FINANCE - CLASSIFICATION

S.no	Algorithm	Accuracy	Type-I error	Percent
1	Logistic Algorithm	0.78	1181	78%
2	SVM Algorithm	0.50	0	50%
3	Decision Tree Algorithm	1.00	9	100%
4	Random Forest Algorithm	1.00	1	100%
5	Kth-Nearest Neighbor Algorithm	0.98	82	98%
6	Naives' Bayes Algorithm(BernoulliNB)	0.96	177	96%

Though Decision Tree and Random Forest Classification Algorithm gives 100% Accuracy, We choose Random Forest because of the Low Type I Error (i.e. 1 for RF and 9 for DT). Hence, We can choose the Random Forest Classification Algorithm for this problem statement.

1. Logistic Regression Algorithm:

```
[[1541 1181]
 [  29 2725]]
```

```
from sklearn.metrics import classification_report
cr=classification_report(Y_test,Y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       0.98      0.57      0.72      2722
           1       0.70      0.99      0.82      2754

    accuracy                           0.78      5476
   macro avg       0.84      0.78      0.77      5476
weighted avg       0.84      0.78      0.77      5476
```

Accuracy for LogisticRegression = 0.78

2. SVM algorithm:

```
[[2722    0]
 [2740   14]]
```

```
from sklearn.metrics import classification_report
cr=classification_report(Y_test,Y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       0.50      1.00      0.67      2722
           1       1.00      0.01      0.01      2754

    accuracy                           0.50      5476
   macro avg       0.75      0.50      0.34      5476
weighted avg       0.75      0.50      0.34      5476
```

Accuracy for SupportVectorMachine = 0.50

3. Decision Tree Algorithm:

```
[[2713    9]
 [   6 2748]]
```

```
from sklearn.metrics import classification_report
cr=classification_report(Y_test,Y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2722
           1       1.00      1.00      1.00      2754

    accuracy                           1.00      5476
   macro avg       1.00      1.00      1.00      5476
weighted avg       1.00      1.00      1.00      5476
```

Accuracy for DecisionTreeClassification Algorithm = 1.00

4. Random Forest Algorithm:

```
[[2721    1]
 [   9 2745]]
```

```
from sklearn.metrics import classification_report
cr=classification_report(Y_test,Y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      2722
           1       1.00      1.00      1.00      2754

    accuracy                           1.00      5476
   macro avg       1.00      1.00      1.00      5476
weighted avg       1.00      1.00      1.00      5476
```

Accuracy for RandomForestClassification Algorithm = 1.00

5. Kth- Nearest Neighbor Algorithm:

```
[[2640   82]
 [  49 2705]]
```

```
from sklearn.metrics import classification_report
cr=classification_report(Y_test,Y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       0.98      0.97      0.98      2722
           1       0.97      0.98      0.98      2754

    accuracy                           0.98      5476
   macro avg       0.98      0.98      0.98      5476
weighted avg       0.98      0.98      0.98      5476
```

Accuracy for Kth- Nearest Neighbor Algorithm = 0.98

6. Naives' Bayes Algorithm:

```
[[2545  177]
 [  64 2690]]
```

```
from sklearn.metrics import classification_report
cr=classification_report(Y_test,Y_pred)
print(cr)
```

```
              precision    recall  f1-score   support

           0       0.98      0.93      0.95      2722
           1       0.94      0.98      0.96      2754

    accuracy                           0.96      5476
   macro avg       0.96      0.96      0.96      5476
weighted avg       0.96      0.96      0.96      5476
```

Accuracy for Naives' bayes Algorithm (Done in BernoulliNB) = 0.96