## 4.3 Trip Purpose Classification Models

4.3.1:

To better account for the spatial and temporal irregularity and dependency (identified in 4.2), we explicitly divide the data into 5 unique subsets and these are detailed in **Table 4.8**.

***Table 4.8*** *Subsets used in the building of classification models*

|  |  |  |
| --- | --- | --- |
| *Subset* | *N* | *% of total trips* |
| All Data | 71,801 | 100 |
| Rush Hour | 40,945 | 56.1 |
| Off-Peak | 32,084 | 43.9 |
| City | 68,275 | 93.5 |
| Non-city | 4,754 | 6.5 |

The results from the hyperparameter tuning are as follows

Support vector model built using the

'alpha': 0.01,

'hidden\_layer\_sizes': (50, 50, 50),

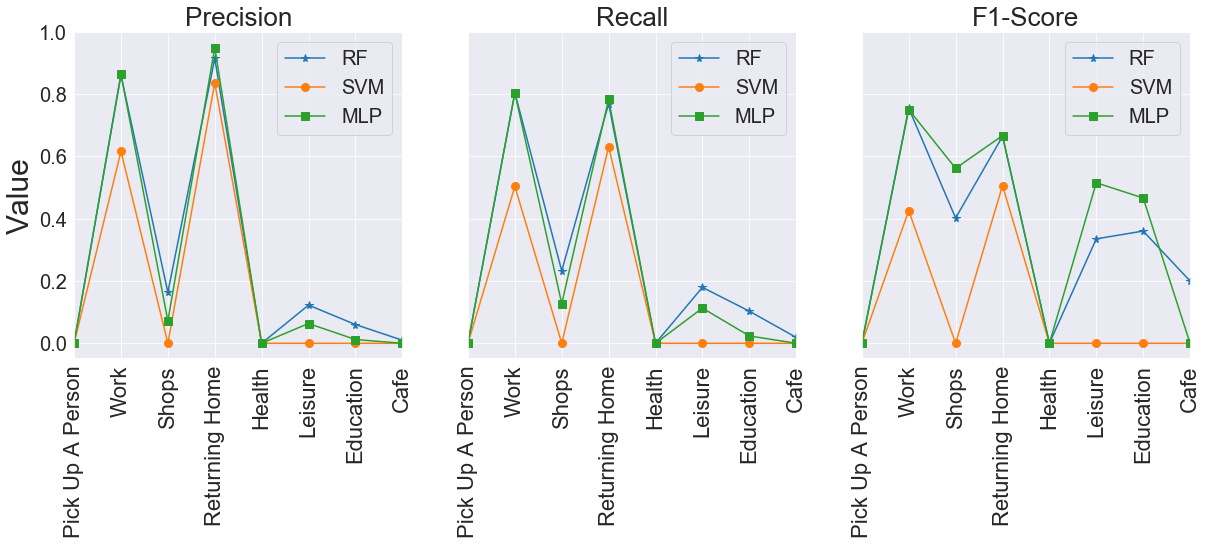
'max\_iter': 500,

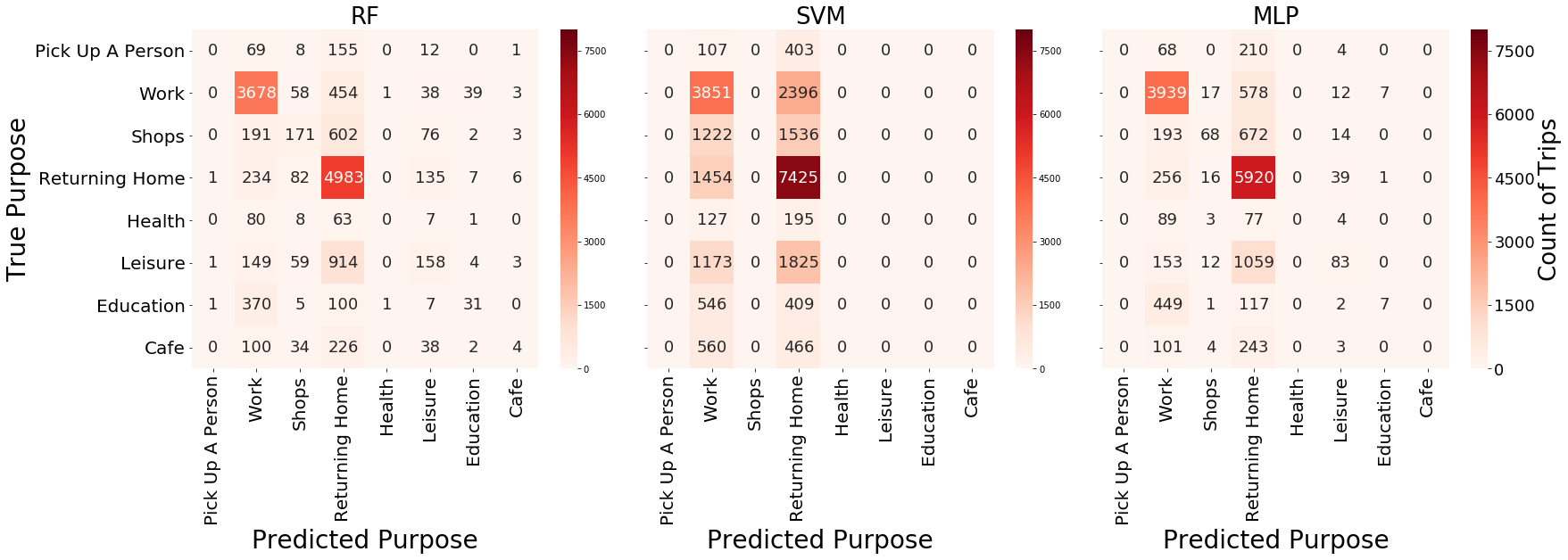
'solver': 'lbfgs'

Results:

