Project Title : Cars Model Prediction

Date : 16-06-2023 Prepared By : SURIYAKRISHNAN

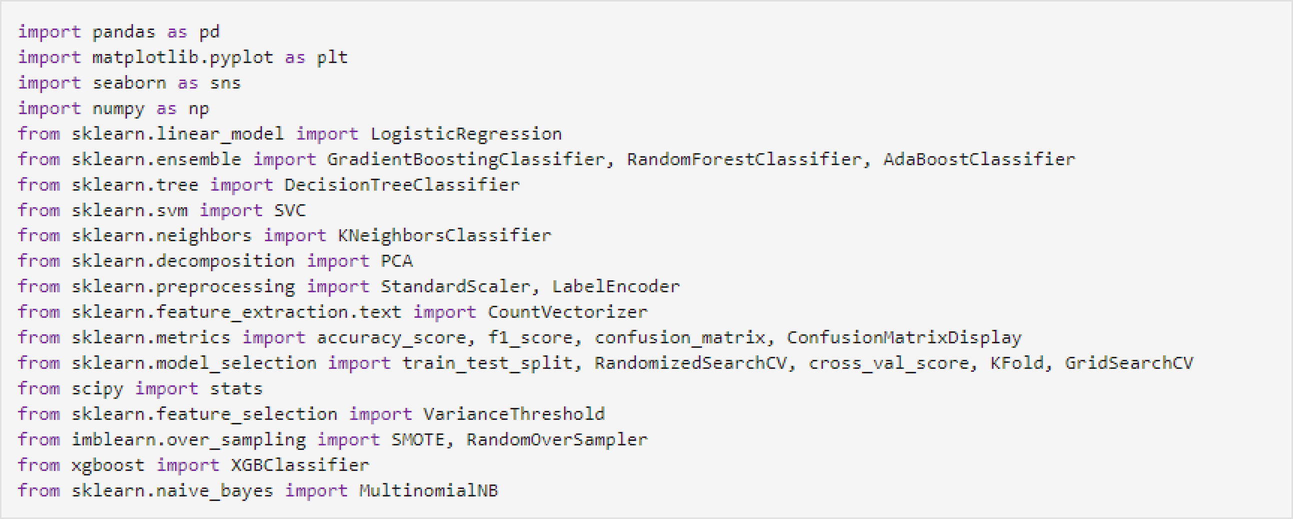
Project Justification :
This project is to predict the cars model its depends on the cars features, and this is a Multiclass Classification datasets. This is a classification problem.

Project Requirements :

✓ Dataset

✓ Python’s libraries(necessory)

import necessory libraries



```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import GradientBoostingClassifier, RandomForestClassifier, AdaBoostClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score, f1_score, confusion_matrix, ConfusionMatrixDisplay
from sklearn.model_selection import train_test_split, RandomizedSearchCV, cross_val_score, KFold, GridSearchCV
from scipy import stats
from sklearn.feature_selection import VarianceThreshold
from imblearn.over_sampling import SMOTE, RandomOverSampler
from xgboost import XGBClassifier
from sklearn.naive_bayes import MultinomialNB
```

First need to import or load the necessary and related libraries, Some very primary and almost necessary packages for Machine Learning are — NumPy, Pandas, Matplotlib and Scikit-Learn.

Dataset details:

Dataset file: ‘cars_class.csv’

This is a multi-class classification data set.

