Capstone Project
Car evaluation

Machine Learning Nanodegree

Shaik Arshiya Yasmine

27th Feb, 2019

1. Definition

Domain Background:

Essentially these are the means for assessing a car. By utilizing this we can pass judgment on whether the vehicle is commendable or not.
Stage I: Research the Car's VIN. Each vehicle has a novel vehicle distinguishing proof number(VIN).
Stage II: Informal Inspection.
Stage III: Test Drive.
Stage IV: Professional Inspection
While coming to specialized viewpoints these are the past use Vehicle Evaluation Database was gotten from a basic various levelled choice model

initially produced for the exhibition of DEX, M. Bohanec,V. Rajkovic: Expert framework for basic leadership. Sistemica 1(1), pp. 145-157, 990.). M. Bohanec and V. Rajkovic: Knowledge obtaining and clarification for multi-trait basic leadership. In eighth Intl Workshop on Expert Frameworks and their Applications, Avignon, France. pages 59-78, 1988.

Problem Statement:

The issue explanation is to assess a vehicle dependent on an objective idea (CAR), the model incorporates three halfway ideas: PRICE, TECH, COMFORT
The objective variable is CAR (car agreeableness) which implies whether the vehicle is commendable or not. This should be possible by halfway ideas which I described below.
The model assesses cars as indicated by the accompanying idea structure:
CAR -car acceptability

 PRICE- overall price

  o buying- buying price

  o maint -price of the maintenance

 TECH -technical characteristics

 COMFORT-comfort

  o doors-number of doors

  o persons-capacity in terms of persons to carry

  o lug_boot- the size of luggage boot

  o safety stimated safety of the car

By utilizing these intermediate ideas we will make judgment about that car. Based on these ideas we will get an end whether the vehicle is great or awful.

Metrics:

I need to use accuracy score as an evaluation metric for the forecast of credit approval. In the dataset, the class marks (+,-) are firmly balanced. So we can utilize accuracy score as the evaluation metric.

Here I am foreseeing the accuracy score of the chose models. We will choose a model whose accuracy score is more noteworthy than the various models and we treat it as the best.

For discovering the accuracy, we have the following equation:
Accuracy Score= TP+TN/TP+FP+FN+TN

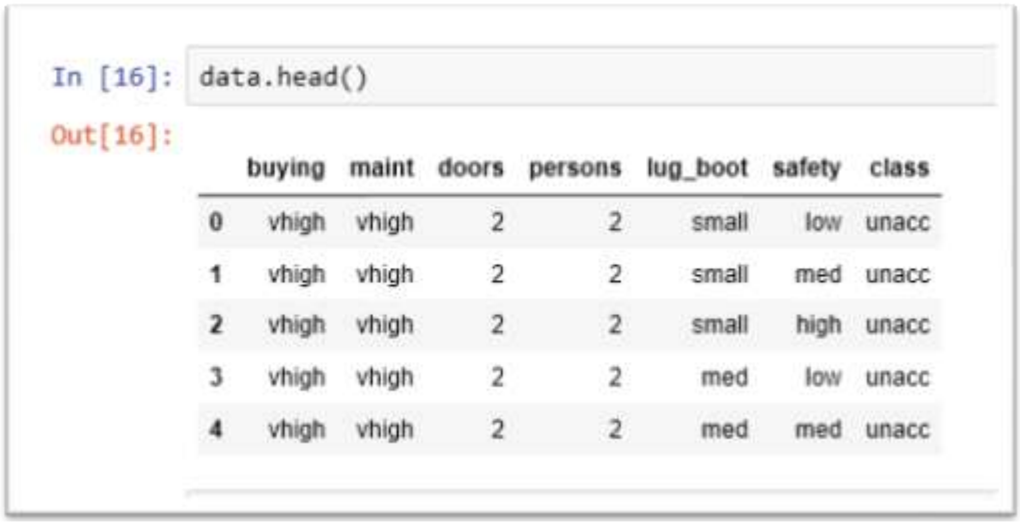
True Positives: are the qualities which are accurately anticipated as positives
True Negatives: are the qualities which are effectively delegated negatives.
False Positives: are the qualities which are wrongly delegated positives. These are likewise type-1 mistakes.

False Negatives: are the qualities which are wrongly named negatives. These are additionally called as sort 2 blunders.

II. Analysis

Data Exploration:

In this dataset I have used 1728 instances and 6 attributes to evaluate accuracy score. The attributes are shown below:



```
In [16]: data.head()
Out[16]:
```

	buying	maint	doors	persons	lug_boot	safety	class
0	vhigh	vhigh	2	2	small	low	unacc
1	vhigh	vhigh	2	2	small	med	unacc
2	vhigh	vhigh	2	2	small	high	unacc
3	vhigh	vhigh	2	2	med	low	unacc
4	vhigh	vhigh	2	2	med	med	unacc

<u>Attribute information:</u>

Buying: v-high, high, med, low

maint: v-high, high, med, low

doors: 2, 3, 4, 5-more

persons: 2, 4, more

lug_boot: small, med, big

safety: low, med, high

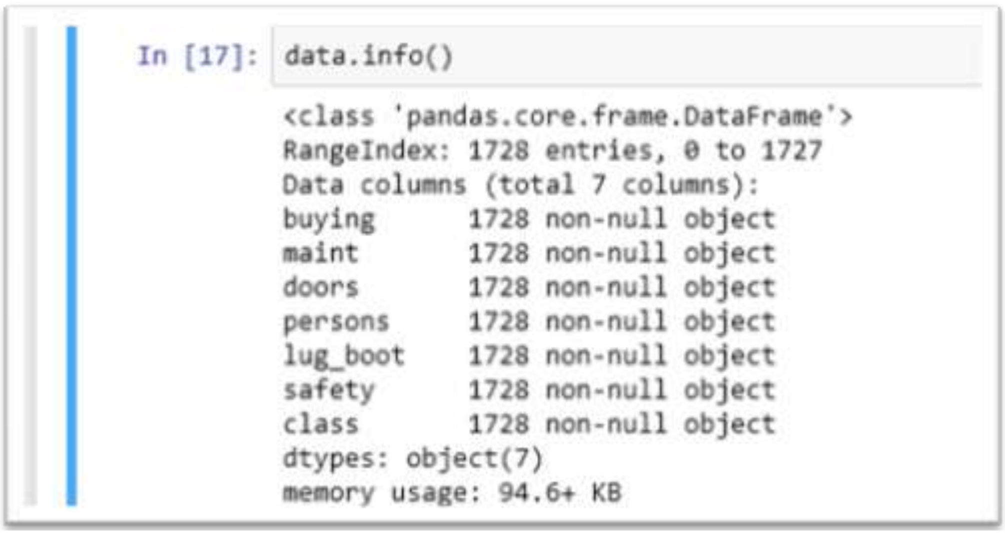
<u>**Class Distribution (number of instances per class):**</u>

class	N	N[%]
unacc	1210	(70.023 %)
acc	384	(22.222 %)
good	69	(3.993 %)
v-good	65	(3.762 %)

This describes the distribution of target class(CAR acceptability). The total 1728 instances are divided into 4 classes as mentioned above.

