yhdwl57xn

September 12, 2023

##Name - Indranil Bain ##Enrollment No. - 2020CSB039 ##Assignment No. - 4 (FOREST Cover)

6. Download the Forest Cover Type dataset (https://www.kaggle.com/uciml/forest-cover-type-dataset) and preprocess the dummy variables to create training, test, and development set. Reduce the train data size if the system is unable to process the whole dataset.

[16]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call
drive.mount("/content/drive", force_remount=True).
```

[17]:
```
import pandas as pd
BASE_PATH = '/content/drive/MyDrive/CSV Files - COLAB/covtype 2.csv'
cov_df = pd.read_csv(BASE_PATH)
cov_df
```

[17]:

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	\
0	2596	51	3	258	
1	2590	56	2	212	
2	2804	139	9	268	
3	2785	155	18	242	
4	2595	45	2	153	
…	…	…	…	…	
581007	2396	153	20	85	
581008	2391	152	19	67	
581009	2386	159	17	60	
581010	2384	170	15	60	
581011	2383	165	13	60	

	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Roadways	\
0	0	510	
1	-6	390	
2	65	3180	
3	118	3090	
4	-1	391	
…	…	…	

```
581007                            17                             108
581008                            12                              95
581009                             7                              90
581010                             5                              90
581011                             4                              67

        Hillshade_9am  Hillshade_Noon  Hillshade_3pm  \
0                 221             232            148
1                 220             235            151
2                 234             238            135
3                 238             238            122
4                 220             234            150
…                 …               …              …
581007            240             237            118
581008            240             237            119
581009            236             241            130
581010            230             245            143
581011            231             244            141

        Horizontal_Distance_To_Fire_Points  …  Soil_Type32  Soil_Type33  \
0                                     6279  …            0            0
1                                     6225  …            0            0
2                                     6121  …            0            0
3                                     6211  …            0            0
4                                     6172  …            0            0
…                                        …  …            …            …
581007                                 837  …            0            0
581008                                 845  …            0            0
581009                                 854  …            0            0
581010                                 864  …            0            0
581011                                 875  …            0            0

        Soil_Type34  Soil_Type35  Soil_Type36  Soil_Type37  Soil_Type38  \
0                 0            0            0            0            0
1                 0            0            0            0            0
2                 0            0            0            0            0
3                 0            0            0            0            0
4                 0            0            0            0            0
…                 …            …            …            …            …
581007            0            0            0            0            0
581008            0            0            0            0            0
581009            0            0            0            0            0
581010            0            0            0            0            0
581011            0            0            0            0            0

        Soil_Type39  Soil_Type40  Cover_Type
0                 0            0           5
```

```
1                 0            0           5
2                 0            0           2
3                 0            0           2
4                 0            0           5
…               …            …         …
581007            0            0           3
581008            0            0           3
581009            0            0           3
581010            0            0           3
581011            0            0           3

[581012 rows x 55 columns]
```

[18]:
```
cov_df.columns
```

[18]:
```
Index(['Elevation', 'Aspect', 'Slope', 'Horizontal_Distance_To_Hydrology',
       'Vertical_Distance_To_Hydrology', 'Horizontal_Distance_To_Roadways',
       'Hillshade_9am', 'Hillshade_Noon', 'Hillshade_3pm',
       'Horizontal_Distance_To_Fire_Points', 'Wilderness_Area1',
       'Wilderness_Area2', 'Wilderness_Area3', 'Wilderness_Area4',
       'Soil_Type1', 'Soil_Type2', 'Soil_Type3', 'Soil_Type4', 'Soil_Type5',
       'Soil_Type6', 'Soil_Type7', 'Soil_Type8', 'Soil_Type9', 'Soil_Type10',
       'Soil_Type11', 'Soil_Type12', 'Soil_Type13', 'Soil_Type14',
       'Soil_Type15', 'Soil_Type16', 'Soil_Type17', 'Soil_Type18',
       'Soil_Type19', 'Soil_Type20', 'Soil_Type21', 'Soil_Type22',
       'Soil_Type23', 'Soil_Type24', 'Soil_Type25', 'Soil_Type26',
       'Soil_Type27', 'Soil_Type28', 'Soil_Type29', 'Soil_Type30',
       'Soil_Type31', 'Soil_Type32', 'Soil_Type33', 'Soil_Type34',
       'Soil_Type35', 'Soil_Type36', 'Soil_Type37', 'Soil_Type38',
       'Soil_Type39', 'Soil_Type40', 'Cover_Type'],
      dtype='object')
```

[19]:
```
from sklearn.preprocessing import StandardScaler
def standardize(df: "pd.DataFrame", col_name: "str") -> "pd.DataFrame":
    scaler = StandardScaler()
    df[[col_name]] = pd.DataFrame(
        data=scaler.fit_transform(df[[col_name]]),
        index=df.index,
        columns=[col_name]
    )
    return df
```

[20]:
```
_columns_to_scale = ['Elevation', 'Aspect',␣
 ↪'Slope','Horizontal_Distance_To_Hydrology','Vertical_Distance_To_Hydrology','Horizontal_Dis
 ↪'Hillshade_Noon', 'Hillshade_3pm','Horizontal_Distance_To_Fire_Points']
for _col in _columns_to_scale:
    cov_df = standardize(cov_df, _col)
```

