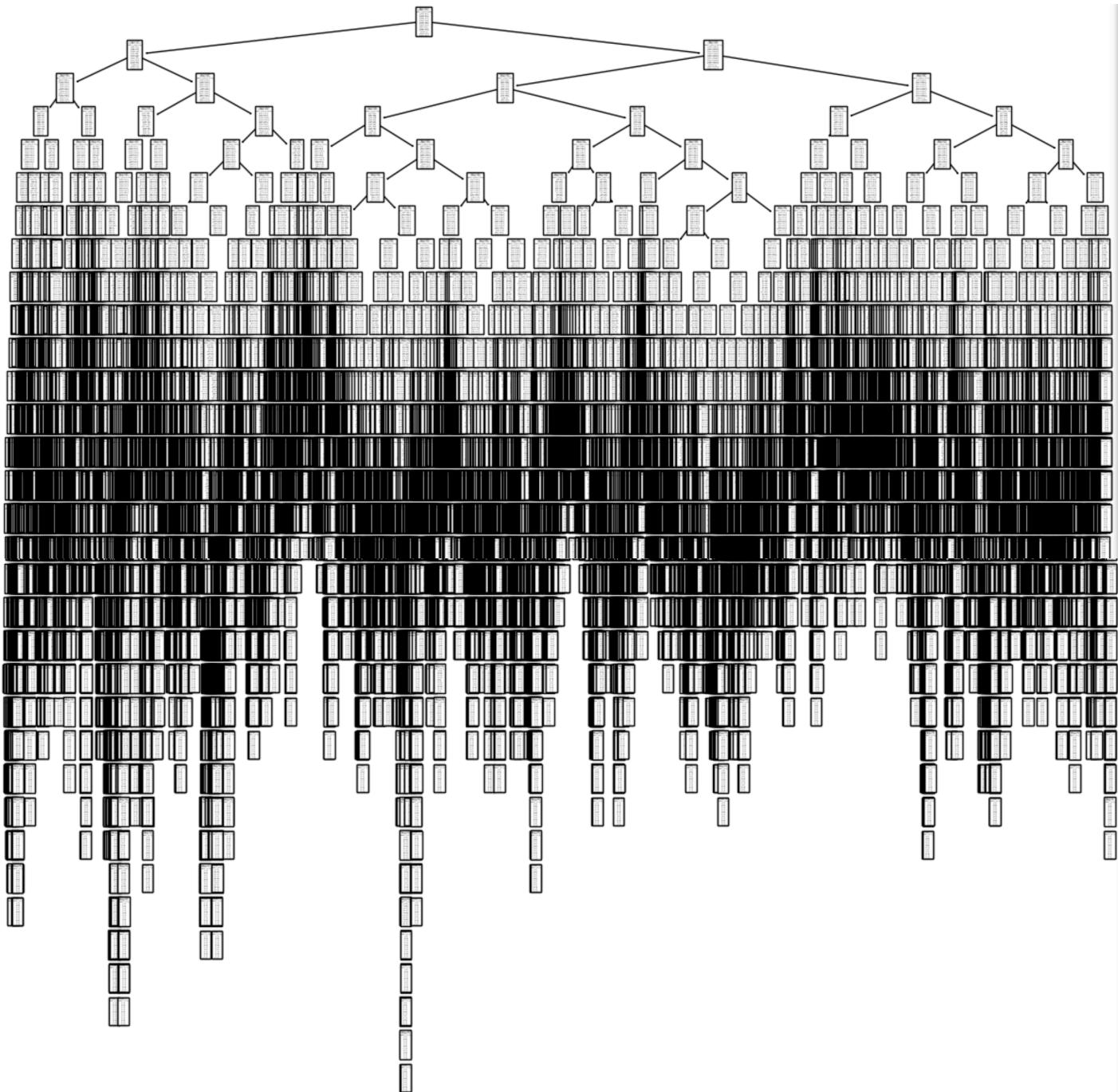


Kirsten Currie  
Career Foundry  
Machine Learning | 1.5  
2.17.25