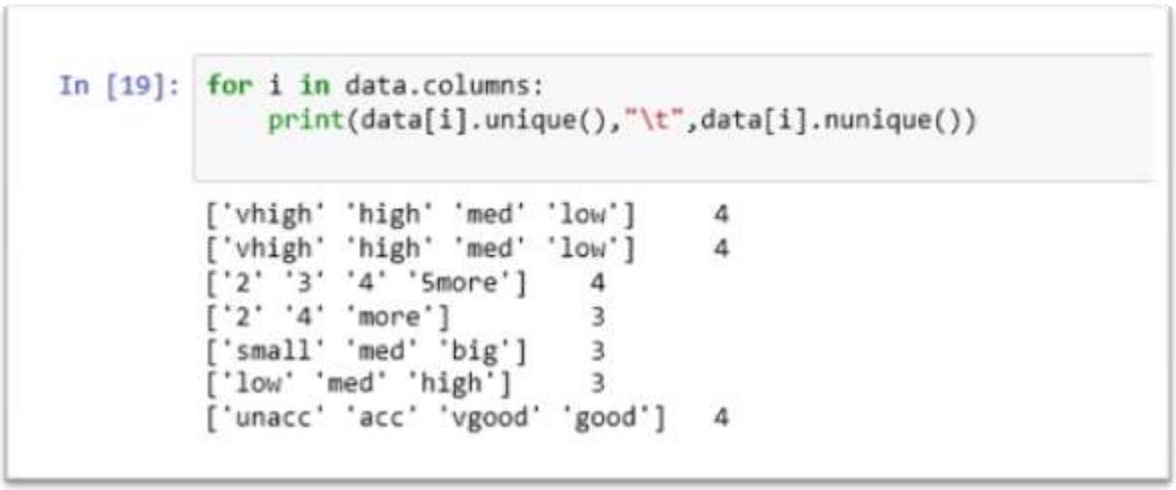
Here I am using a Multivariate dataset.

Information of dataset:



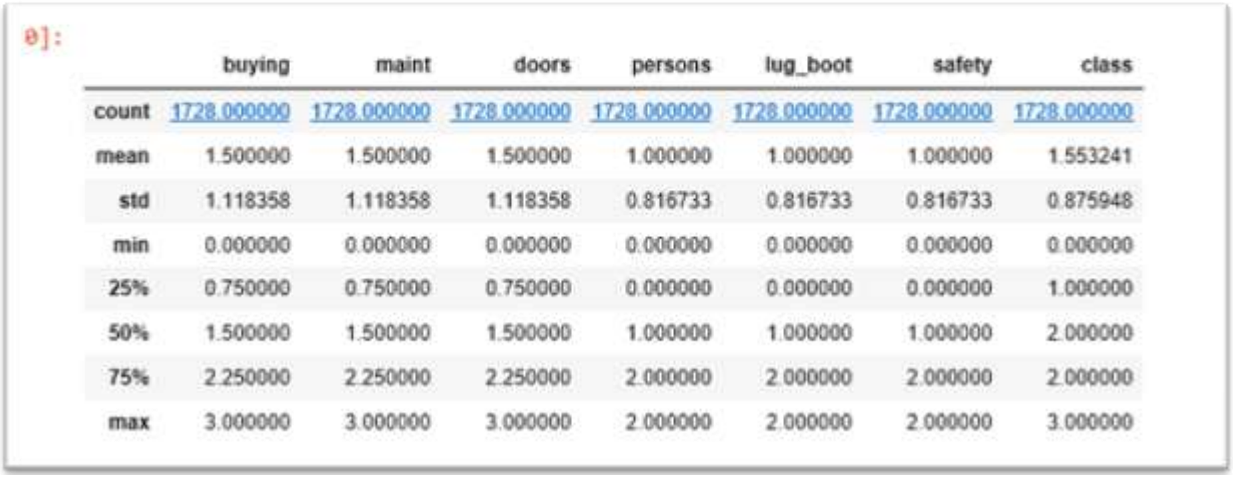
```
In [17]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1728 entries, 0 to 1727
Data columns (total 7 columns):
buying      1728 non-null object
maint       1728 non-null object
doors       1728 non-null object
persons     1728 non-null object
lug_boot    1728 non-null object
safety      1728 non-null object
class       1728 non-null object
dtypes: object(7)
memory usage: 94.6+ KB
```

I have checked for unique values of each column because all the columns are categorical.



```
In [19]: for i in data.columns:
             print(data[i].unique(),"\t",data[i].nunique())

['vhigh' 'high' 'med' 'low']     4
['vhigh' 'high' 'med' 'low']     4
['2' '3' '4' '5more']    4
['2' '4' 'more']         3
['small' 'med' 'big']    3
['low' 'med' 'high']     3
['unacc' 'acc' 'vgood' 'good']   4
```

