

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from sklearn.decomposition import PCA
from sklearn.pipeline import Pipeline
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier

# Filtering Warnings
import warnings
warnings.filterwarnings('ignore')
```

```
In [2]: DiabetesData= pd.read_csv(r"C:\Users\lenka\Downloads\archive.zip")
DiabetesData.head()
```

```
Out[2]:
```

	6	148	72	35	0	33.6	0.627	50	1
0	1	85	66	29	0	26.6	0.351	31	0
1	8	183	64	0	0	23.3	0.672	32	1
2	1	89	66	23	94	28.1	0.167	21	0
3	0	137	40	35	168	43.1	2.288	33	1
4	5	116	74	0	0	25.6	0.201	30	0

```
In [3]: X=DiabetesData.iloc[:,0:8]
y=DiabetesData.iloc[:,8]
```

```
In [4]: X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=1)
```

```
In [5]: LogisticRegressionPipeline = Pipeline([('myscaler',MinMaxScaler()),
                                             ('mypca',PCA(n_components=3)),
                                             ('logistic_classifier',LogisticRegression())])
```

```
In [ ]:
```

```
In [6]: DecisionTreePipeline = Pipeline([('myscaler', MinMaxScaler()),
                                         ('mypca', PCA(n_components=3)),
                                         ('decisiontree_classifier', DecisionTreeClassifier())])
```

```
In [7]: RandomForestPipeline = Pipeline([('myscaler',MinMaxScaler()),
                                         ('mypca',PCA(n_components=3)),
                                         ('randomforest_classifier',RandomForestClassifier())])
```

```
In [8]: mypipeline=[LogisticRegressionPipeline,DecisionTreePipeline,RandomForestPipeline]
```

```
In [11]: for mypipe in mypipeline:
mypipe.fit(X_train,y_train)
```

```
In [12]: accuracy = 0.0
classifier = 0
pipeline = ''
```

```
In [13]: PipelineDict = {0:'Logistic Regression',1:'Decission Tree',2:'Random Forest'}
```

```
In [15]: for i , model in enumerate(mypipeline):
print('{} Test Accuracy: {}'.format(PipelineDict[i],model.score(X_test, y_test)))
```

```
Logistic Regression Test Accuracy: 0.7077922077922078
Decission Tree Test Accuracy: 0.7012987012987013
Random Forest Test Accuracy: 0.7532467532467533
```

```
In [16]: for i, model in enumerate(mypipeline):
if model.score(X_test,y_test):
accuracy = model.score(X_test,y_test)
pipeline =model
classifier=i
print('Classifier with best accuracy:{}'.format(PipelineDict[classifier]))
```

```
Classifier with best accuracy:Random Forest
```

```
In [ ]:
```