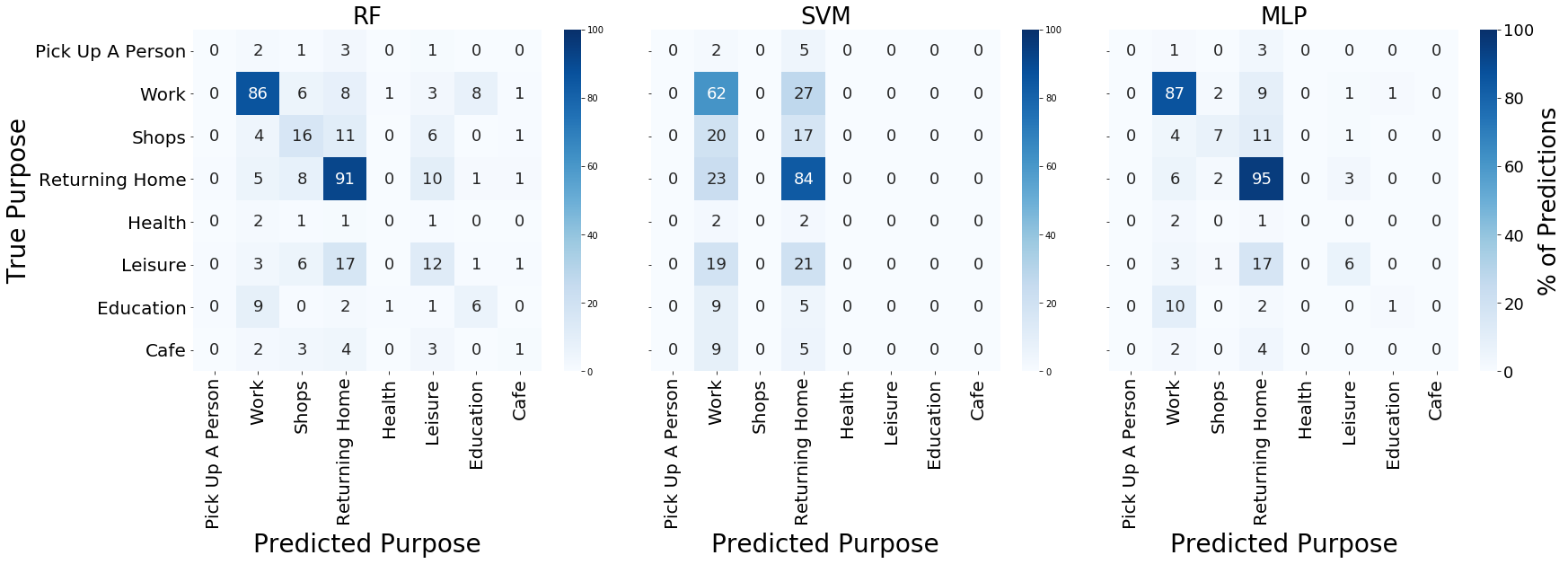
Overall Accuracy

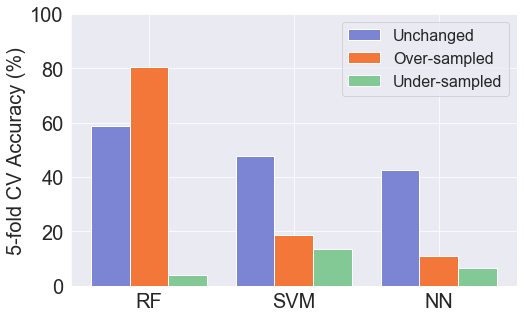
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 5-fold Cross-Validation (All) | | Prediction made | |
|  | Accuracy (%) | Range (%) | % of trips predicted | Accuracy |
| Random Forest | 0.588 | 0.9 | 56.4 | 0.67 |
| SVC | 0.476 | 1.2 | 100 | 0.48 |
| MLP | 0.427 | 1.5 | 60.8 | 0.69 |



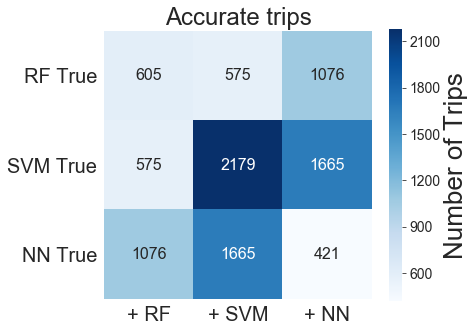