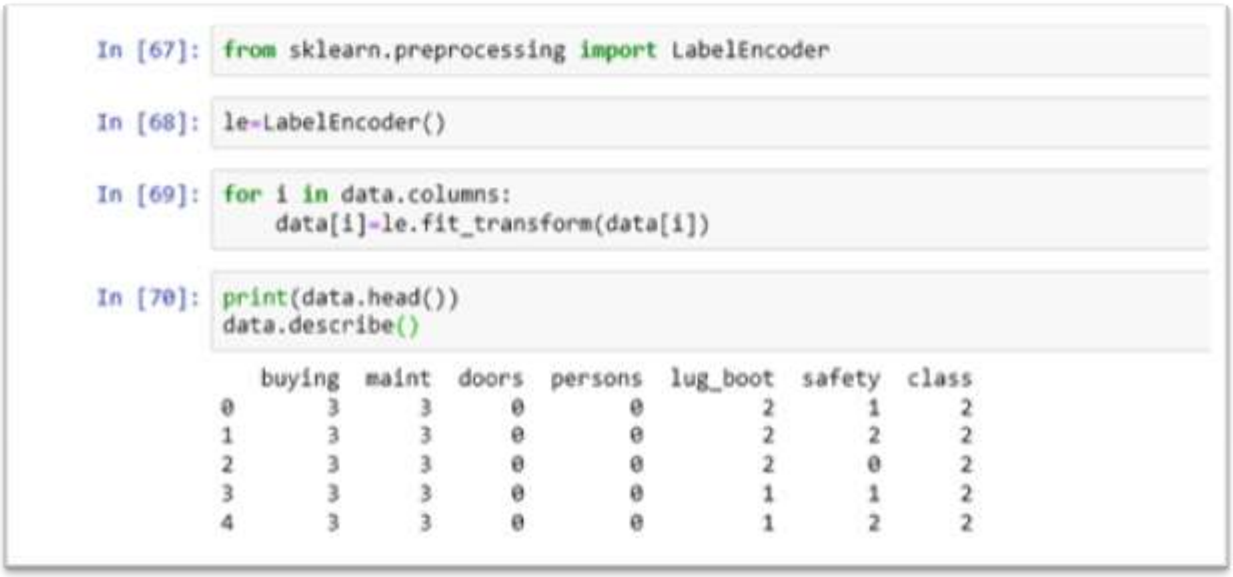
And, we can see how the features are distributed across the dataset using describe() as shown below.



0]:

	buying	maint	doors	persons	lug_boot	safety	class
count	1728.000000	1728.000000	1728.000000	1728.000000	1728.000000	1728.000000	1728.000000
mean	1.500000	1.500000	1.500000	1.000000	1.000000	1.000000	1.553241
std	1.118358	1.118358	1.118358	0.816733	0.816733	0.816733	0.875948
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.750000	0.750000	0.750000	0.000000	0.000000	0.000000	1.000000
50%	1.500000	1.500000	1.500000	1.000000	1.000000	1.000000	2.000000
75%	2.250000	2.250000	2.250000	2.000000	2.000000	2.000000	2.000000
max	3.000000	3.000000	3.000000	2.000000	2.000000	2.000000	3.000000

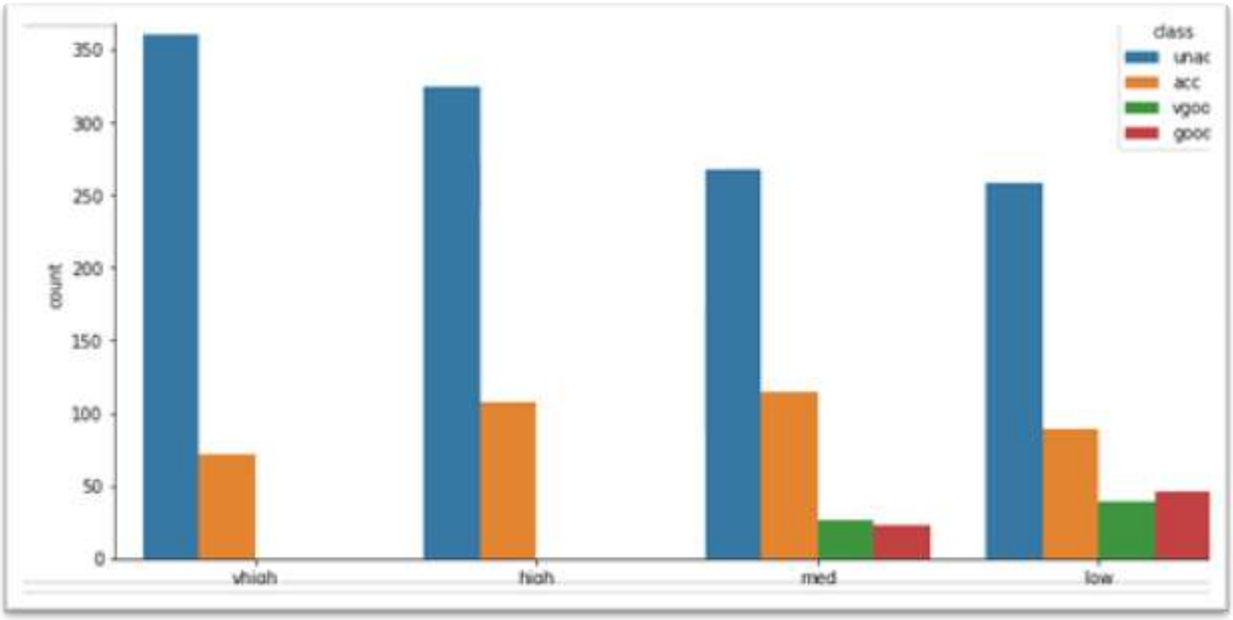
In order to apply describe(), I had to convert the text data into numerical data using LabelEncoder() from sklearn.preprocessing as shown below.



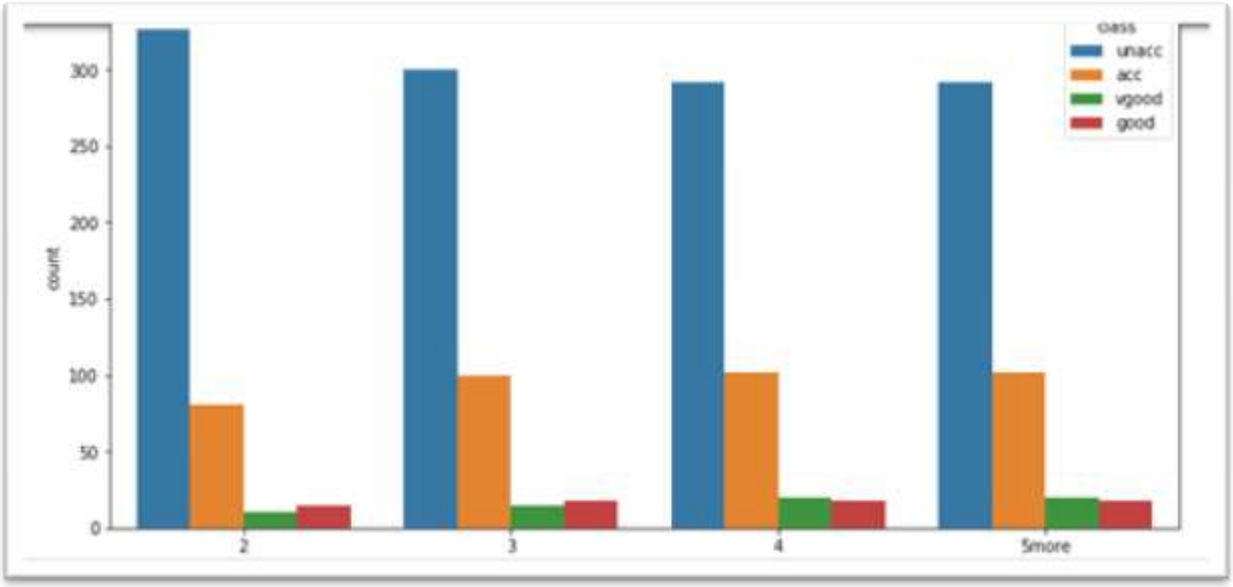
```
In [67]: from sklearn.preprocessing import LabelEncoder

In [68]: le=LabelEncoder()

In [69]: for i in data.columns:
             data[i]=le.fit_transform(data[i])

In [70]: print(data.head())
         data.describe()

   buying  maint  doors  persons  lug_boot  safety  class
0       3      3      0        0         2       1      2
1       3      3      0        0         2       2      2
2       3      3      0        0         2       0      2
3       3      3      0        0         1       1      2
4       3      3      0        0         1       2      2
```

