Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

BE-ETRX B **Sub- AIML Lab**
Name: Shubham Sawant **UID :2019110050**

Name of the Experiment: **Implement different Classifier**

Objective: To explore the different classifier on different dataset

Outcomes:
1. Identifying the classifier based on the dataset
2. Build the model and classify it using linear and nonlinear classifier(SVM , SVR, Random forest, KNN)
3. Draw various plots and interpret them

System Requirements: Windows with MATLAB

Theory:

➢ SVM:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane. SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

➢ SVR:

Support Vector Regression is a supervised learning algorithm that is used to predict discrete values. Support Vector Regression uses the same principle as the SVMs. The basic idea behind SVR is to find the best fit line. In SVR, the best fit line is the hyperplane that has the maximum number of points. Unlike other Regression models that try to minimize the error between the real and predicted value, the SVR tries to fit the best line within a threshold value. The threshold value is the distance between the hyperplane and boundary line. The fit time complexity of SVR is more than quadratic with the number of samples which makes it hard to scale to datasets with more than a couple of 10000 samples.

➢ Random forest:

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model. Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset." Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output. The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting

➢ KNN:

K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well-suited category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric

algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Dataset Description:

Number of Instances: 1000

Number of Attributes: 12

Attribute Information:

1. Age | Objective Feature | age | int (days)
2. Height | Objective Feature | height | int (cm) |
3. Weight | Objective Feature | weight | float (kg) |
4. Gender | Objective Feature | gender | categorical code |
5. Systolic blood pressure | Examination Feature | ap_hi | int |
6. Diastolic blood pressure | Examination Feature | ap_lo | int |
7. Cholesterol | Examination Feature | cholesterol | 1: normal, 2: above normal, 3: well above normal |
8. Glucose | Examination Feature | gluc | 1: normal, 2: above normal, 3: well above normal |
9. Smoking | Subjective Feature | smoke | binary |
10. Alcohol intake | Subjective Feature | alco | binary |
11. Physical activity | Subjective Feature | active | binary |
12. Presence or absence of cardiovascular disease | Target Variable | cardio | binary |

Code:

Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

BE-ETRX B **Sub- AIML Lab**

Name: Shubham Sawant **UID :2019110050**

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.colors import ListedColormap
import seaborn as sns
from sklearn.preprocessing import normalize, LabelEncoder
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.metrics import precision_score,recall_score,accuracy_score
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
```

[ ]

```
# reading csv file and extracting class column to y.
data_set= pd.read_csv(r'C:\Users\Dell\Desktop\Shubham\SEM7\AIML\EXP5\cardio_train.csv',sep=';')
x= data_set.iloc[:, [9,10,11]].values
y= data_set.iloc[:, 12].values
print(x)
print(y)
```

[ ]

```
# Splitting the dataset into training and test set.
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size=0.25, random_state=0)
svclassifier = SVC(kernel='linear')
clf=svclassifier.fit(x_train, y_train)
y_pred = svclassifier.predict(x_test)
```

```
cm= confusion_matrix(y_test, y_pred)
print(cm)
```

```
print(accuracy_score(y_test, y_pred))
print(precision_score(y_test, y_pred))
print(recall_score(y_test, y_pred))
```

```
X = x_train[np.logical_or(y_train==0,y_train==1)]
Y = y_train[np.logical_or(y_train==0,y_train==1)]
# The equation of the separating plane is given by all x so that
# np.dot(svc.coef_[0], x) + b = 0
# Solve for w3 (z)
z = lambda x,y: (-clf.intercept_[0]-clf.coef_[0][0]*x-clf.coef_[0][1]*y) / clf.coef_[0][2]
tmp = np.linspace(-5,5,30)
x,y = np.meshgrid(tmp,tmp)
# print(x,y,z)
fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot3D(X[Y==0,0], X[Y==0,1], X[Y==0,2],'ob')
ax.plot3D(X[Y==1,0], X[Y==1,1], X[Y==1,2],'sr')
ax.plot_surface(x, y, z(x,y))
ax.view_init(30, 60)
plt.show()
```

Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

BE-ETRX B **Sub- AIML Lab**

Name: Shubham Sawant **UID :2019110050**