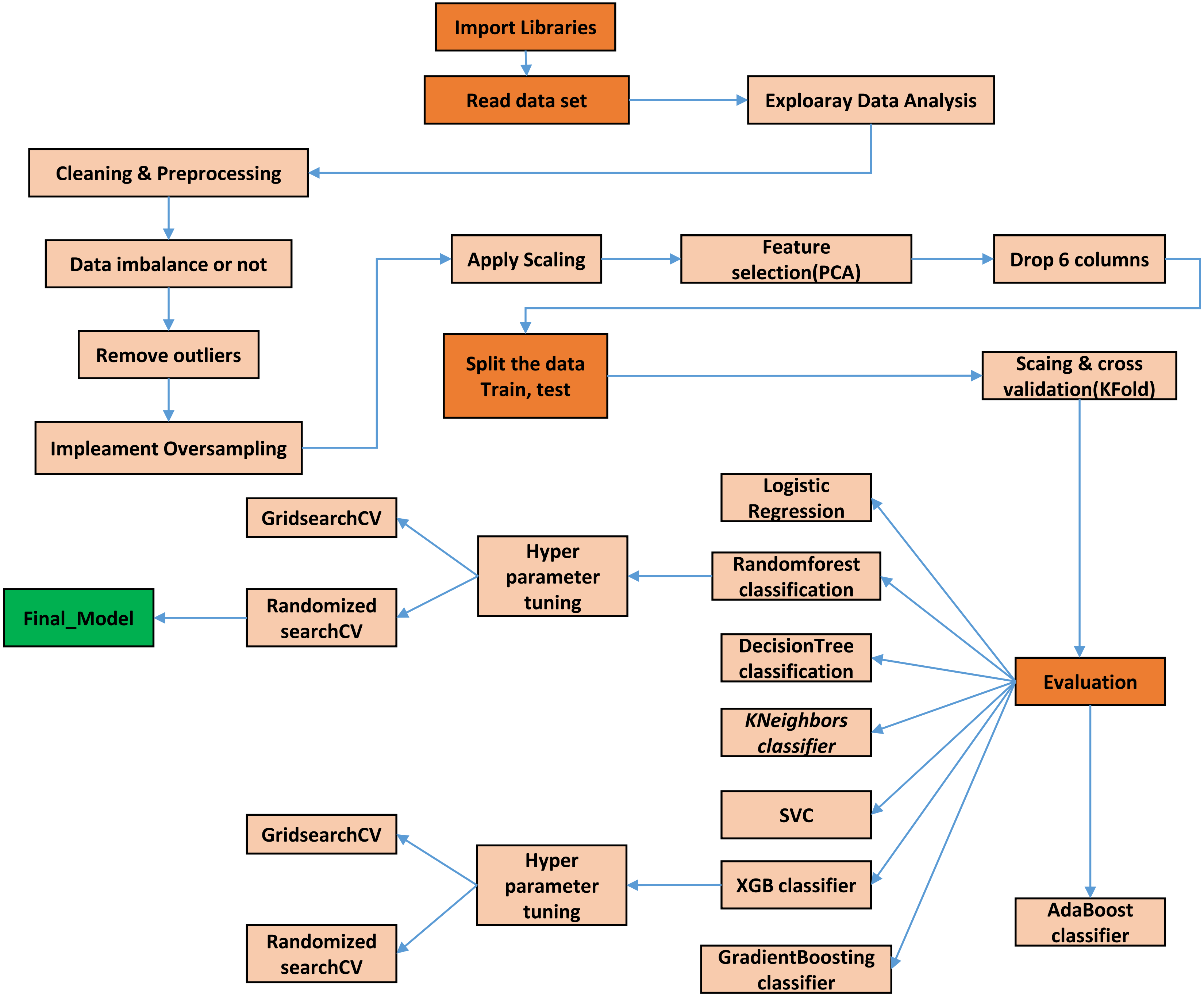
The data set has 719 samples.

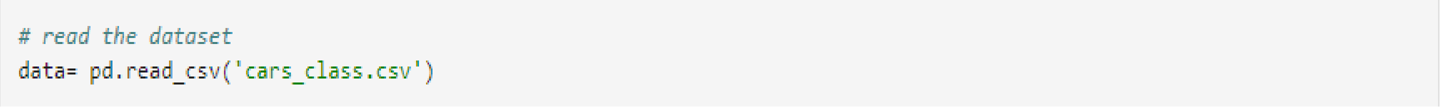
There are 18 numerical features.(without ‘ID’ and target feature-’class’)

The target variable is the class of the car which may be one of: 0 –bus, 1 – Opel Manta, 2 – Saab, 3 – Van

Description of attributes:

- ✓ Comp: Compactness
- ✓ Circ: Circularity
- ✓ D.Circ: Distance Circularity
- ✓ Rad.Ra: Radius ratio
- ✓ Pr.Axis.Ra: pr.axis aspect ratio
- ✓ Max.L.Ra: max.length aspect ratio
- ✓ Scat.Ra: scatter ratio
- ✓ Elong: elongatedness
- ✓ Pr.Axis.Rect: pr.axis rectangularity
- ✓ Max.L.Rect: max.length rectangularity
- ✓ Sc.Var.Maxis: scaled variance along major axis
- ✓ Sc.Var.maxis: scaled variance along minor axis
- ✓ Ra.Gyr: scaled radius of gyration
- ✓ Skew.Maxis: skewness about major axis
- ✓ Skew.maxis: skewness about minor axis
- ✓ Kurt.maxis: kurtosis about minor axis
- ✓ Kurt.Maxis: kurtosis about major axis
- ✓ Holl.Ra: hollows ratio


Import Libraries
Read data set
Exploaray Data Analysis
Cleaning & Preprocessing
Data imbalance or not
Remove outliers
Impleament Oversampling
Apply Scaling
Feature selection(PCA)
Drop 6 columns
Split the data Train, test
Scaing & cross validation(KFold)
Evaluation
Logistic Regression
Randomforest classification
DecisionTree classification
KNeighbors classifier
SVC
XGB classifier
GradientBoosting classifier
AdaBoost classifier
Hyper parameter tuning
GridsearchCV
Randomized searchCV
Final_Model
Hyper parameter tuning
GridsearchCV
Randomized searchCV



```
# read the dataset
data= pd.read_csv('cars_class.csv')
```

Once the libraries are loaded, need to get the data loaded. Pandas has a very straightforward function to perform this task — pandas.read_csv. The read.csv function is not just limited to csv files, but also can read other text based files as well.

Exploratory Data Analysis:

✓ To check how many duplicates values in the dataset

✓ To show how many Non-values are there, whthere dataset is imbalance or not, suppose the dataset is imbalanced we need to impleament oversampling but the dataset is not a imbalanced.