DATA Visualization:

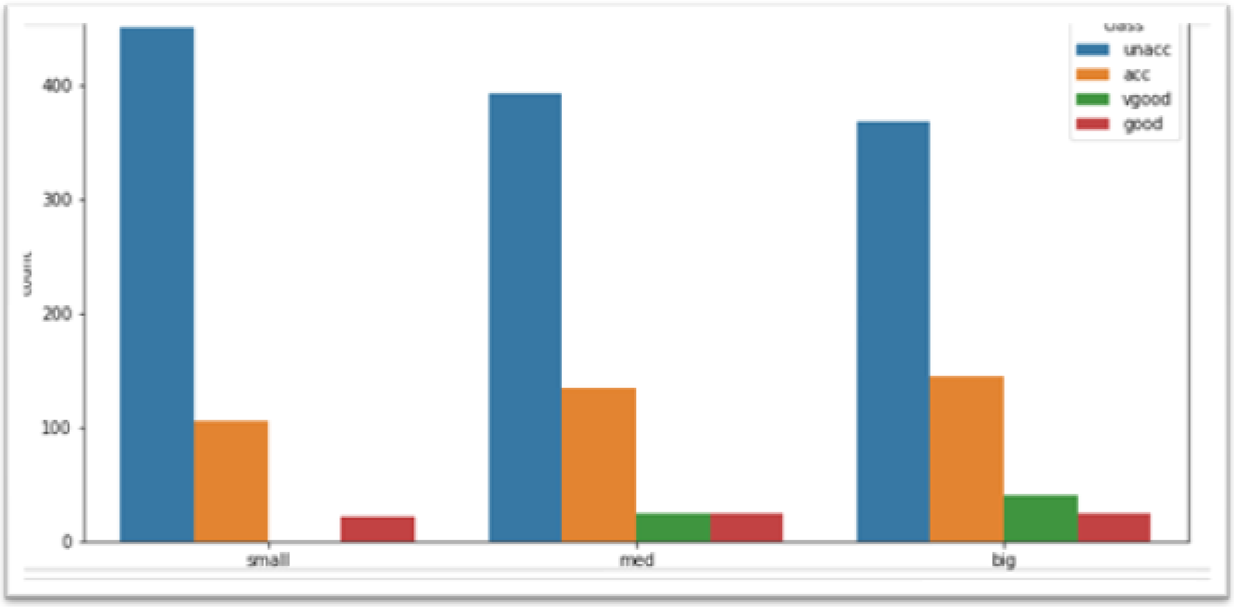
Buying



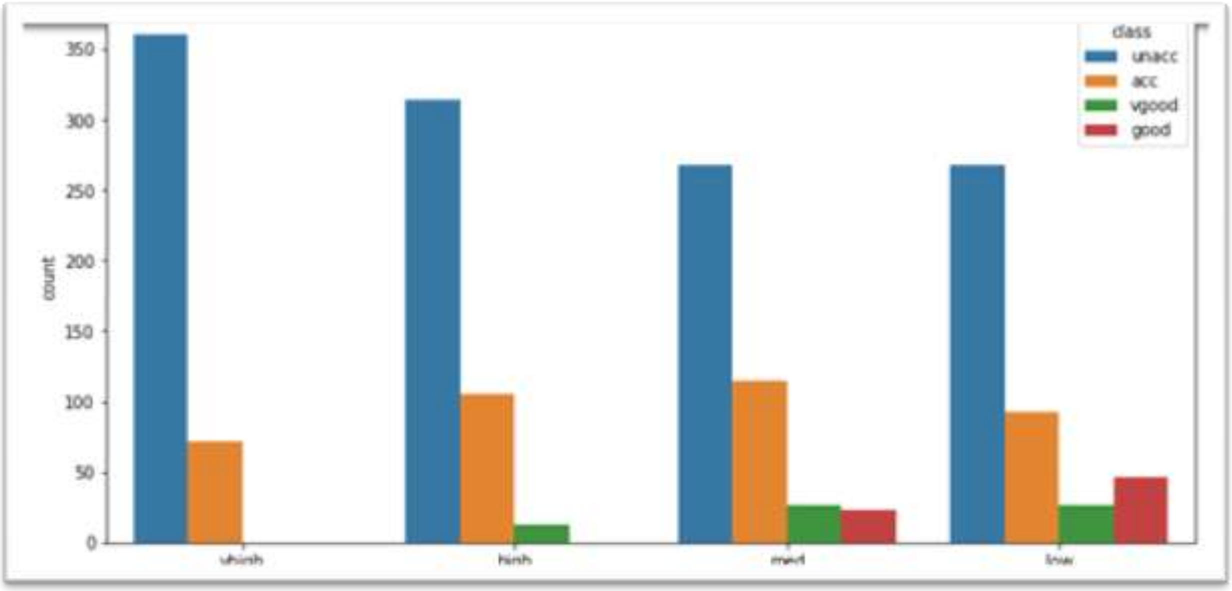
Doors



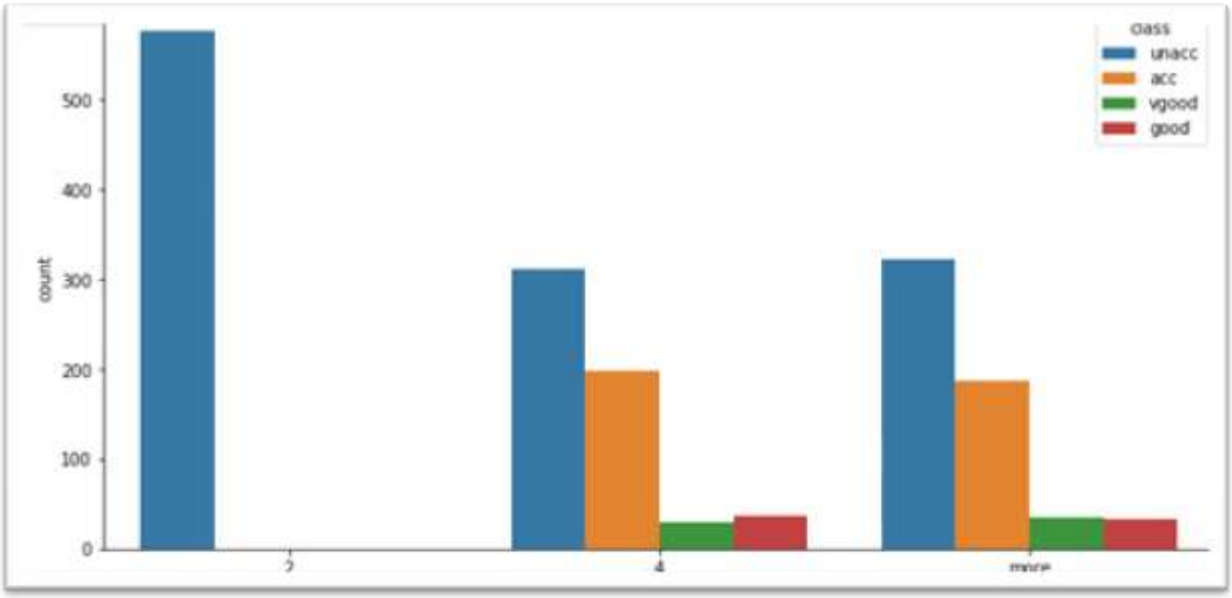
Lug_boot:



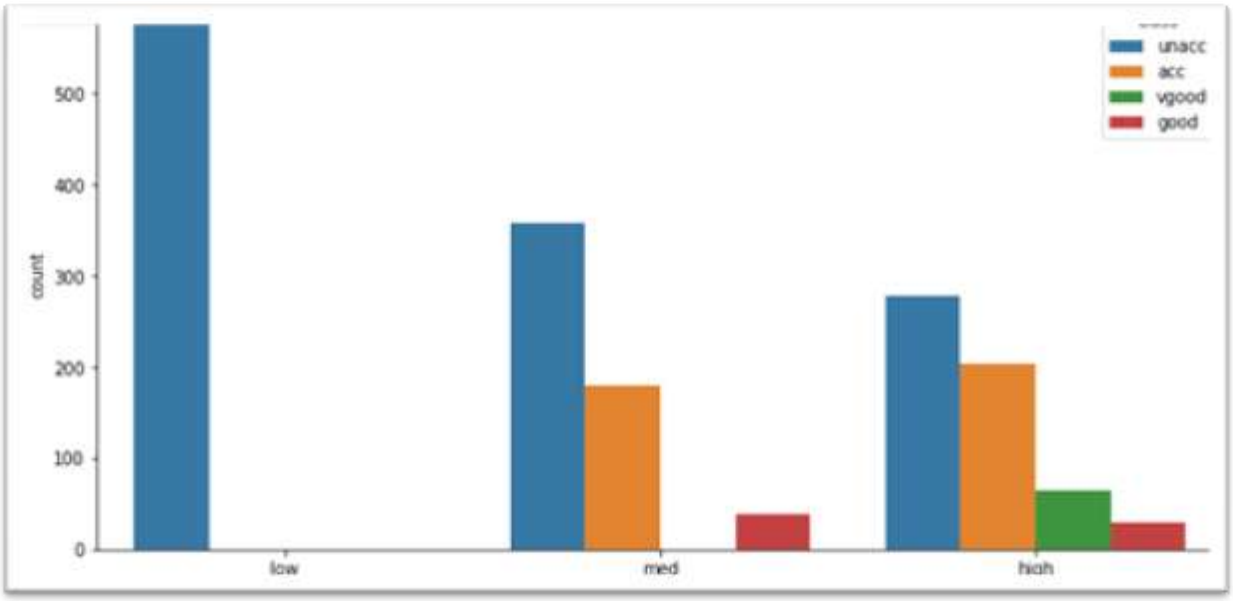
Maint:



Persons:



Safety:

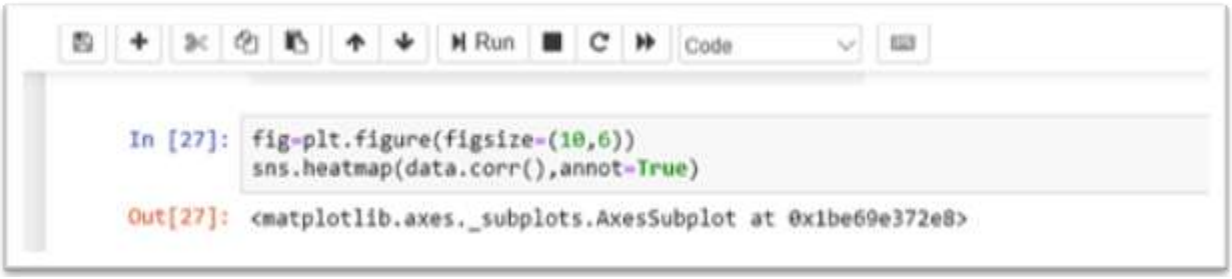


The above visualizations show the co-relation between features and target class. Each feature will be in relation with target class differently.

