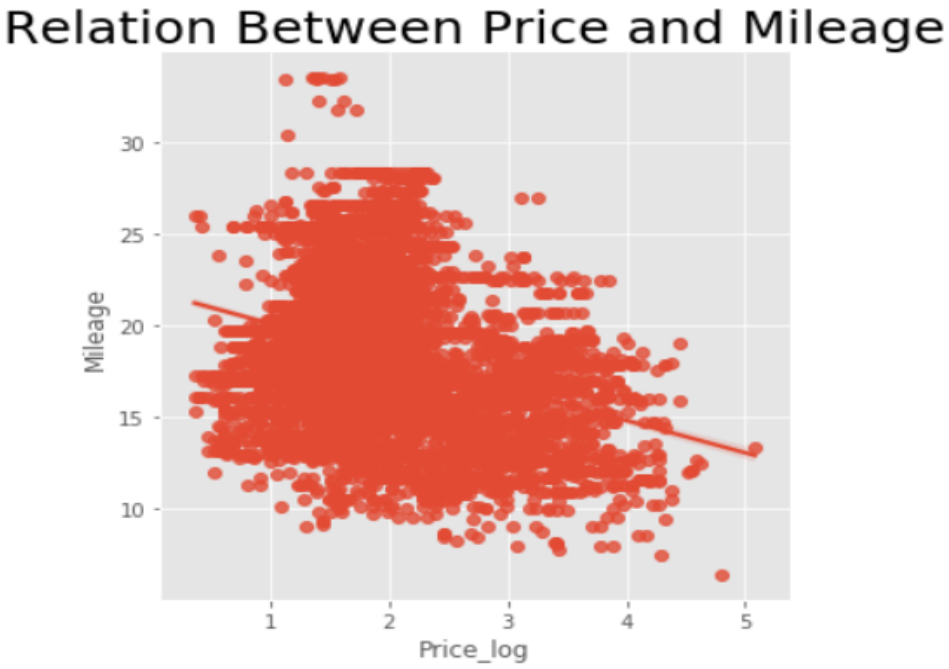
In [52]:
```
sns.lmplot('Price_log','Kilometers_Driven',data=df_train)
plt.title('Relation Between Price and Kilometres Driven',fontsize=25)
```

Out[52]: Text(0.5, 1, 'Relation Between Price and Kilometres Driven')



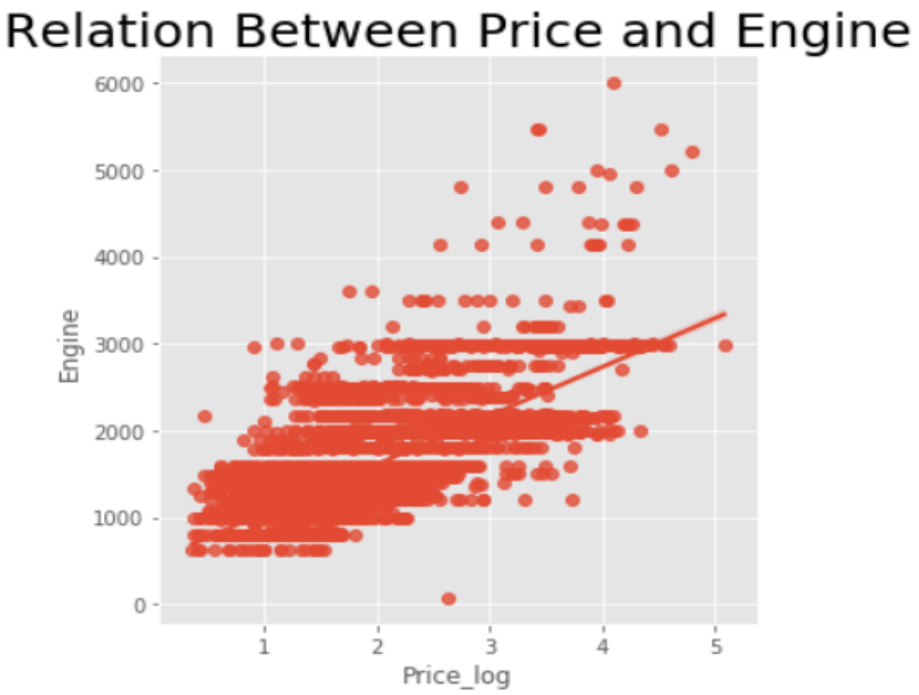
In [56]:
```
sns.lmplot('Price_log','Mileage',data=df_train)
plt.title('Relation Between Price and Mileage',fontsize=25)
```

Out[56]: Text(0.5, 1, 'Relation Between Price and Mileage')



