Experiment No.5

Title: Design and implement of SVM classification algorithm.

Batch: 4   **Roll No.: 16010420117**   **Experiment No.: 5**

Aim: Design and implement of SVM classification algorithm.

Resources needed: Python 3.6 onwards, RapidMiner

Theory:

A Bayesian classifier is a simple probabilistic classifier. Bayesian classifier can predict membership probabilities such as the probabilities that a sample belongs to a particular class or groupings.

Bayesian classification is based on Bayes theorem and this technique tends to be highly accurate and fast, making it useful on large databases. **SVM Classification Algorithm:**

More formally, a support-vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks like outliers detection.[3] Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin, the lower the generalization error of the classifier.[4]

Kernel machine

Whereas the original problem may be stated in a finite-dimensional space, it often happens that the sets to discriminate are not linearly separable in that space. For this reason, it was proposed that the original finite-dimensional space be mapped into a much higher- dimensional space, presumably making the separation easier in that space. To keep the computational load reasonable, the mappings used by SVM schemes are designed to ensure that dot products of pairs of input data vectors may be computed easily in terms of the variables in the original space, by defining them in terms of a kernel function selected to suit the problem. The hyperplanes in the higher-dimensional space are defined as the set of points whose dot product with a vector in that space is constant, where such a set of vectors is an orthogonal (and thus minimal) set of vectors that defines a hyperplane space.

Procedure / Approach /Algorithm / Activity Diagram:

1. Identify attributes suitable for applying classification algorithm
2. Implement SVM on your dataset using Python and RapidMiner.

3. Apply SVM to classify unknown tuple.

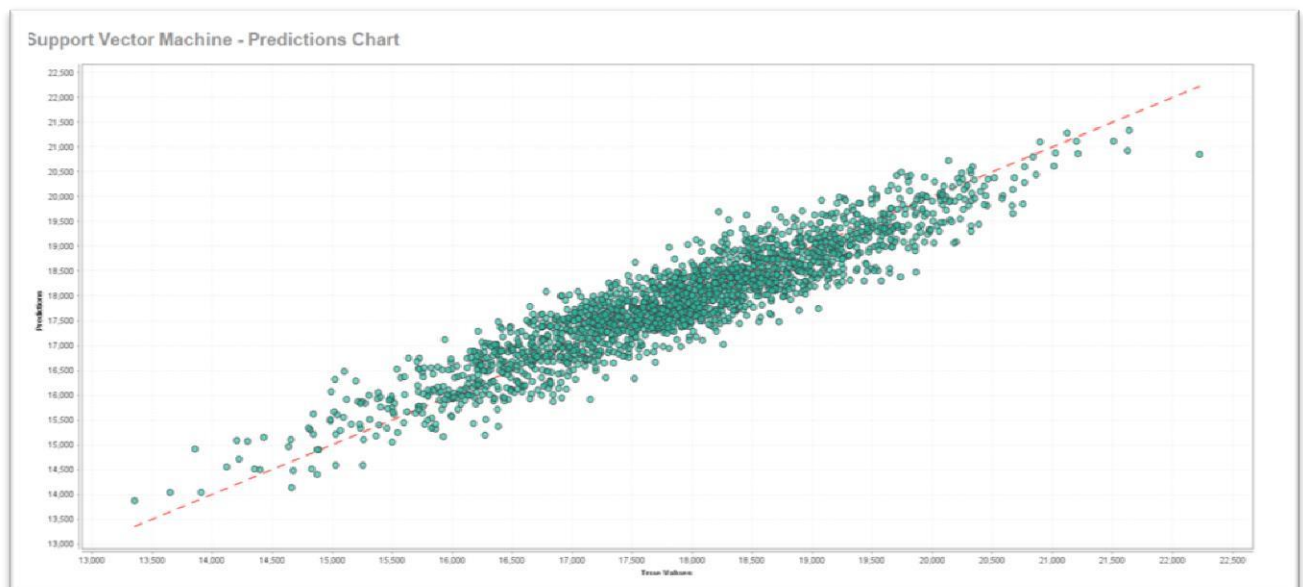**Results: (Program printout with output / Document printout as per the format)**
1.
The attributes selected:
- ' blueAvgLevel ',
- ' blueExperienceDiff ',
- ' blueGoldDiff ',
- ' blueTotalExperience '