```
# reading csv file and extracting class column to y.
data_set= pd.read_csv(r'C:\Users\Dell\Desktop\Shubham\SEM7\AIML\EXP5\cardio_train.csv',sep=';')
x= data_set.iloc[:, 9:10].values
y= data_set.iloc[:, 11:12].values
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size=
0.25, random_state=0)
```

```
sc_X = StandardScaler()
sc_y = StandardScaler()
X = sc_X.fit_transform(x_train)
y = sc_y.fit_transform(y_train)
X_test = sc_X.fit_transform(x_test)
y_test = sc_y.fit_transform(y_test)
#4 Fitting the Support Vector Regression Model to the dataset
# Create your support vector regressor here
from sklearn.svm import SVR
# most important SVR parameter is Kernel type. It can be
# linear,polynomial or gaussian SVR. We have a non-linear condition
# #so we can select polynomial or gaussian but here weselect RBF(a #gaussian type) kernel.
regressor = SVR(kernel='rbf')
regressor.fit(X,y)
#5 Predicting a new result
y_pred = regressor.predict(y_test)
plt.scatter(X, y, color = 'magenta')
plt.plot(x_test, regressor.predict(x_test), color = 'green')
plt.title('Truth or Bluff (Support Vector Regression Model)')
plt.xlabel('Activity')
plt.ylabel('Cardio vascular disease')
plt.show()
X_grid = np.arange(min(X), max(X), 0.1)
X_grid = X_grid.reshape((len(X_grid), 1))
plt.scatter(X, y, color = 'red')
plt.plot(X_grid, regressor.predict(X_grid), color = 'blue')
```

```
x= data_set.iloc[:, 2:11].values
y= data_set.iloc[:, 12].values
print(x)
print(y)
# Splitting the dataset into training and test set.
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test= train_test_split(x, y, test_size=0.25, random_state=0)
```