```
cov_df
```

[20]:

	Elevation	Aspect	Slope	Horizontal_Distance_To_Hydrology	\
0	-1.297805	-0.935157	-1.482820	-0.053767	
1	-1.319235	-0.890480	-1.616363	-0.270188	
2	-0.554907	-0.148836	-0.681563	-0.006719	
3	-0.622768	-0.005869	0.520322	-0.129044	
4	-1.301377	-0.988770	-1.616363	-0.547771	
…	…	…	…	…	
581007	-2.012130	-0.023740	0.787408	-0.867697	
581008	-2.029988	-0.032675	0.653865	-0.952383	
581009	-2.047847	0.029873	0.386780	-0.985317	
581010	-2.054990	0.128163	0.119694	-0.985317	
581011	-2.058562	0.083486	-0.147392	-0.985317	

	Vertical_Distance_To_Hydrology	Horizontal_Distance_To_Roadways	\
0	-0.796273	-1.180146	
1	-0.899197	-1.257106	
2	0.318742	0.532212	
3	1.227908	0.474492	
4	-0.813427	-1.256464	
…	…	…	
581007	-0.504653	-1.437962	
581008	-0.590424	-1.446299	
581009	-0.676194	-1.449506	
581010	-0.710502	-1.449506	
581011	-0.727656	-1.464256	

	Hillshade_9am	Hillshade_Noon	Hillshade_3pm	\
0	0.330743	0.439143	0.142960	
1	0.293388	0.590899	0.221342	
2	0.816364	0.742654	-0.196691	
3	0.965786	0.742654	-0.536343	
4	0.293388	0.540313	0.195215	
…	…	…	…	
581007	1.040496	0.692069	-0.640851	
581008	1.040496	0.692069	-0.614724	
581009	0.891075	0.894409	-0.327327	
581010	0.666942	1.096749	0.012325	
581011	0.704298	1.046164	-0.039929	

	Horizontal_Distance_To_Fire_Points	…	Soil_Type32	Soil_Type33	\
0	3.246283	…	0	0	
1	3.205504	…	0	0	
2	3.126965	…	0	0	
3	3.194931	…	0	0	
4	3.165479	…	0	0	

```
…                                    …  …          …           …
581007                         -0.863386  …          0           0
581008                         -0.857345  …          0           0
581009                         -0.850548  …          0           0
581010                         -0.842997  …          0           0
581011                         -0.834690  …          0           0

        Soil_Type34  Soil_Type35  Soil_Type36  Soil_Type37  Soil_Type38  \
0                 0            0            0            0            0
1                 0            0            0            0            0
2                 0            0            0            0            0
3                 0            0            0            0            0
4                 0            0            0            0            0
…               …            …            …            …            …
581007            0            0            0            0            0
581008            0            0            0            0            0
581009            0            0            0            0            0
581010            0            0            0            0            0
581011            0            0            0            0            0

        Soil_Type39  Soil_Type40  Cover_Type
0                 0            0           5
1                 0            0           5
2                 0            0           2
3                 0            0           2
4                 0            0           5
…               …            …           …
581007            0            0           3
581008            0            0           3
581009            0            0           3
581010            0            0           3
581011            0            0           3