## Decision Tree



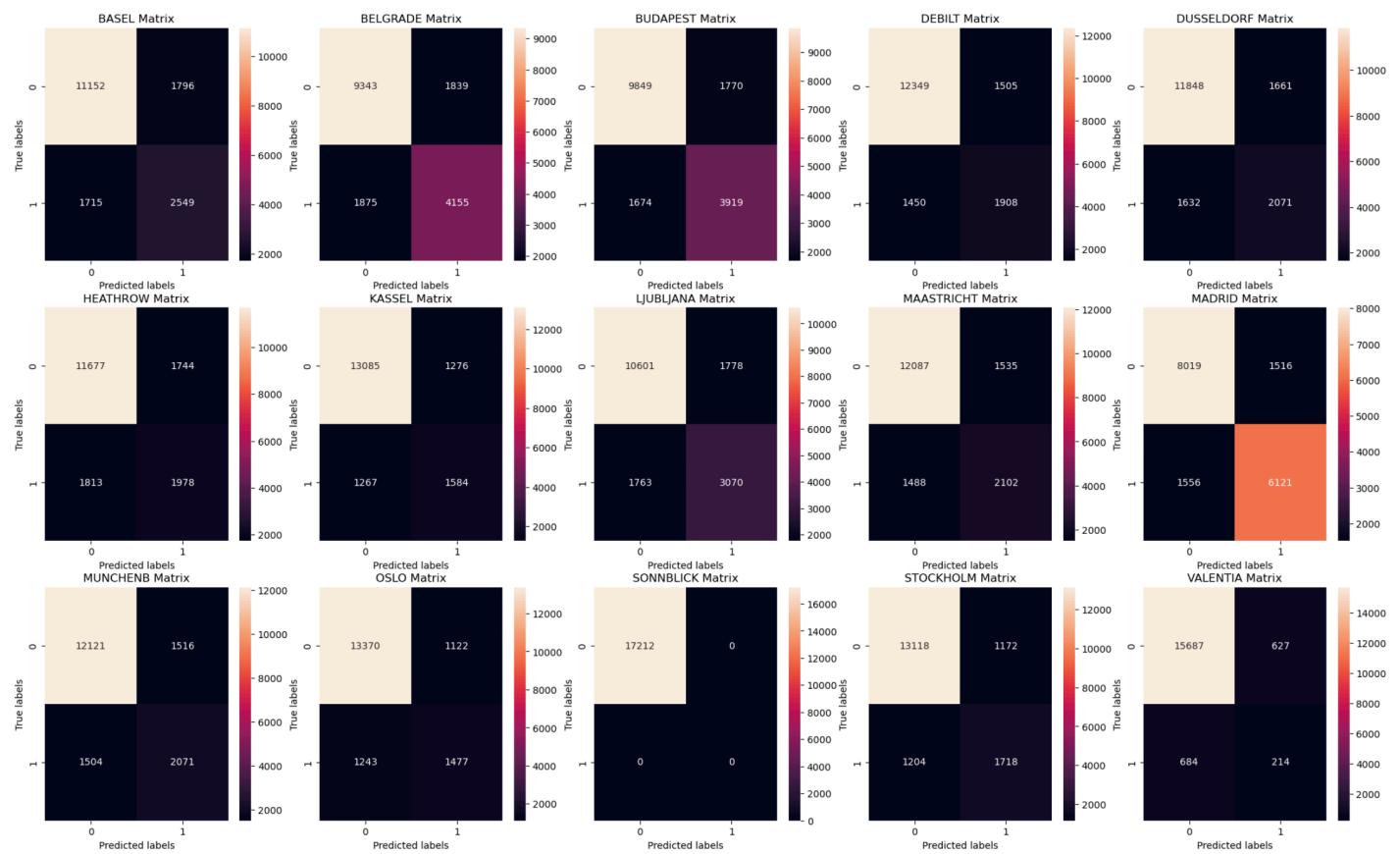
## Decision Tree Confusion Matrix - Train Data:

Train Accuracy Score: **39.8%**

Model Success F1 Score (micro avg): **63%**

**Would be advisable to prune tree as it's not decipherable in current form.**

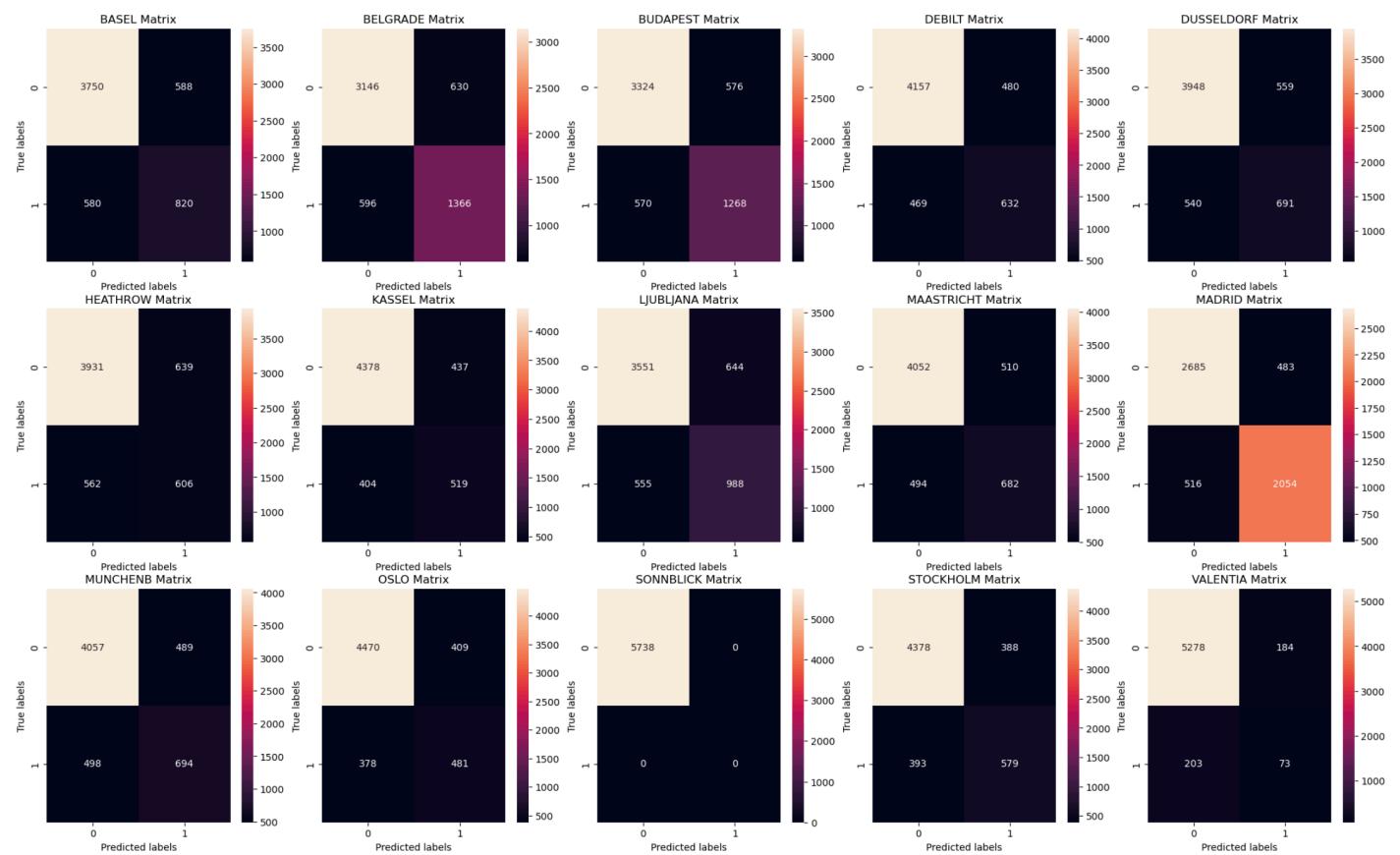
	precision	recall	f1-score	support
0	0.58	0.60	0.59	4264
1	0.69	0.69	0.69	6030
2	0.69	0.70	0.69	5593
3	0.56	0.57	0.56	3358
4	0.56	0.56	0.56	3703
5	0.53	0.52	0.52	3791
6	0.55	0.56	0.55	2851
7	0.64	0.64	0.64	4833
8	0.57	0.58	0.58	3590
9	0.80	0.80	0.80	7677
10	0.57	0.59	0.58	3575
11	0.56	0.55	0.56	2720
12	0.00	0.00	0.00	0
13	0.60	0.58	0.59	2922
14	0.24	0.24	0.24	898
micro avg	0.63	0.63	0.63	55805
macro avg	0.54	0.54	0.54	55805
weighted avg	0.63	0.63	0.63	55805
samples avg	0.36	0.36	0.34	55805