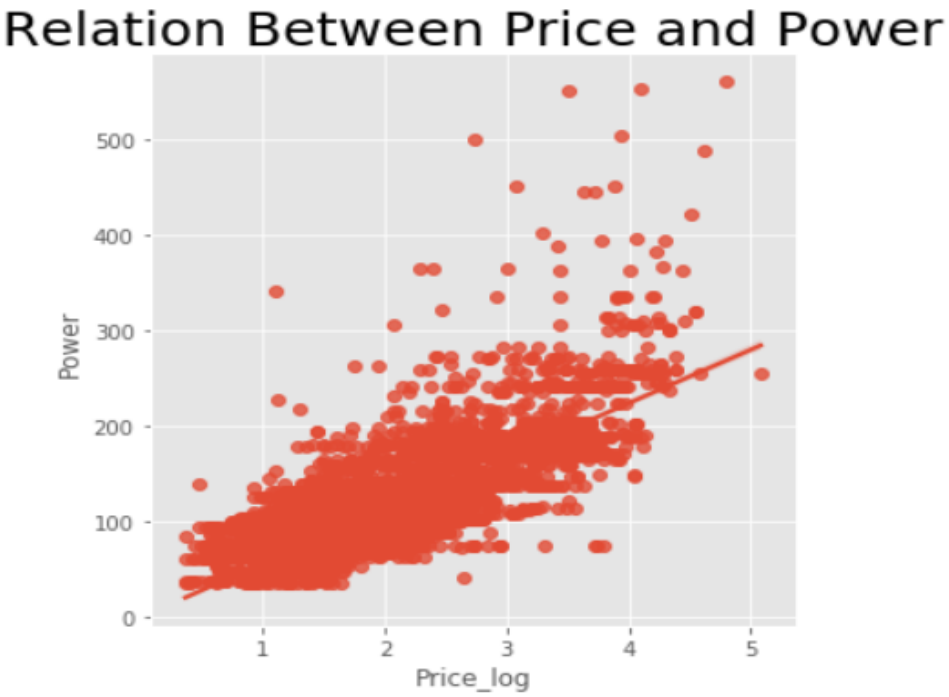
In [57]:
```
sns.lmplot('Price_log','Engine',data=df_train)
plt.title('Relation Between Price and Engine',fontsize=25)
```

Out[57]: Text(0.5, 1, 'Relation Between Price and Engine')



In [58]:
```
sns.lmplot('Price_log','Power',data=df_train)
plt.title('Relation Between Price and Power',fontsize=25)
```

Out[58]: Text(0.5, 1, 'Relation Between Price and Power')



In [59]:
```
sns.lmplot('Price_log','Seats',data=df_train)
plt.title('Relation Between Price and Seats',fontsize=25)
```

Out[59]: Text(0.5, 1, 'Relation Between Price and Seats')

8. Conclusions

We can predict car resale value using Random forest regression and Support Vector regression.

We took a dataset of approximately 6050 cars re saled in India of different locations. We trained the dataset using the above regression methods one at a time. 25% of the data was used for prediction or testing purpose whereas 75% of the data was used for training purpose.

We calculated the accuracy scores of the algorithms on the dataset, random forest regression gave 93.56% of accuracy and support vector regression gave 68.3% of accuracy.

Therefore, we can conclude that random forest regression gave the highest accuracy than the support vector regression. So, random forest regression is best fit for this dataset.

The results we got clearly supports our models assumption.

9. Future Work

Machine Learning is an application of Artificial Intelligence. It allows software applications to become accurate in predicting outcomes. Machine Learning focuses on the development of computer programs, and the primary aim is to allow computers to learn automatically without human intervention.

A similar web or mobile application can be made using AI in the future which can determine the resale value of the used cars. In that app user will enter the details like location, name, seats, engine type, kilometres driven etc…, the application will analyse the past records and predict the re sale value of the entered car details.

Besides this application we can also use for the resale of the books, electrical appliances like television, radio, laptop etc. with the appropriate dataset and changing the training model and can give it to any resale shops.

10. REFERENCES

We have taken the help of following websites during the course of our project which has helped us in making our project successful.

- https://link.springer.com/chapter/10.1007/978-1-4302-5990-9_4#:~:text=As%20in%20classification%2C%20support%20vector,in%20real%2Dvalue%20function%20estimation.

- https://www.udemy.com/course/data-science-and-machine-learning-with-python-hands-on/

- https://medium.com/swlh/random-forest-and-its-implementation-71824ced454f

- https://www.geeksforgeeks.org/random-forest-regression-in-python/

Implementation (or) Source code

https://drive.google.com/drive/folders/1B8XYac8MVlEf3hPFuOEMD_BhSEs-wDtD?usp=sharing