[581012 rows x 55 columns]
```

[21]:
```
cov_df[['Cover_Type']].value_counts()
```

[21]:
```
Cover_Type
2             283301
1             211840
3              35754
7              20510
6              17367
5               9493
4               2747
dtype: int64
```

[22]:
```
cov_df = cov_df.sample(frac=0.1)
X = cov_df.drop('Cover_Type', axis=1)
y = cov_df[['Cover_Type']]
```

[23]:
```
y.value_counts()
```

[23]:
```
Cover_Type
2             28291
1             21257
3              3574
7              2041
6              1700
5               951
4               287
dtype: int64
```

[24]:
```
from sklearn.model_selection import train_test_split
X_train, _X_rest, y_train, _y_rest = train_test_split(X, y, train_size=0.8)
X_val, X_val, y_val, y_val = train_test_split(_X_rest, _y_rest, train_size=0.5)
```

7. Train the one vs rest and one-vs-one SVM model on the above dataset for multiclass classification. Plot and Analyze the Confusion matrix for the above models. Show the accuracy in the graph. State the difference of the two approaches using the model parameters.

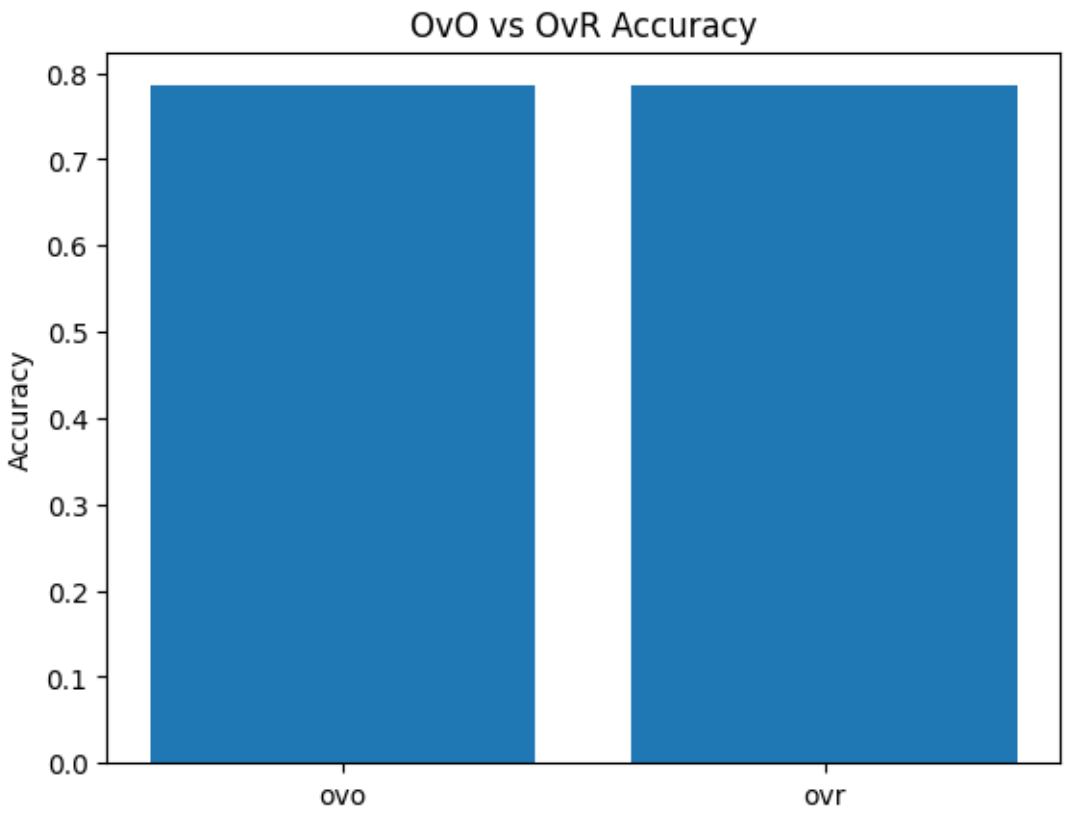
[25]:
```
from sklearn.svm import SVC
from sklearn.metrics import confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt

def display_confusion_matrix(X_test: "pd.DataFrame", y_test: "pd.DataFrame",
                                    model: "SVC"):
   y_predict = model.predict(X_test)
   matrix = confusion_matrix(y_test, y_predict)
   fig = plt.figure(figsize=(10,10))

   sns.heatmap(
       matrix,
       xticklabels=range(1,8),
       yticklabels=range(1,8),
       linewidth=0.5,
       cmap='coolwarm',
       annot=True,
       cbar=True,
       square=True)
   plt.title('HeatMap for the model')
   plt.ylabel('Actual Value')
```

