In [1]:

```
#Ensembles
#Name: Shreevatsa
#Regno: 200970093
#Dataset: mtcars
import pandas as pd
```

In [2]:

```
df = pd.read_csv('Dataset/mtcars.csv')
df.head()
```

Out[2]:

	model	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
0	Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
1	Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
2	Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
3	Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
4	Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

In [3]:

```
df.columns
```

Out[3]:

```
Index(['model', 'mpg', 'cyl', 'disp', 'hp', 'drat', 'wt', 'qsec', 'vs', 'am',
       'gear', 'carb'],
      dtype='object')
```

In [4]:

```
df.isna().sum()
```

Out[4]:

```
model    0
mpg      0
cyl      0
disp     0
hp       0
drat     0
wt       0
qsec     0
vs       0
am       0
gear     0
carb     0
dtype: int64
```

In [5]:

```
df.shape
```

Out[5]:

```
(32, 12)
```

In [6]:

```
x= df.drop(columns = 'carb')
y = df['carb']
x = x.drop(columns = 'model')
```

In [7]:

```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test=train_test_split(x,y,test_size=0.3)
```

In [8]:

```
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier()
```

```
knn = model.fit(X_train,Y_train)
```

In [9]:

```
from sklearn.tree import DecisionTreeClassifier
model2=DecisionTreeClassifier(criterion='entropy',max_depth=1)
dt = model2.fit(X_train,Y_train)
```

In [10]:

```
from sklearn.naive_bayes import GaussianNB
model3=GaussianNB()
nb = model3.fit(X_train,Y_train)
```

In [11]:

```
from sklearn.svm import SVC
model4=SVC()
svm = model4.fit(X_train,Y_train)
```

In [12]:

```
pred1=model.predict(X_test)
pred2=model2.predict(X_test)
pred3=model3.predict(X_test)
pred4=model4.predict(X_test)
```

In [13]:

```
print("KNN: ",model.score(X_test,Y_test)*100)
```

```
KNN:  60.0
```

In [14]:

```
print("Decision Tree: ",model2.score(X_test,Y_test)*100)
```

```
Decision Tree:  30.0
```

In [15]:

```
print("Naive Bayes: ",model3.score(X_test,Y_test)*100)
```

```
Naive Bayes:  50.0
```

In [16]:

```
print("SVM: ",model4.score(X_test,Y_test)*100)
```

```
SVM:  40.0
```

In [17]:

```
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
```

In [18]:

```
knnPipeline = Pipeline([('myscalar',MinMaxScaler()),
                        ('mypca',PCA(n_components=2)),
                        ('knn_classifier',KNeighborsClassifier())])
```

In [19]:

```
dtPipeline = Pipeline([('myscalar',MinMaxScaler()),
                       ('mypca',PCA(n_components=2)),
                       ('dt_classifier',DecisionTreeClassifier())])
```

In [20]:

```
nbPipeline = Pipeline([('myscalar',MinMaxScaler()),
                       ('mypca',PCA(n_components=2)),
                       ('nb_classifier',GaussianNB())])
```

In [21]:

```
svmPipline = Pipeline([('myscalar',MinMaxScaler()),
                       ('mypca',PCA(n_components=2)),
                       ('svm_classifier',SVC())])
```

In [22]:

```
#Model training
mypipelines = [knnPipeline,dtPipeline,nbPipeline,svmPipline]
```

In [23]:

```
#creating a pipeline dictionary
pipe_dict = {0:'KNN Classifier',1:'DecisionTree Classifier',2:'NaiveBays Classifier',3:'SV
```

In [24]:

```
for pipe in mypipelines:
    pipe.fit(X_train,Y_train)
```

In [25]:

```
#checking accuracy for each classifiers
for i,model in enumerate(mypipelines):
    print("{} Test Accuracy : {} ".format(pipe_dict[i],
                                          model.score(X_test,Y_test)*100))
```

```
KNN Classifier Test Accuracy : 50.0
DecisionTree Classifier Test Accuracy : 60.0
NaiveBays Classifier Test Accuracy : 40.0
SVM  Test Accuracy : 40.0
```

In [26]:

```
from sklearn.ensemble import AdaBoostClassifier
AdaBoost=AdaBoostClassifier(base_estimator=DecisionTreeClassifier(),
                            n_estimators=10,
                            learning_rate=1)
AdaBoost.fit(X_train,Y_train)
AdaBoost.score(X_test,Y_test)*100
```

Out[26]:

```
80.0
```