Heat map:

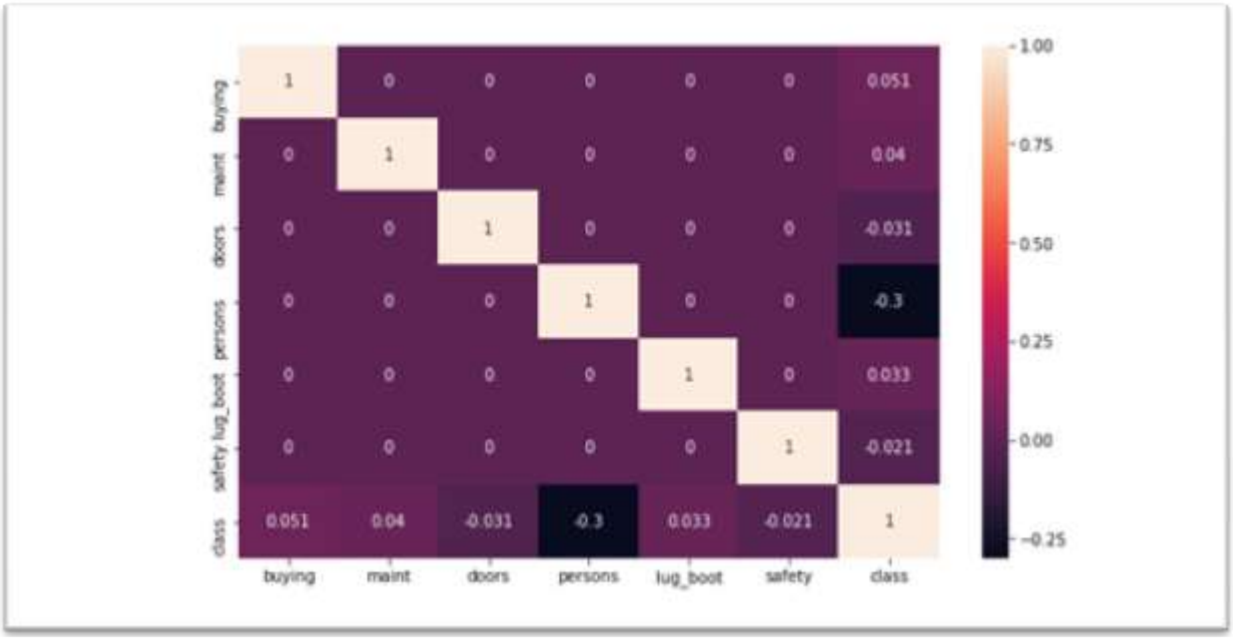
The heat map is a 2-D representation of data in which values are represented by colours. A simple heat map provides the immediate visual summary of information. More elaborate heat maps allow the user to understand complex data.
By using the below code we will generate the heat map between the attributes and try to

deduce the correlation between the attributes. We will use the following code to do so.



```
In [27]: fig=plt.figure(figsize=(10,6))
         sns.heatmap(data.corr(),annot=True)

Out[27]: <matplotlib.axes._subplots.AxesSubplot at 0x1be69e372e8>
```

By executing the above code we get the following heat map:



	buying	maint	doors	persons	lug_boot	safety	class
buying	1	0	0	0	0	0	0.051
maint	0	1	0	0	0	0	0.04
doors	0	0	1	0	0	0	-0.031
persons	0	0	0	1	0	0	-0.3
lug_boot	0	0	0	0	1	0	0.033
safety	0	0	0	0	0	1	-0.021
class	0.051	0.04	-0.031	-0.3	0.033	-0.021	1

Disregarding the diagonal values, it very well may be seen that a large portion of the columns demonstrates very weak relationship with 'class'. 'persons' column is showing a weak connection with 'class'. Other columns aside from 'class' demonstrates no connection with one another.

Algorithms and techniques:

Here mainly we will use three algorithms.

1. Logistic Regression (Benchmark Model)
2. KNN classifier.
3. Random Forest.

Logistic Regression:

Logistic Regression is a Machine Learning classification algorithm that is used to foresee the probability of a categorical dependent variable. In logistic regression, the target variable is a binary variable that contains data coded as 1 (yes, success, and so on.) or 0 (no,failed, and so on.). As such, the logistic regression model predicts P(Y=1) as a function of X.

Advantages:

Because of its efficient and clear nature, doesn't require high computation power,, simple to execute, effectively interpretable, used generally by data analyst and scientist. Likewise, it doesn't require scaling of highlights. Logistic regression gives a probability score to predictions.

Disadvantages:

Logistic regression can't deal with countless categorical values. It is powerless against over fitting. Likewise, can't take care of the non-linear problem with the logistic regression, that is the reason it requires a transformation of non-linear features. Logistic regression won't perform well with independent factors that are not corresponded to the objective variable and are fundamentally the same as or associated to one another.

Sample code:

```
from sklearn.model_selection import
learning_curve,cross_val_score,validation_curve
param_range=[0.0001,0.001,0.1,1]
curve=validation_curve(logreg,X_train,y_train,cv=
5,param_name='C',
param_range=param_range,n_jobs=-1,)
```

As the application is classification oriented, the techniques used here are taken from classification techniques.

1. K-Nearest Neighbours (KNN):

K-Nearest Neighbours algorithm is a non-parametric strategy utilized for classification and regression. The principle behind nearest neighbour strategies is to discover a predefined number of training samples nearest in distance to the new point and anticipate the label from these.

Sample code:

```
knn=KNeighborsClassifier(n_jobs=-1)
knn.fit(X_train,y_train)
pred=knn.predict(X_test)
knn.score(X_test,y_test)
```

Advantages:

The K-Nearest Neighbour (KNN) Classifier is a very simple classifier that works well on the basic recognition problems.

Disadvantages:

The fundamental drawback of the KNN algorithm is that it is a lazy learner, for example it doesn't learn (take in anything) from the training data and simply utilizes the training data itself for classification.

To foresee or predict the label of a new instance the KNN algorithm will find the K nearest neighbours to the new instance from the training data, the predicted class label will at that point be set as the most common label among the K closest neighbouring points.

The algorithm must process the distance and sort all the training data at every prediction, which can be slow if there is an extensive number of training data.

Examples:

Another disadvantage of this methodology is that the algorithm does not take in anything from the training data, which can result in the algorithm not summing up well and furthermore not being powerful to noisy data.

Random Forest:

Tree models are known to be high difference, low bias models. In result, they are inclined to over fit the training data. This is catchy on the off chance that we restate what a tree model does on the off chance that we don't prune it or present early stopping criteria like a minimum number of instances per leaf node. Indeed, it tries to split the data along the features until the instances are pure in regards to the values of the target feature, there are no data left, or there are no features left to spit the dataset on. In the event that one of the above holds true, we grow a leaf node.

The result is that the tree show is developed to the maximal depth and therewith attempts to reshape the training data as precise as possible which can easily lead to over fitting. Another disadvantage of traditional tree models like the (ID3 or CART) is that they are generally unstable. This instability can lead to the circumstance that a little change in the arrangement of the data set prompts a totally different tree model.