```
    plt.xlabel('Predicted Value')
    plt.show()
```

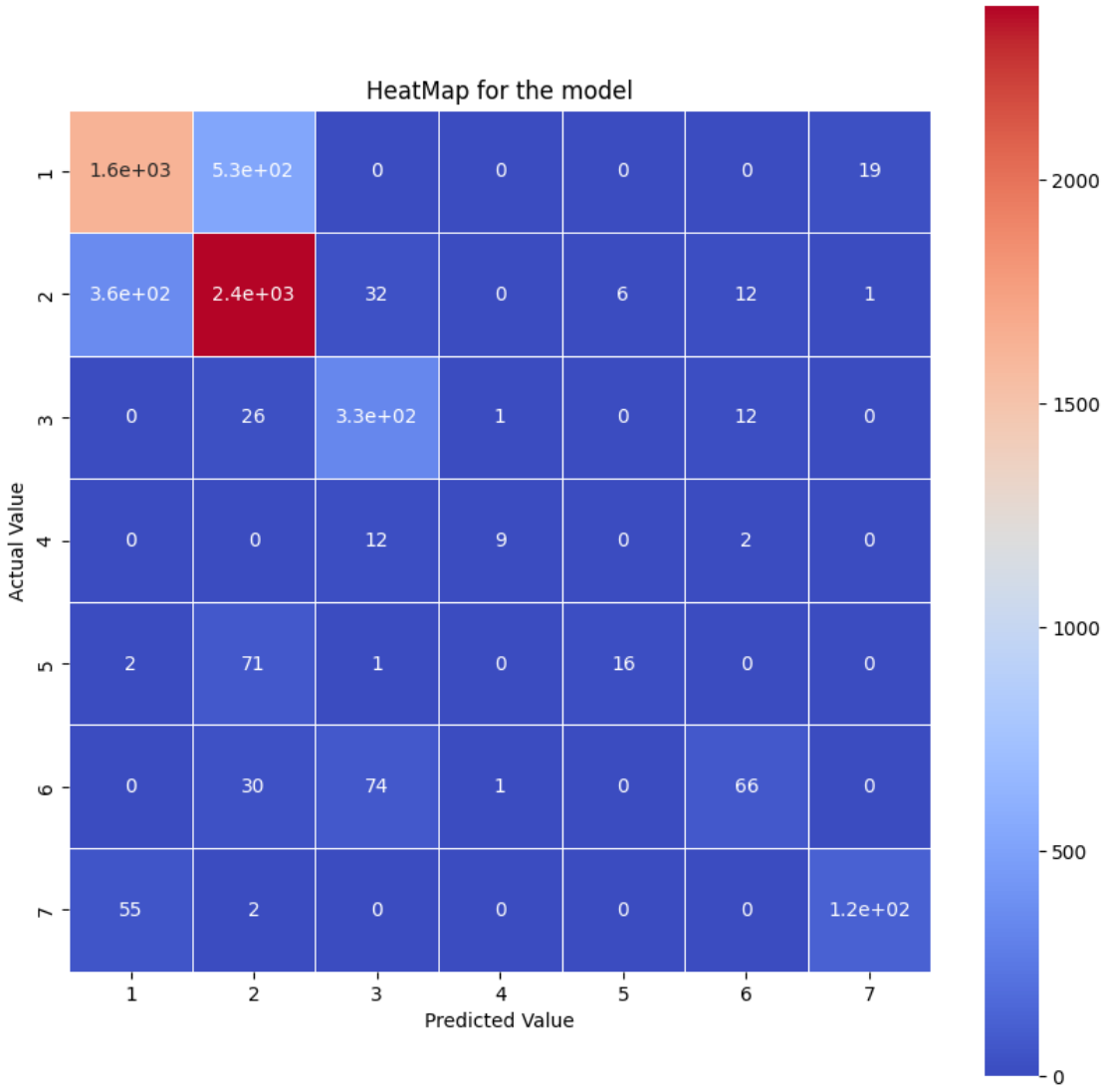
[26]:
```
decision_function_shapes = ['ovo', 'ovr']
models = [
    SVC(decision_function_shape=shape).fit(X_train, y_train.iloc[:, 0])
    for shape in decision_function_shapes
]
```

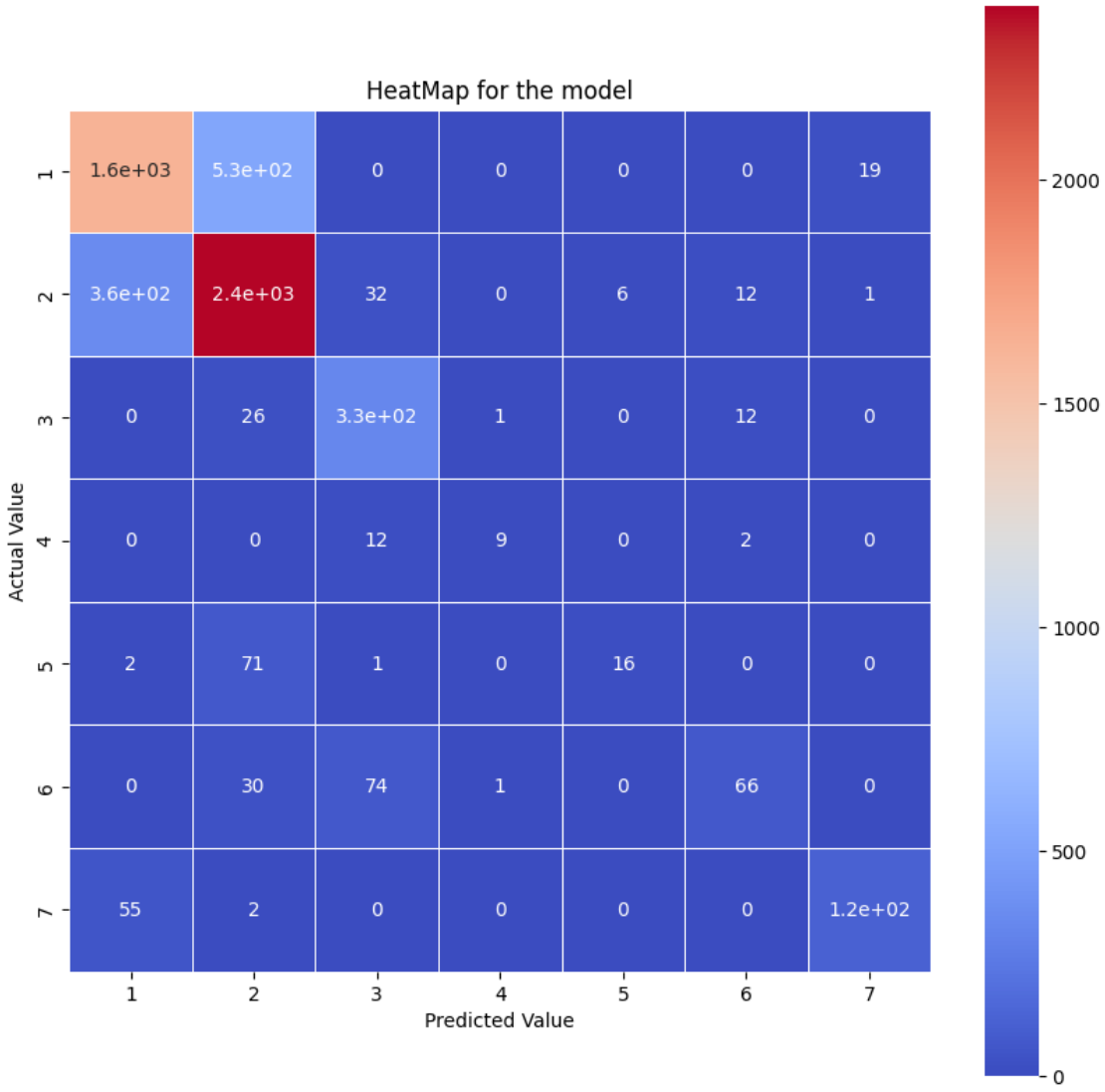
[27]:
```
# Accuracies
accuracies = [model.score(X_val, y_val) for model in models]