## Decision Tree Confusion Matrix - Test Data:

Train accuracy score: 0.39821041845477284

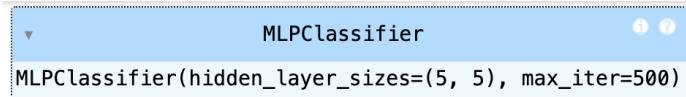
	precision	recall	f1-score	support
0	0.58	0.59	0.58	1400
1	0.68	0.70	0.69	1962
2	0.69	0.69	0.69	1838
3	0.57	0.57	0.57	1101
4	0.55	0.56	0.56	1231
5	0.49	0.52	0.50	1168
6	0.54	0.56	0.55	923
7	0.61	0.64	0.62	1543
8	0.57	0.58	0.58	1176
9	0.81	0.80	0.80	2570
10	0.59	0.58	0.58	1192
11	0.54	0.56	0.55	859
12	0.00	0.00	0.00	0
13	0.60	0.60	0.60	972
14	0.28	0.26	0.27	276
micro avg	0.62	0.63	0.62	18211
macro avg	0.54	0.55	0.54	18211
weighted avg	0.62	0.63	0.62	18211
samples avg	0.36	0.37	0.34	18211



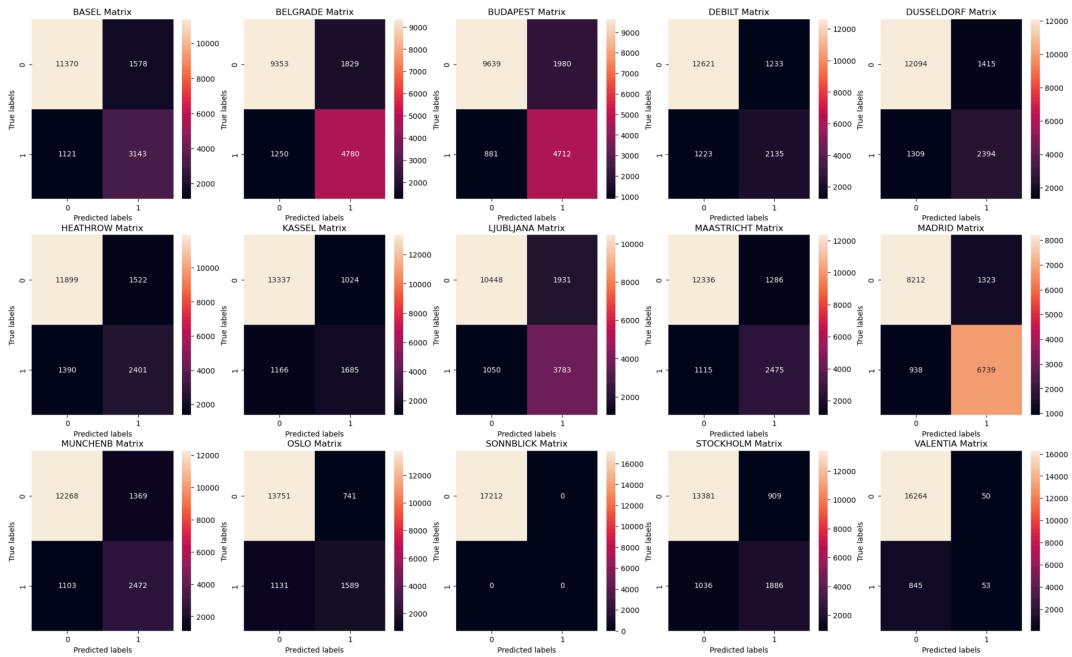
## ANN - 1st Iteration:

Train accuracy score: 0.43684638624215666

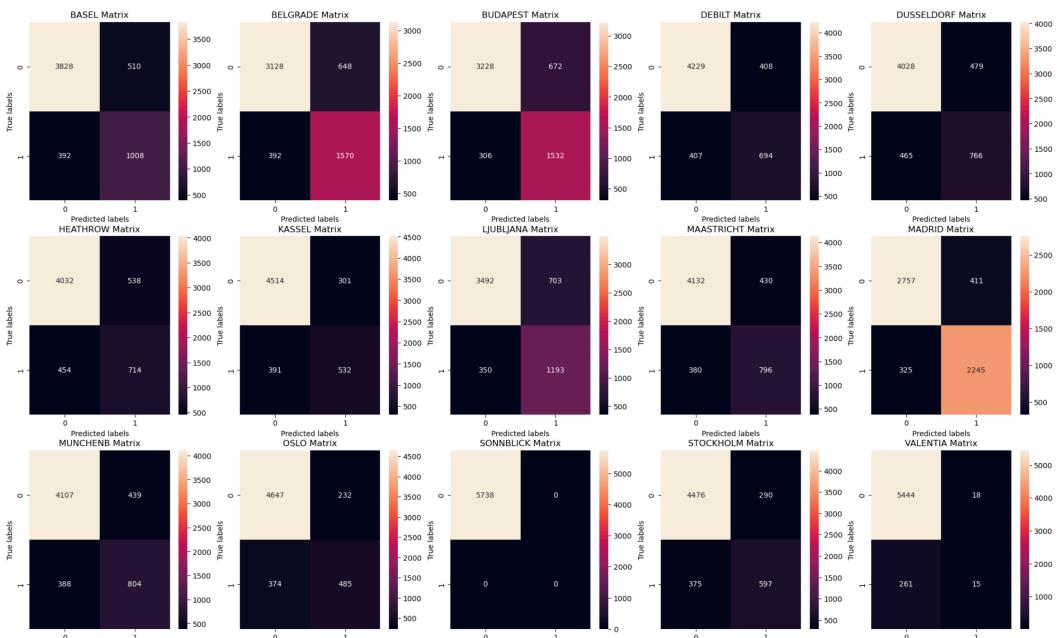
Test accuracy score: 0.439177413733008



Train Matrix:



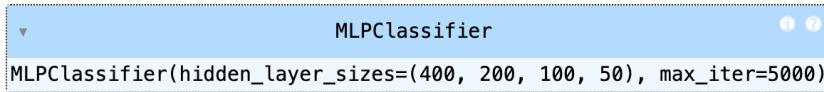
Test Matrix:



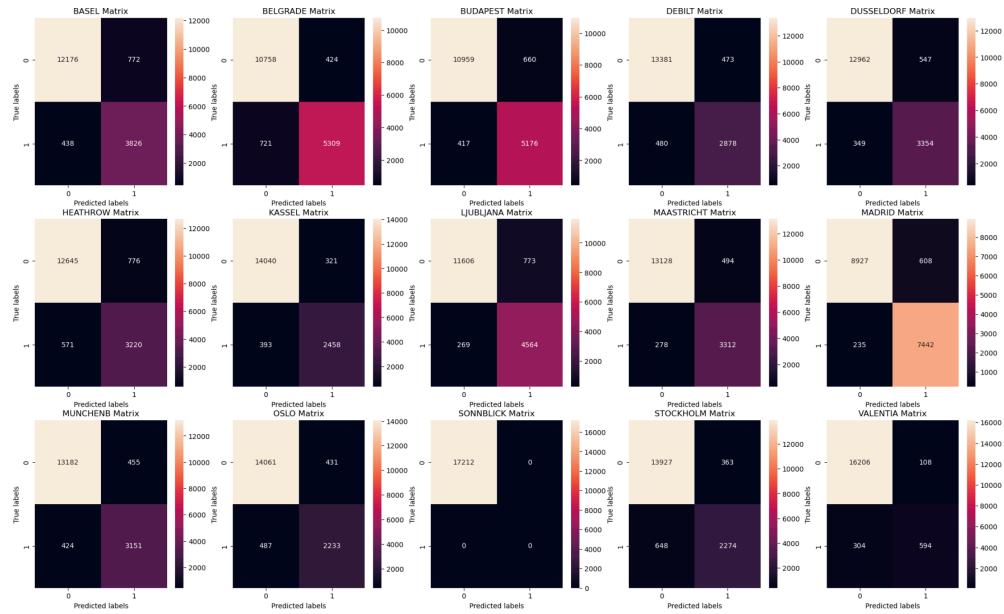
## ANN - 2nd Iteration:

Train accuracy score: 0.43684638624215666

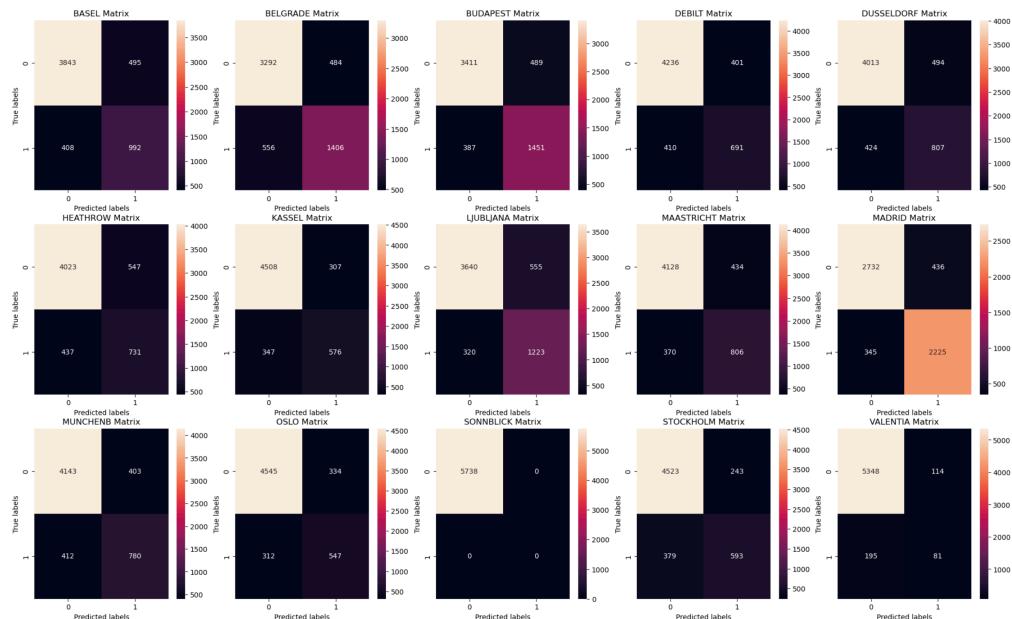
Test accuracy score: 0.42436388985709306



Train Matrix:



Test Matrix:



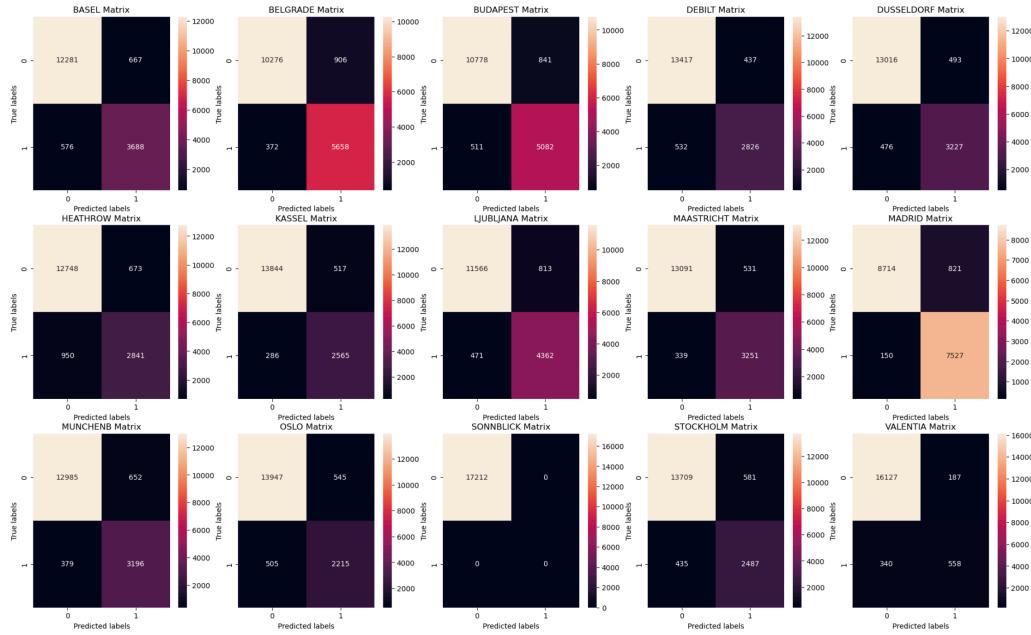
## ANN - 3rd Iteration:

Train accuracy score: 0.43684638624215666

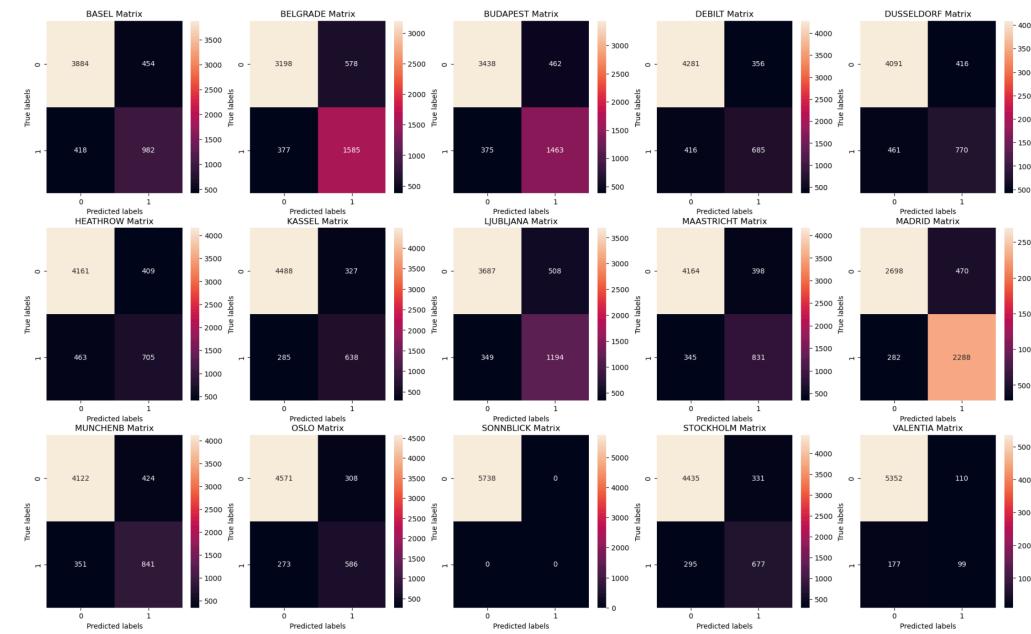
Test accuracy score: 0.4224468455907982

```
MLPClassifier
MLPClassifier(hidden_layer_sizes=(400, 200, 100, 50), learning_rate='adaptive',
max_iter=10000)
```

Train Matrix:



Test Matrix:



## ANN - 4th Iteration:

Train accuracy score: 0.43684638624215666

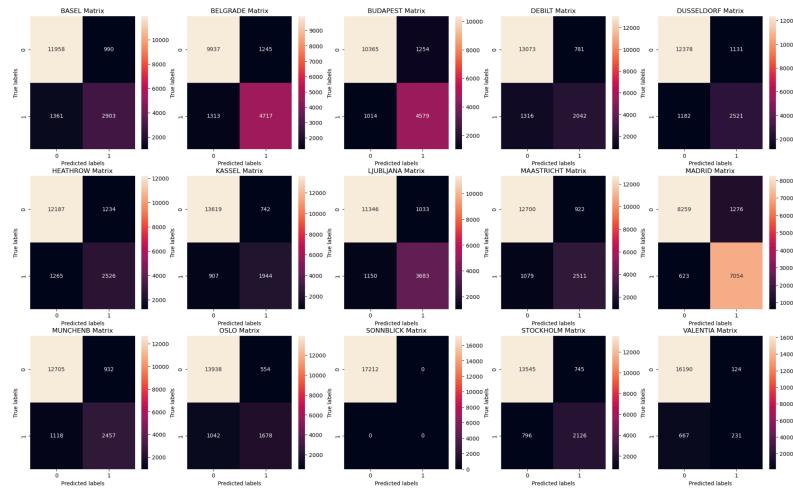
Test accuracy score: 0.4578250261415127

```
MLPClassifier
MLPClassifier(hidden_layer_sizes=(35, 35), learning_rate='adaptive',
max_iter=500)
```