Sample code:

```
from sklearn.ensemble import
RandomForestClassifier
rfc=RandomForestClassifier(n_jobs=-1,
random_state=51)
from sklearn.metrics import f1_score
rfc.fit(X_train,y_train)
print(rfc.score(X_test,y_test))
print(f1_score(y_test,rfc.predict(X_test),
average='macro'))
```

Advantages:

1. Reduction in over fitting: by averaging a few trees, there is an essentially lower danger of over fitting.
2. Less variance: By utilizing various trees, you decrease the opportunity of staggering over a

classifier that doesn't perform well as a result of the relation between the train and test data.

Disadvantages:

1. It requires more time to train samples.

Benchmark Model:

Here, we will pick Logistic regression model as the benchmark model. By applying this logistic regression we got an accuracy of 0.718 that is 71.8%. And now, we will attempt and accomplish better accuracy over this model by utilizing the above mentioned referenced classification models.

III. Methodology:

Data Pre-processing:

In this step, we will pre-process the information. Data pre-processing is viewed as the first and foremost step that will be done before beginning any procedure. We will get the data by using read_csv. Then, we will know the size and shape of the data-set. After that we will find the unique class qualities by using unique(). By using the info() we will know the information of the attributes. Then, we will check whether there are any null values by utilizing isnull ().

LabelEncoder can be utilized to normalize labels. We will import LabelEncoder from sklearn.preprocessing. We will likewise utilize fit_transform(y) for Fit label encoder and return encoded labels. Then, that we will utilize le.transform () to transform labels to standardized encoding.

After that we will divide the entire data into training and testing data. We will assign 70% of the data to the training set and the remaining 30% of the data into testing data. We will do this by using train_test_split from sklearn.model_selection.

Implementation:

Out of the picked models we will initially take KNN classifier model. We will take the classifier and fit the training data. After that we will predict that by utilizing predict (X_train). Presently we will predict the accuracy of the testing data by using accuracy_score (y_test, pred).

By doing as such for the Logistic regression, which is the benchmark, will give us the accuracy of 0.71.

We will now, pick the model which will give us the better score over the benchmark model out of KNN classifier and Random Forest. By following a similar strategy over that is fitting, predicting and finding the accuracy score we will get the accuracy score as below.

KNN classifier:0.90
Irregular Forest: 0.984

From the above reports Random Forest is by all accounts performing admirably.

Initially, as the columns consist of textual data, I had no idea how to know the distribution of the features across the dataset. But, the use of LabelEncoder() from sklearn.preprocessing made the task easier.

Refinement:

I found the random forest as the best classifier out of the picked classifiers. Now, we will perform tuning of random forest classifier so as to achieve the better accuracy. For this we will utilize GridSearchCV.

For doing refinement we will simply tune the parameter.
Here, I tried tuning the following parameters.
criterion : [gini, entropy]
max_depth : [2, 5, 10, 20]
max_features : [2, 4, 6, auto]
max_leaf_nodes : [2, 3, None]
And, the best parameters I got are:
Criterion : entropy
max_depth : 10
max_features : 6
max_leaf_nodes : None
And, the accuracy got increased to 98.4%.

IV. Result

Model evaluation and Metrics:

The final model we have picked is tuned random forest which gave us more accuracy that is 0.984. So as achieve this accuracy we took n_estimators=[50] and 'paradigm': ['gini', 'entropy'] however without utilizing the n_estimators and standard we got the accuracy of just 0.977 which is a bit less than the tuned esteem. Here we can say that the solution is sensible on the grounds that we are getting significantly much less accuracy while utilizing different models.

The final model that is tuned random forest has been tried with different data sources to assess whether the model sums up well. This model is likewise sufficiently robust for the given problem. Our data already has less features and even if we drop the least important feature, then also the accuracy is reducing to 93.64%

Justification:

My final model's solution is better than that of benchmark model.

	Tuned random forest	Benchmark model
(Accuracy)	0.984	0.71

From the above we can conclude that the results for the final model are stronger than the benchmark model.
Hence we can say that tuned random forest provides the significant to solve the problem of predicting the outcome of the car evaluation.

V. Conclusion

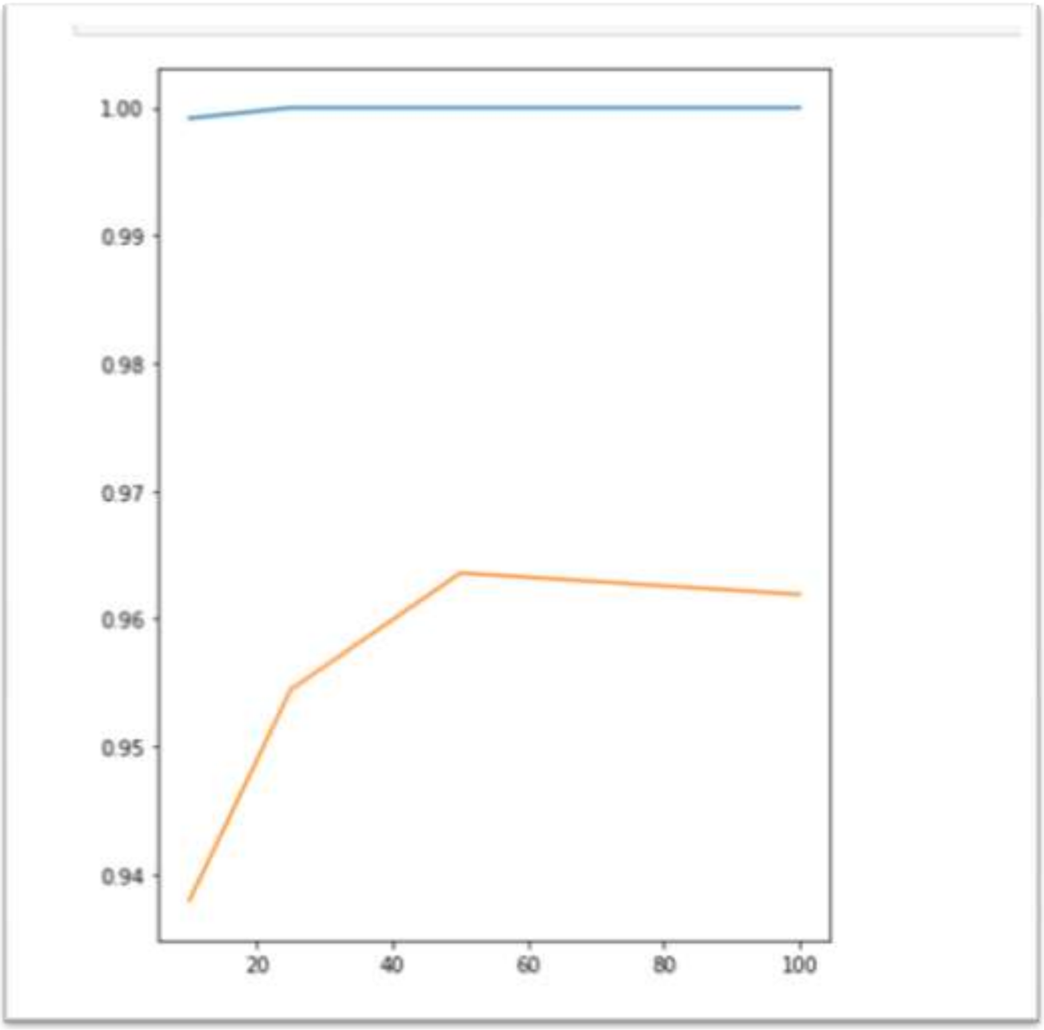
The goal of the project was to compare different machine learning algorithms and predict the car is worthy or not by using different features like buying, maintenance, doors, persons, lug_boot, safety, class.
ACCURACY