print(pd.DataFrame(columns=['decision_function_shape', 'Accuracy'],
             data=zip(decision_function_shapes, accuracies)))

plt.bar(range(0, len(accuracies)), accuracies)
plt.title('OvO vs OvR Accuracy')
plt.ylabel('Accuracy')
plt.xticks(ticks=[0,1], labels=['ovo', 'ovr'])
plt.show()
```

```
  decision_function_shape  Accuracy
0                     ovo  0.785235
1                     ovr  0.785235
```



[28]:
```
for model in models:
  display_confusion_matrix(X_val, y_val, model)
```


HeatMap for the model
1.6e+03
5.3e+02
0
0
0
0
19
3.6e+02
2.4e+03
32
0
6
12
1
0
26
3.3e+02
1
0
12
0
0
0
12
9
0
2
0
2
71
1
0
16
0
0
0
30
74
1
0
66
0
55
2
0
0
0
0
1.2e+02
Actual Value
Predicted Value
1
2
3
4
5
6
7
2000
1500
1000
500
0


HeatMap for the model
1.6e+03
5.3e+02
0
0
0
0
19
3.6e+02
2.4e+03
32
0
6
12
1
0
26
3.3e+02
1
0
12
0
0
0
12
9
0
2
0
2
71
1
0
16
0
0
0
30
74
1
0
66
0
55
2
0
0
0
0
1.2e+02
Actual Value
Predicted Value
1
2
3
4
5
6
7
2000
1500
1000
500
0