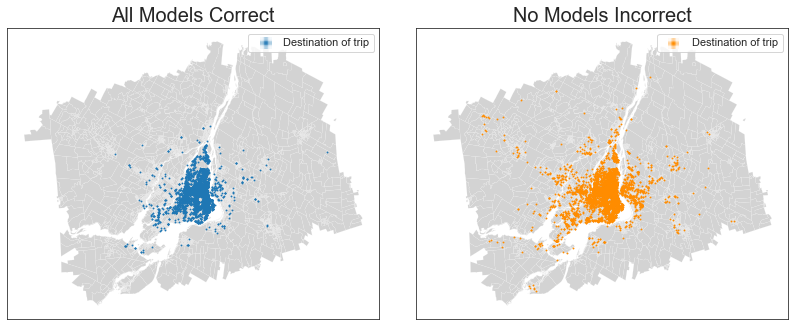
**Figure 4.X**





We find that all the models are accurate 6,857 trips and all inaccurate in 10,317

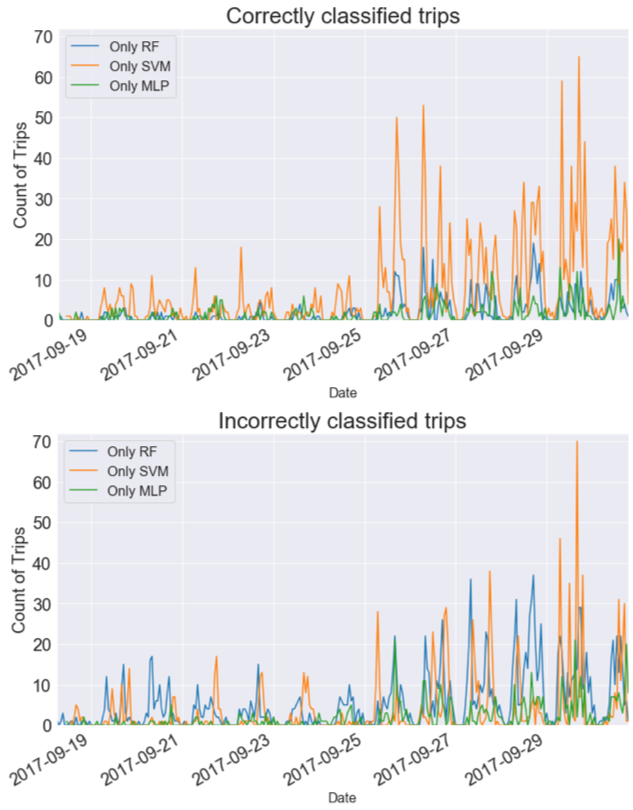




Temporality no true -3.083525711470756, 0.027796626120368234

All true -3.9825091284053635,

0.0015049193012149386,



For the purposes of the classification models, the data is normalised for the purpose of the

