ted2-2

June 7, 2017

1 Exercice 2: Part 2

2 Import the necessary libraries and open the data set

```
In [10]: import pandas as pd
         import numpy as np
         from sklearn.model_selection import KFold
         from sklearn.tree import DecisionTreeClassifier
         from sklearn.ensemble import RandomForestClassifier
         from sklearn.cross_validation import train_test_split, ShuffleSplit, cross_val_score
         from sklearn import preprocessing
         from sklearn.naive_bayes import MultinomialNB
         from sklearn import svm
         from sklearn.metrics import accuracy_score
         import matplotlib.pyplot as plt
         import csv
         import numpy

         df = pd.read_csv('train.tsv', sep='\t')
         target = df["Label"]
```

3 Convert Categorical To Numerical

```
In [11]: categories = ["Attribute1","Attribute3","Attribute4","Attribute6","Attribute7","Attribu

         new_df = df.copy()
         target
         for x in categories:
             converted = pd.Categorical(df[x])
             new_df[x] = converted.codes
         my_df = new_df.copy()
         exclude=["Id", "Label"]
         dfne =df.columns.difference(exclude)
         new_df = new_df[dfne]
```

4 SVM

In [12]:
```
kf = KFold(n_splits=10)

RANDOM_STATE = 123

accuracy = 0.0

for x,y in kf.split(new_df):
    test = new_df.loc[y,new_df.columns]
    train = new_df.loc[x, new_df.columns]
    current_target = target[x]

    classifier = svm.LinearSVC(random_state=RANDOM_STATE)
    clf=classifier.fit(train, current_target)
    yPred = clf.predict(test)
    acc = accuracy_score(target[y], yPred)
    accuracy += acc

accuracySVM = accuracy/10
print(accuracySVM)
```

```
0.65
```

5 RANDOM FOREST CLASSIFICATION

In [13]:
```
kf = KFold(n_splits=10)

RANDOM_STATE = 123

accuracy = 0.0

for x,y in kf.split(new_df):
    test = new_df.loc[y,new_df.columns]
    train = new_df.loc[x, new_df.columns]
    current_target = target[x]

    classifier= RandomForestClassifier(random_state=RANDOM_STATE)
    clf = classifier.fit(train,current_target)
    yPred = clf.predict(test)
    acc = accuracy_score(target[y], yPred)
    accuracy += acc

accuracyRF = accuracy/10
print(accuracyRF)
```

```
0.75
```

6 Naive Bayes

In [14]:
```
kf = KFold(n_splits=10)

accuracy = 0.0

for x,y in kf.split(new_df):
    test = new_df.loc[y,new_df.columns]
    train = new_df.loc[x, new_df.columns]
    current_target = target[x]

    classifier= MultinomialNB()
    clf = classifier.fit(train,current_target)
    yPred = clf.predict(test)
    acc = accuracy_score(target[y], yPred)
    accuracy += acc

accuracyNB = accuracy/10
print(accuracyNB)
```

```
0.6375
```

7 We will use the 3rd part of the exercise to create the TestSetCategories.csv but we could do that here by executing the following code:

In [15]:
```
# with open('EvalutionMetric_10fold.csv', 'w') as csvfile5:
#     fieldnames = ['Statistic Measure', 'SVM', 'Random Forest', 'Naive Bayes']
#     writer = csv.DictWriter(csvfile5, fieldnames = fieldnames)
#     writer.writeheader()
#     measure = 'Accuracy'
#     writer.writerow({'Statistic Measure' : measure,  'SVM': accuracySVM , 'Random For
# dfT = pd.read_csv('test.tsv', sep='\t')
# #target = dfT["Id"]
# categories = ["Attribute1","Attribute3","Attribute4","Attribute6","Attribute7","Attri
# new_dfT = dfT.copy()

# IDs = dfT["Id"]
# for x in categories:
#     converted = pd.Categorical(dfT[x])
#     new_dfT[x] = converted.codes

# exclude=["Id"]
# dfneT =dfT.columns.difference(exclude)
# new_dfT = new_dfT[dfneT]
```

```
# category_dict = {1:'Good', 2:'Bad'}

# classifier = RandomForestClassifier(warm_start=True, max_features="sqrt", random_stat
# classifier.fit(new_df, target)
# prediction = classifier.predict(new_dfT)
# id =0
# with open('testSet_categories_RandomForest.csv', 'w') as csvfile4:
#     fieldnames = ['ID', 'Label']
#     writer = csv.DictWriter(csvfile4, fieldnames = fieldnames)
#     writer.writeheader()
#     for i in range(len(prediction)):
#         writer.writerow({'ID': IDs[i], 'Label': category_dict[prediction[i]]})
#         id +=1
# print('Created testSet_categories_RandomForest.csv')
```

8 Conclusion

As we can see from the above accuracies, Random Forest is the Classifier with the best accuracy so we are going to use it on the 3rd Part of the Exercise!