```
st_x= StandardScaler()
x_train= st_x.fit_transform(x_train)
x_test= st_x.transform(x_test)
#Fitting K-NN classifier to the training set
# from sklearn.neighbors import KNeighborsClassifier
classifier= KNeighborsClassifier(n_neighbors=24,metric='minkowski', p=2 )
classifier.fit(x_train, y_train)
y_pred= classifier.predict(x_test)
#Creating the Confusion matrix
# from sklearn.metrics import confusion_matrix
cm= confusion_matrix(y_test, y_pred)
print(cm)
# from sklearn.metrics import precision_score,recall_score,accuracy_score
print(accuracy_score(y_test, y_pred))
print(precision_score(y_test, y_pred))
print(recall_score(y_test, y_pred))
error_rate = []
for i in range(1,40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(x_train,y_train)
    pred_i = knn.predict(x_test)
    error_rate.append(np.mean(pred_i != y_test))
plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed',marker='o',markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
```

Bharatiya Vidya Bhavan's
Sardar Patel Institute of Technology
Bhavan's Campus, Munshi Nagar, Andheri (West), Mumbai-400058-India
(Autonomous College Affiliated to University of Mumbai)

BE-ETRX B **Sub- AIML Lab**
Name: Shubham Sawant **UID :2019110050**

Interpretation of output:

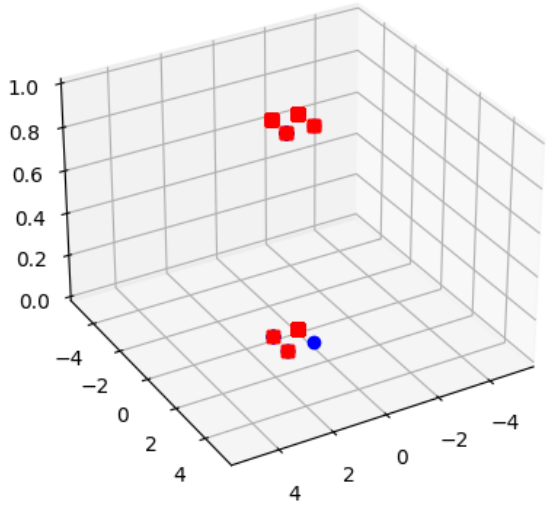

Fig 5.1: Support vectors and classes

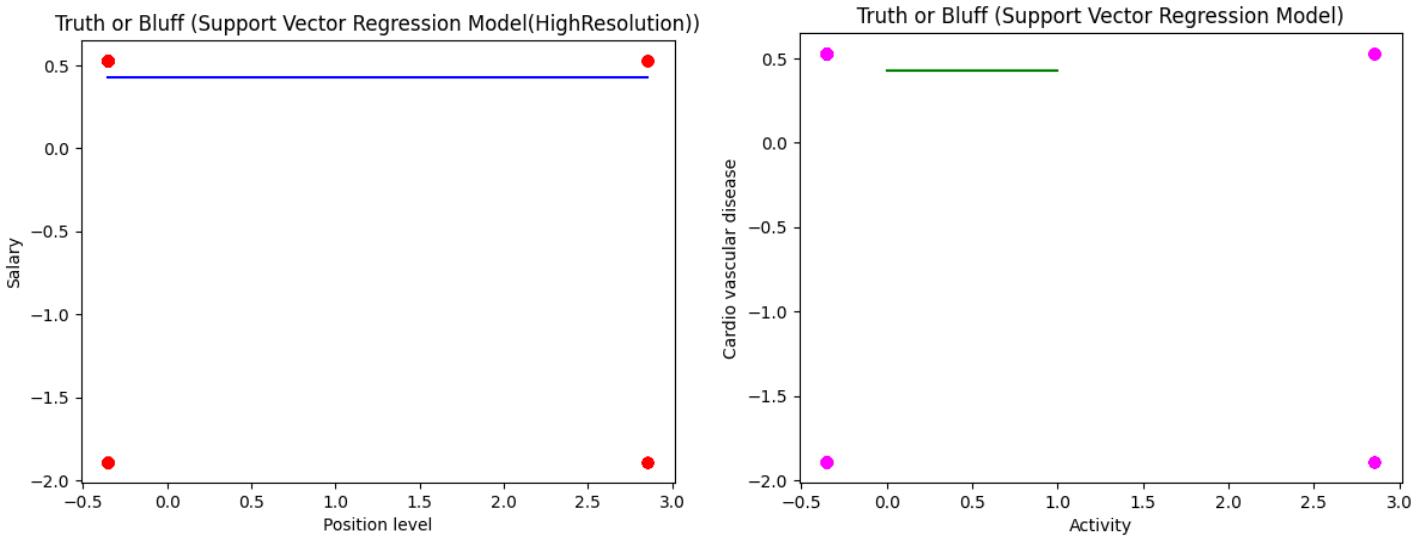

Fig 5.2: SVR

BE-ETRX B **Sub- AIML Lab**
Name: Shubham Sawant **UID :2019110050**

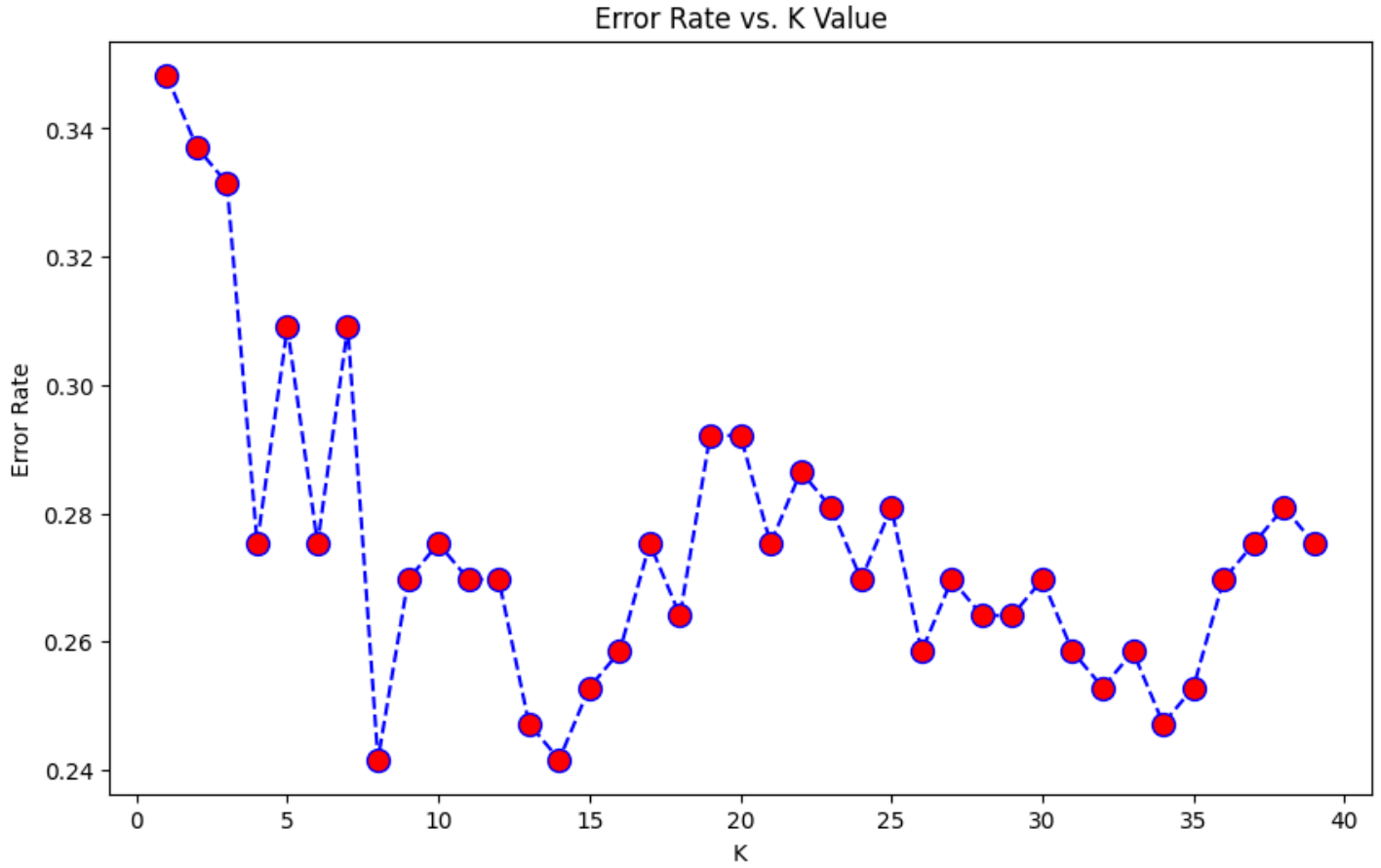

Fig 5.3: KNN Error Rate

Interpretation:

1. The accuracy for KNN is 73.03 % for the dataset.
2. The accuracy for Random Forest is 68.8% for the dataset.
3. The accuracy for SVM is 51%.
4. The graph of KNN for error vs K value is an elbow shaped curve.
5. The SVM into two separate classes in a 3D plane.

Conclusion:

1. The performance metrics for KNN is the highest, hence it is the ideal algorithm for the dataset.
2. The highest accuracy is observed at k =24 where the curve shows least error.
3. The second-best algorithm is Random Forest which gives good performance metrics compared to other algorithms.
4. SVM is a good algorithm for the dataset but does not predict correctly for many of the cases.
5. Here linear SVM has been used.
6. For SVR, Feature scaling has been done to give accurate results.