Cs = [0.1, 1, 10]

gammas = [0.01, 0.1, 1]

Best == {'C': 0.1, 'gamma': 0.01}

Best estimator NN:

ctivation='relu', alpha=0.1, batch\_size='auto', beta\_1=0.9,

beta\_2=0.999, early\_stopping=False, epsilon=1e-08,

hidden\_layer\_sizes=5, learning\_rate='constant',

learning\_rate\_init=0.001, max\_iter=500, momentum=0.9,

n\_iter\_no\_change=10, nesterovs\_momentum=True, power\_t=0.5,

random\_state=3, shuffle=True, solver='lbfgs', tol=0.0001,

validation\_fraction=0.1, verbose=False, warm\_start=False

tested for alpha == 1e-05, 0.0001, 0.05

'sgd', 'adam', 'lbfgs'

Alpha == learning rate

'alpha': 0.01,

'hidden\_layer\_sizes': (50, 50, 50),

'max\_iter': 500,

'solver': 'lbfgs'

*Classification of purpose of travel:*

Take 1st week -> predict then Take 2nd week -> predict then … compare accuracy scores

Sizes of data:

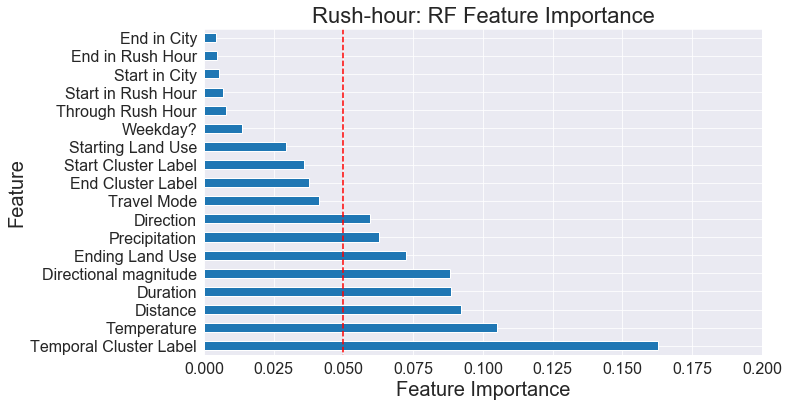
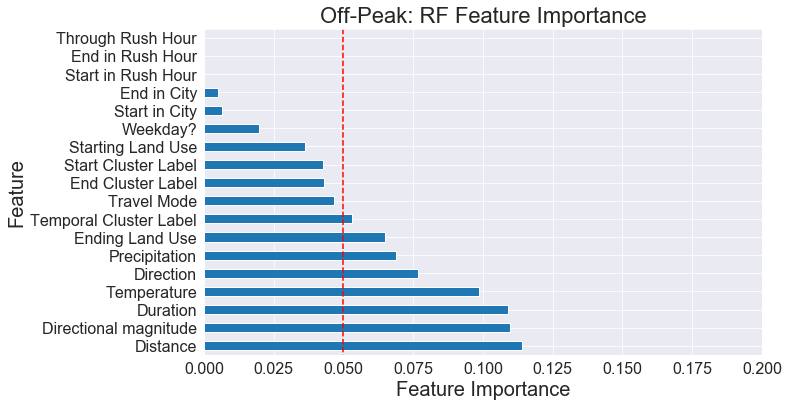
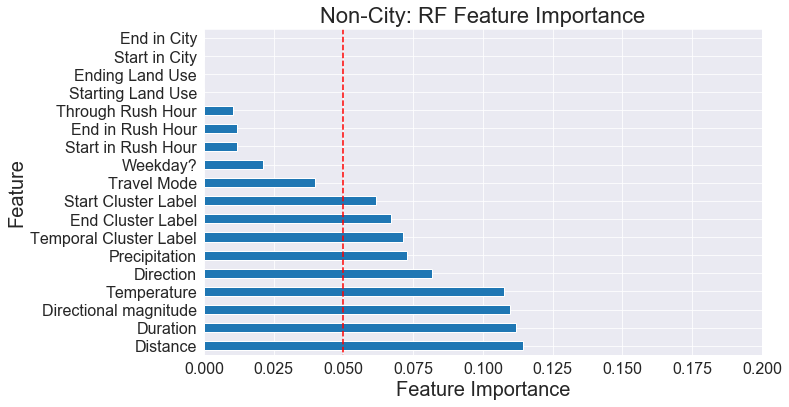