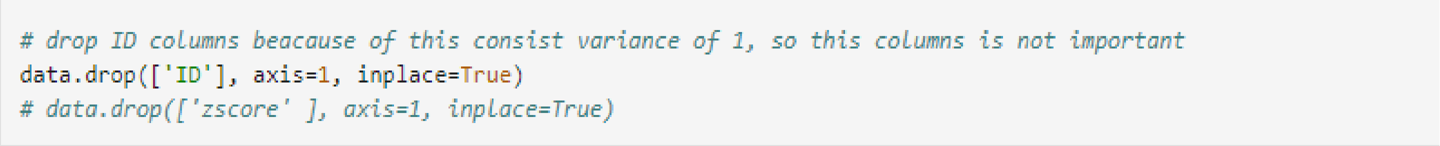
✓ This is dataset balanced dataset

Remove Outliers:

✓ Remove the outliers in the dataset based on the Z-score(-3 to +3) , but in this case have a some ouliers in the dataset but these are acceptable level.

✓ Suppose remove the ouliers that is will produce very low accuracy score

Data cleaning & Preprocessing:



```
# drop ID columns beacause of this consist variance of 1, so this columns is not important
data.drop(['ID'], axis=1, inplace=True)
# data.drop(['zscore' ], axis=1, inplace=True)
```

✓ Columns "ID" is not a high varience columns and this columns will not affect the model

✓ And then split the data of X and y (X is undependent columns, y is dependent column- Target column)

✓ This is dataset is very low size so that this dataset is giving low accuracy score. It's affect the model prediction , so apply the OverSampling(SMOTE)

✓ Scaling (StandardScaler) apply to the X

Feature Selection(PCA):

PCA applied to X and check how much of varience in each columns , the last few columns are giving very low varience

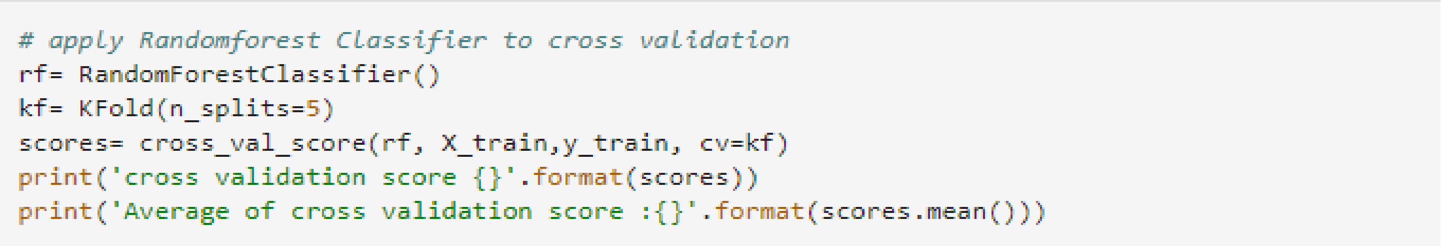
The last 6 unnecessary coumns removed from the dataset

Remove Outliers:

- ✓ Remove the outliers in the dataset based on the Z-score(-3 to +3) , but in this case have a some ouliers in the dataset but these are acceptable level.
- ✓ Suppose remove the ouliers that is will produce very low accuracy score

Evaluation:

- ✓ Split the data to Train and Test for the Evaluation
- ✓ Training set have a 80% and The Test 20% of the data
- ✓ Standard Scaler implement to X_train and X_test sperately



```
# apply Randomforest Classifier to cross validation
rf= RandomForestClassifier()
kf= KFold(n_splits=5)
scores= cross_val_score(rf, X_train,y_train, cv=kf)
print('cross validation score {}'.format(scores))
print('Average of cross validation score :{}'.format(scores.mean()))
```

- ✓ Kfold Cross Validation apply to RandomForest Classifier it's give the average cross validation score is 0.72
- ✓ To predict the model for some most commonly using classification models

Evaluation for model Prediction:

<u>**Model**</u>		<u>**accuracy_score**</u>
✓ ***Logistic Regression***	*:*	**0.74**
✓ ***RandomForest Classifier***	***:***	***0.81***

	Model		accuracy_score
✓	***DecisionTree Classifier***	***:***	***0.74***
✓	***Kneighbors Classifier***	***:***	***0.79***
✓	**SVC**	**:**	**0.79**
✓	***XGB Classifier***	***:***	***0.84***
✓	***AdaBoost Classifier***	***:***	***0.62***
✓	***GradientBoosting Classifier***	***:***	***0.80***

Hyper Parameter Tuning:

In this case Randomforest classifier and XGB boost classifier are giving the best accuracy score , so RandomizedsearchCV and GridsearchCV applied to both.

After hyper parameter tuning (Randomized SearchCV) Randomforest Classifier is give the best score

Conclusion:

RandomForest Classifier Tuned using RandomizedSearchCV and GridSearchCV. I get better from RandomizedSearchCV so finalize this model to ‘final_model’

Final_Model	Accuracy score	F1_score
RandomForest Classifier (RandomizedSearchCV)	**0.8289**	**0.8245**