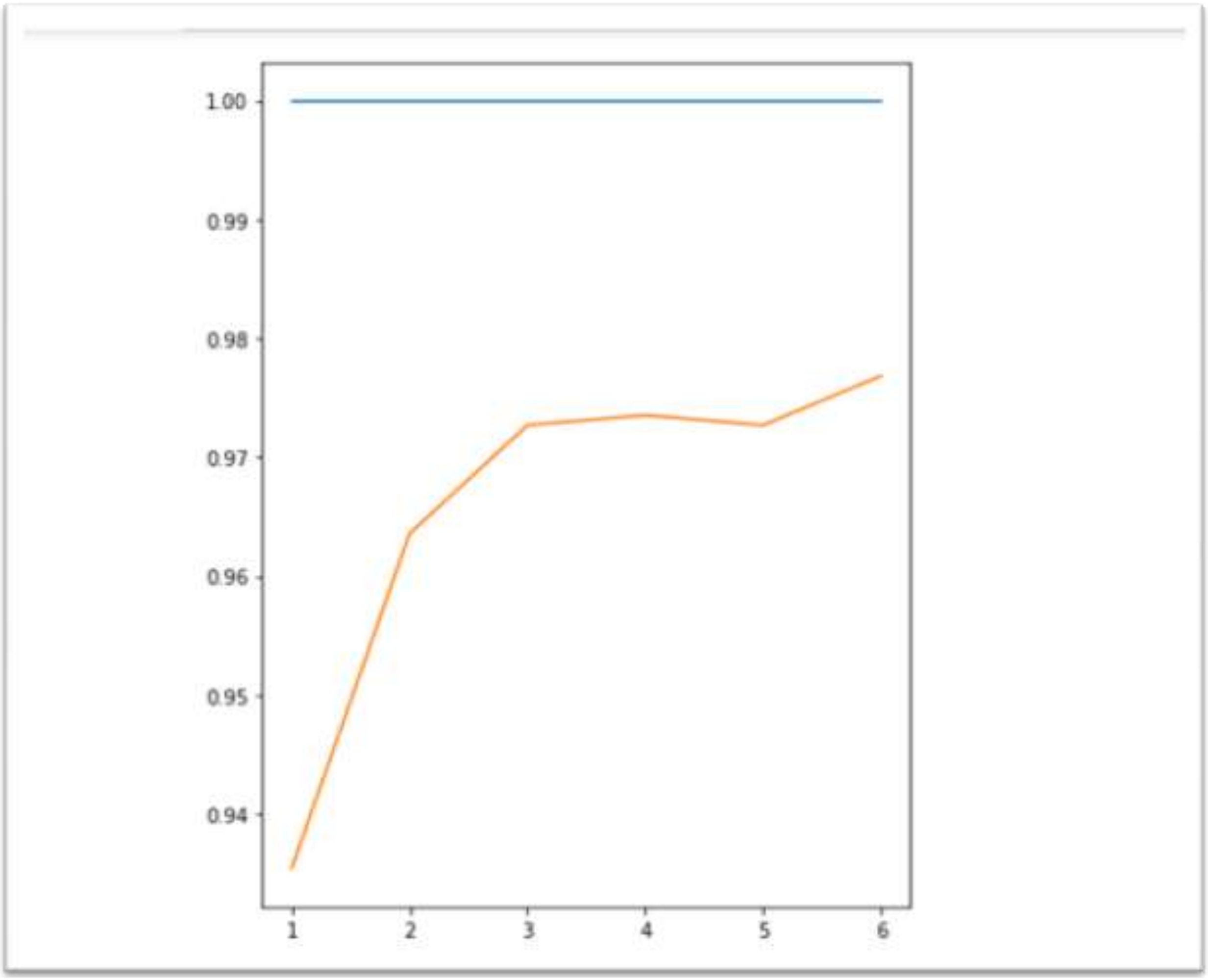
Logistic regression 0.71
KNN 0.90
Random Forest 0.984
From the results we can easily see that the Random forest has the highest accuracy score of 0.984 compared with the other models. By producing decent results, simpler methods proved to be useful as well.

Free-form visualization:

For better visualization, I used learning curves. A learning curve is the representation in graph form of the rate of learning something over time or repeated experiences.

The below graphs represent the learning curves for the classifier random forest with the parameters (n_estimators and max_features resp.).


1.00
0.99
0.98
0.97
0.96
0.95
0.94
20
40
60
80
100



Reflection:

1. I have learnt how to visualize and understand the data.
2. I have learnt that the data cleaning place a very vital role in data analytics.
3. Removing the data features which are not necessary in evaluating model is very important.
4. I got to know how to use the best technique for the data using appropriate ways.

5. I got to know how to tune the parameters in order to achieve the best score.
6. On a whole I learnt how to graph a data set and applying cleaning techniques on it and to fit the best techniques to get best score.

Improvement:

The process which I have followed can be improved to classify not only for cars but can be extended to automobile industry which consists large vehicles also. Some Features will be added based on that vehicle. As we can say that there has never been an end in the machine learning there will be many more models to learn. By taking more amounts of data sets, we can get the model more generalized and optimized to get better accurate result.