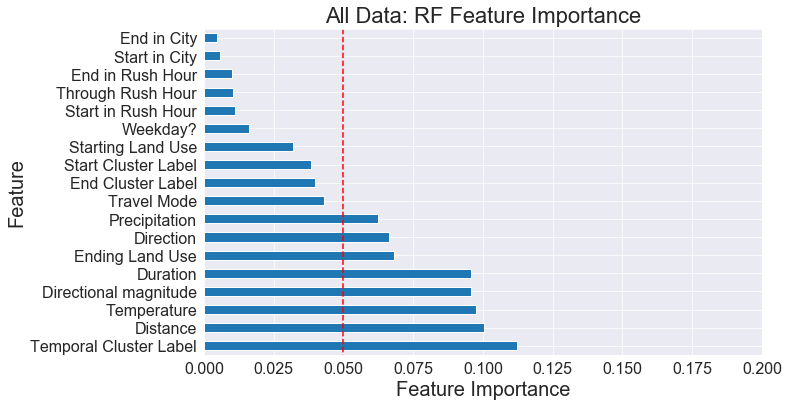
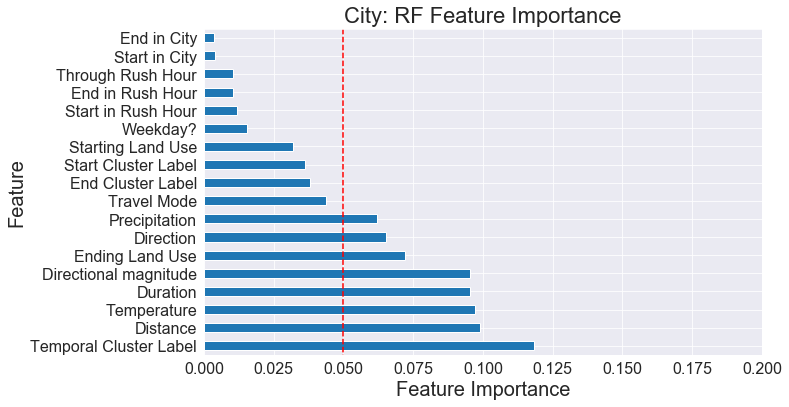
len(purp\_gdf), len(purp\_city), len(purp\_noncity), len(purp\_rush), len(purp\_nonrush), (71801, 66029, 5772, 39695, 32106)

*Random Forest:*

Random Forest:

ADD CONFUSION MATRIX OF PREDICTIONS FOR EACH CLASS WITH PROPER NAMES

* Feature importance
* Plot residuals



|  |  |  |  |
| --- | --- | --- | --- |
|  | RF |  | n |
| All | 0.8116480270099177 |  |  |
| City | 0.8131174162118273 |  |  |
| Non-City | 0.8086500655307994 |  |  |
| Rush | 0.8489446405768798 |  |  |
| Non-Rush | 0.7686380273550376 |  |  |

RF normal:

'purp\_gdf': array([0.80827185, 0.81544217, 0.8151112 , 0.81278586, 0.81742566]),

'purp\_city': array([0.81263883, 0.8150411 , 0.82103977, 0.80835463, 0.8143127 ]),

'purp\_noncity': array([0.80515298, 0.81320451, 0.80322581, 0.82200647, 0.82524272]),

'purp\_rush': array([0.84744822, 0.84464583, 0.84739179, 0.84680851, 0.84749213]),