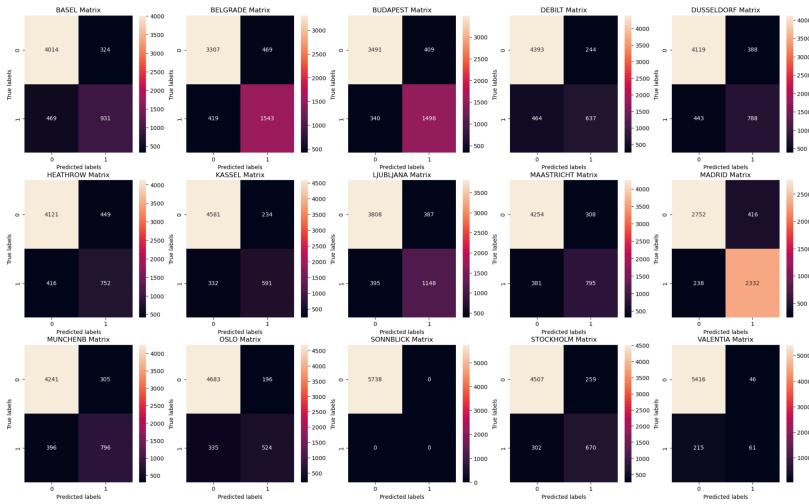
## Classification Report:

Train Classification Report:				Test Classification Report:					
	precision	recall	f1-score		precision	recall	f1-score	support	
0	0.67	0.74	0.70	4264	0	0.74	0.67	0.70	1400
1	0.72	0.79	0.76	6030	1	0.77	0.79	0.78	1962
2	0.70	0.84	0.77	5593	2	0.79	0.82	0.80	1838
3	0.63	0.64	0.63	3358	3	0.72	0.58	0.64	1101
4	0.63	0.65	0.64	3783	4	0.67	0.64	0.65	1231
5	0.61	0.63	0.62	3791	5	0.63	0.64	0.63	1168
6	0.62	0.59	0.61	2851	6	0.72	0.64	0.68	923
7	0.66	0.78	0.72	4833	7	0.75	0.74	0.75	1543
8	0.66	0.69	0.67	3590	8	0.72	0.68	0.70	1176
9	0.84	0.88	0.86	7677	9	0.85	0.91	0.88	2570
10	0.64	0.69	0.67	3575	10	0.72	0.67	0.69	1192
11	0.68	0.58	0.63	2720	11	0.73	0.61	0.66	859
12	0.00	0.00	0.00	0	12	0.00	0.00	0.00	0
13	0.67	0.65	0.66	2922	13	0.72	0.69	0.70	972
14	0.51	0.06	0.11	898	14	0.57	0.22	0.32	276
micro avg	0.69	0.72	0.70	55805	micro avg	0.75	0.72	0.73	18211
macro avg	0.62	0.61	0.60	55805	macro avg	0.67	0.62	0.64	18211
weighted avg	0.69	0.72	0.70	55805	weighted avg	0.74	0.72	0.73	18211
samples avg	0.39	0.39	0.37	55805	samples avg	0.43	0.41	0.40	18211

## Train Matrix:



## Test Matrix:



## Observations:

### 1. Model Performance:

KNN:

```
For k = 1:  
    Training Accuracy: 1.0000  
    Testing Accuracy: 0.4089  
  
For k = 2:  
    Training Accuracy: 0.5370  
    Testing Accuracy: 0.4230  
  
For k = 3:  
    Training Accuracy: 0.5270  
    Testing Accuracy: 0.4258  
  
Best model (k=3):  
    Best Training Accuracy: 0.5270  
    Best Testing Accuracy: 0.4258
```

Decision Tree:

**Decision Tree Confusion Matrix - Test Data:**  
Train accuracy score: 0.39821041845477284

ANN:

**ANN - 4th Iteration:**  
Train accuracy score: 0.43684638624215666  
Test accuracy score: 0.4578250261415127

```
MLPClassifier(hidden_layer_sizes=(35, 35), learning_rate='adaptive',  
max_iter=500)
```

From the three supervised models run, the KNN appears to have provided the most accurate scores. However, the ANN model closer, but there was some discrepancy between the train/test scores and the confusion matrix false predictions. ANN & KNN Classification Report F1 scores were comparable, with Train scores being slightly better for KNN (ANN - 70% Train, 73% Test; KNN - 83% Train, 68% Test).

**So to summarize, KNN appears to be the better model of the three, and would be recommended for ClimateWise.**

**2. Overfitting / Full Accuracy:** Sonnblick continues to be fully accurate in all models which raises concern for bias or feasibility of the datapoints for this station.