2.
RapidMiner:

Python:
Code:

```python
import pandas as pd import
numpy as np import
matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split from
sklearn import svm
from sklearn.metrics import confusion_matrix, accuracy_score

csv_file =
pd.read_csv('/Users/Soham/Downloads/SEM_5_College_Stuff/ML_Hon/high_diamond_ranked_10
min - Copy.csv')

X = csv_file[['blueAvgLevel', 'blueExperienceDiff']][0:500]
Y = csv_file['blueTotalExperience'][0:500]
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20)

fig, axs = plt.subplots(1, 1, figsize =(15, 10), tight_layout = True)
for side in ['top', 'bottom', 'left', 'right']:
axs.spines[side].set_visible(False)
axs.xaxis.set_ticks_position('none')
axs.yaxis.set_ticks_position('none')  axs.xaxis.set_tick_params(pad
= 5)  axs.yaxis.set_tick_params(pad = 10)
axs.grid(b = True, color ='grey', linestyle ='-.', linewidth = 0.5, alpha = 0.6)

plt.scatter(csv_file[['blueExperienceDiff']], csv_file['blueTotalExperience'], s = 500,
edgecolor='black')
plt.scatter(csv_file[['blueGoldDiff']], csv_file['blueTotalExperience'], s = 500, edgecolor='black')

plt.xlabel('X') plt.ylabel('Y')
plt.title('SVM Predictions') plt.show()
```
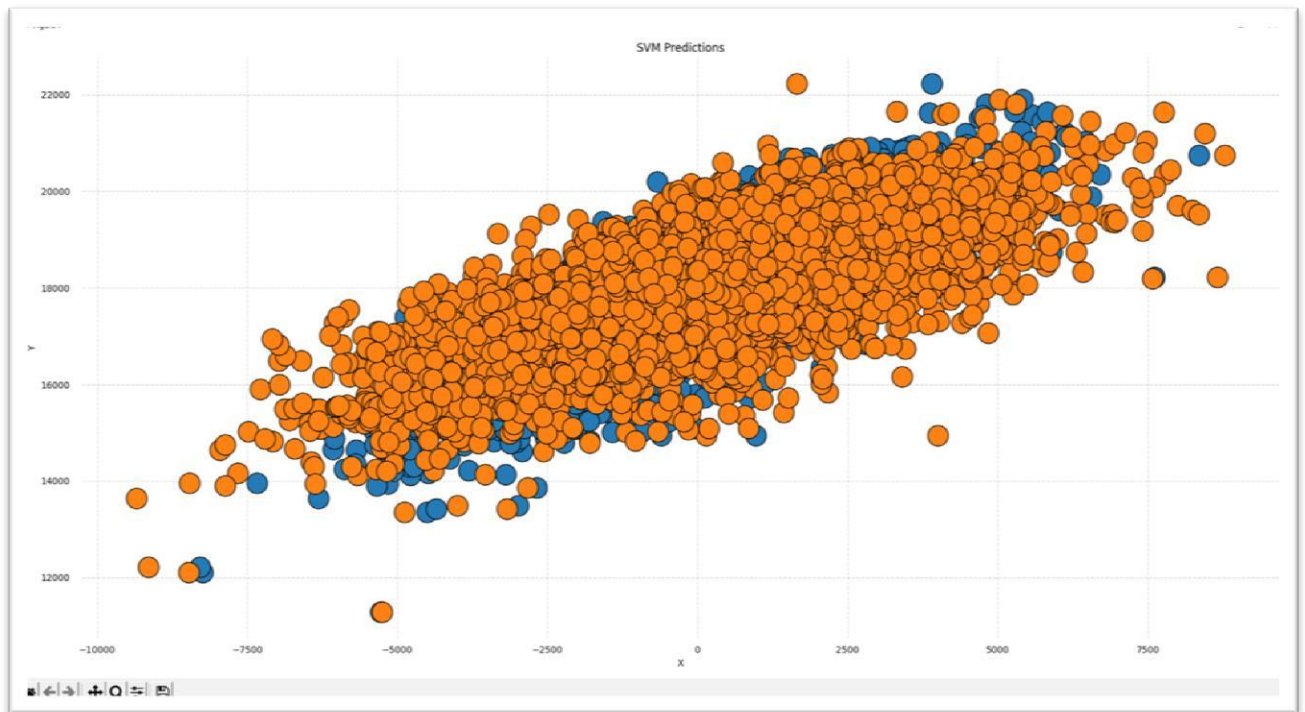
Output:

3.

Code:

```
   import pandas as pd
import numpy as np import
matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn import svm from sklearn.svm import
SVC
from sklearn.metrics import confusion_matrix, accuracy_score from
sklearn.metrics import classification_report, confusion_matrix

csv_file = pd.read_csv('high_diamond_ranked_10min - Copy.csv')

X = csv_file[['blueAvgLevel', 'blueExperienceDiff']]
Y = csv_file['blueFirstBlood']
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.20)
svclassifier = SVC(kernel='linear') svclassifier.fit(X_train, Y_train)
Y_pred = svclassifier.predict(X_test) print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
```

Output:
Confusion Matrix:

Classification Report:


```
              precision    recall  f1-score   support

           0       0.61      0.60      0.60        52
           1       0.57      0.58      0.58        48

    accuracy                           0.59       100
   macro avg       0.59      0.59      0.59       100
weighted avg       0.59      0.59      0.59       100

>>>
```

Questions:

1. What are advantages and disadvantages of SVM?

   Advantages:
   1. SVM works relatively well when there is a clear margin of separation between classes.
   2. SVM is more effective in high dimensional spaces.
   3. SVM is effective in cases where the number of dimensions is greater than the number of samples.
   4. SVM is relatively memory efficient

   Disadvantages:
   1. SVM algorithm is not suitable for large data sets.
   2. SVM does not perform very well when the data set has more noise i.e. target classes are overlapping.
   3. In cases where the number of features for each data point exceeds the number of training data samples, the SVM will underperform.
   4. As the support vector classifier works by putting data points, above and below the classify.

2. Explain over fitting problem in Machine Learning.

   Overfitting happens when a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new data. This means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model

Outcomes:

CO3: Comprehend radial-basis-function (RBF) networks and Kernel learning method
Conclusion: (Conclusion to be based on the objectives and outcomes achieved)

We understood the concept of classifier and classification and various classification algorithms. We also implemented SVM classification algorithm in rapid miner as well as python code for our dataset.

Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

1. Han, Kamber, "Data Mining Concepts and Techniques", Morgan Kaufmann 3nd Edition

(Autonomous College Affiliated to University of Mumbai)