'purp\_nonrush': array([0.76879298, 0.76944972, 0.76826376, 0.77100142, 0.77171334])

Cross-validated multi-class

{'purp\_gdf': array([0.73622947, 0.73838478, 0.73453903, 0.74275023, 0.74586841]),

'purp\_city': array([0.73694735, 0.74294601, 0.74016885, 0.72891901, 0.73780691]),

'purp\_noncity': array([0.70645161, 0.7516129 , 0.6983871 , 0.69789984, 0.73505654]),

'purp\_rush': array([0.79670736, 0.78838328, 0.7891232 , 0.80281169, 0.79278446]),

'purp\_nonrush': array([0.67220114, 0.66816888, 0.66500593, 0.67852906, 0.66476868])}

Predictions RF:

1.0 21108

0.0 2419

2.0 167

3.0 1

*Comparison of Models:*

Dimensions:

'purp\_gdf': (71801, 22),

'purp\_city': (67177, 22),

'purp\_noncity': (4624, 22),

'purp\_rush': (40342, 22),

'purp\_nonrush': (31459, 62)}

*SVC:*

cv\_scores\_rf

{'purp\_gdf': 0.7364844903988184,

'purp\_city': 0.7402228336866796,

'purp\_noncity': 0.7070773263433814,

'purp\_rush': 0.7959137684969578,

'purp\_nonrush': 0.6675014448083221}

cv\_scores\_svc

{'purp\_gdf': 0.6609411268200042,

'purp\_city': 0.6556903784564031,

'purp\_noncity': 0.7450851900393185,

'purp\_rush': 0.7543003079696537,

'purp\_nonrush': 0.7545752263533038}

cv\_scores\_nn

{'purp\_gdf': 0.7864950411479215,

'purp\_city': 0.7898867788353106,

'purp\_noncity': 0.7234600262123198,

'purp\_rush': 0.8369263126267558,

'purp\_nonrush': 0.7356963976112503}

num\_dims

{'purp\_gdf': (71801, 23),

'purp\_city': (67177, 23),

'purp\_noncity': (4624, 23),

'purp\_rush': (40342, 7),

'purp\_nonrush': (31459, 63)}

Hyper Parameter for SVC:

RF

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trip Purpose | Precision | recall | F1-score | support |
| 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| average |  |  |  |  |

NN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trip Purpose | Precision | recall | F1-score | support |
| 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| average |  |  |  |  |

NN

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Trip Purpose | Precision | recall | F1-score | support |
| 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| average |  |  |  |  |

ADD CONFUSION MATRIX OF PREDICTIONS FOR EACH CLASS WITH PROPER NAMES

Results on the test set: RF:

precision recall f1-score support

0.0 0.00 0.00 0.00 0

1.0 0.85 0.89 0.87 15661

2.0 0.00 0.00 0.00 955

3.0 0.00 0.00 0.00 322

4.0 1.00 0.00 0.00 510

5.0 0.76 0.57 0.65 6247

accuracy 0.74 23695

macro avg 0.43 0.24 0.25 23695

weighted avg 0.78 0.74 0.74 23695

Results on the test set: SVC:

precision recall f1-score support

1.0 0.66 1.00 0.80 15661

2.0 0.00 0.00 0.00 955

3.0 0.00 0.00 0.00 322

4.0 0.00 0.00 0.00 510

5.0 0.00 0.00 0.00 6247

accuracy 0.66 23695

macro avg 0.13 0.20 0.16 23695

weighted avg 0.44 0.66 0.53 23695

Results on the test set: NN:

precision recall f1-score support

0.0 0.00 0.00 0.00 0

1.0 0.83 0.93 0.88 15661

2.0 0.00 0.00 0.00 955

3.0 0.00 0.00 0.00 322

4.0 0.00 0.00 0.00 510

5.0 0.73 0.64 0.68 6247

accuracy 0.79 23695

macro avg 0.26 0.26 0.26 23695

weighted avg 0.74 0.79 0.76 23695

NN:

10,000 iterations/epochs for NN

ADD CONFUSION MATRIX OF PREDICTIONS FOR EACH CLASS WITH PROPER NAMES

Predictions NN:

1.0 23136

0.0 559