

Pythoh Code of SVM

Rae YIngqi Zhang

```
from sklearn import preprocessing
from sklearn.svm import LinearSVC
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
import seaborn as sns
import pandas as pd
import matplotlib.pyplot as plt

filenameNum="/Users/raezh1/Documents/Georgetown/ANLY501/assignment_5new/files/Crop_recommendation1.csv"
CropDF = pd.read_csv(filenameNum)
print(CropDF.head())
```

Get rid of the quatitaiive data

```
CropDF = CropDF.drop(["humidity_level"], axis=1)

CropTrainDF, CropTestDF = train_test_split(CropDF, test_size=0.3)
```

Seperate labels from data

Test data

```
CropTestLabels = CropTestDF["label"] ## save labels
print(CropTestLabels)
CropTestDF = CropTestDF.drop(["label"], axis=1) ##drop labels
print(CropTestDF)
```

Train data

```
CropTrainLabels = CropTrainDF["label"] ## save labels
CropTrainDF = CropTrainDF.drop(["label"], axis=1) ##drop labels
```

SCALE ALL DATA to between 0 and 1

#from sklearn import preprocessing

```
x = CropTrainDF.values #returns a numpy array
min_max_scaler = preprocessing.MinMaxScaler()
x_scaled = min_max_scaler.fit_transform(x)
CropTrainDF_S = pd.DataFrame(x_scaled)
```

```
x2 = CropTestDF.values #returns a numpy array
min_max_scaler2 = preprocessing.MinMaxScaler()
x_scaled2 = min_max_scaler2.fit_transform(x2)
CropTestDF_S = pd.DataFrame(x_scaled2)
print(CropTestDF_S)
```

SVM

linear kernel

```
SVM_Model1 = LinearSVC(C=1)
SVM_Model1.fit(CropTrainDF_S, CropTrainLabels)
SVM_matrix = confusion_matrix(CropTestLabels, SVM_Model1.predict(CropTestDF_S))
labels = SVM_Model1.classes_.tolist()
```

make a function to produce the heatmap of confusion matrix

```
def get_SVMprediction_confusionmatrix_heatmap(mapname, matrix):
    print("\nThe confusion matrix is:")
    print(matrix)
    print("\n\n")

    matrix_df = pd.DataFrame(matrix)
    matrix_df.columns = labels
    labels.reverse()
    matrix_df.index = labels
    print(matrix_df)

    sns.set(rc={'axes.facecolor':'#fbeed1', 'figure.facecolor':'#fbeed1'})
    fig = sns.heatmap(matrix_df, cmap="YlGnBu")
    plt.tight_layout()
    plt.title(mapname, fontname="ITC Officina Sans", fontweight="bold", color="#726abb")
    plt.xlabel("Prediction Labels", fontname="ITC Officina Sans", fontweight="bold", color="#726abb")
    plt.ylabel("Known Labels", fontname="ITC Officina Sans", fontweight="bold", color="#726abb")
    plt.xticks(fontname="ITC Officina Sans", color="#726abb")
    plt.yticks(fontname="ITC Officina Sans", color="#726abb")

    get_SVMprediction_confusionmatrix_heatmap("SVM Confusion Matrix Heatmap Linear C=1", SVM_matrix)
```

Other Kernels

Radial Basis Function

```
SVM_Model2 = sklearn.svm.SVC(C=1, kernel='rbf', degree=3, gamma="auto")
SVM_Model2.fit(CropTrainDF_S, CropTrainLabels)

SVM_matrix2 = confusion_matrix(CropTestLabels, SVM_Model2.predict(CropTestDF_S))
get_SVMprediction_confusionmatrix_heatmap("SVM Confusion Matrix Heatmap RBF C=1", SVM_matrix2)
```

Polynomial

```
SVM_Model3 = sklearn.svm.SVC(C=100, kernel='poly', degree=2, gamma="auto")
SVM_Model3.fit(CropTrainDF_S, CropTrainLabels)
```

```
SVM_matrix3 = confusion_matrix(CropTestLabels, SVM_Model3.predict(CropTestDF_S))  
get_SVMprediction_confusionmatrix_heatmap("SVM Confusion Matrix Heatmap POLY C=100", SVM_matrix3)
```