**Support Vector Machine(SVM) classifier**

* SVM classifier model works better on small dataset.
* We have selected simple ‘linear’ kernel to separate hyperplanes linearly ,can experiment with ‘rbf’ and ‘polynomial’ kernels as well.
* Below is the screenshot of the metrics captured from the SVM classifier model execution.

\*\*\*\*\*\*\*\*\*\*\*\*\*\*Printing training metrics\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[[268 13]

[ 16 160]]

precision recall f1-score support

False 0.94 0.95 0.95 281

True 0.92 0.91 0.92 176

avg / total 0.94 0.94 0.94 457

\*\*\*\*\*\*\*\*\*\*\*\*\*\*Printing testing metrics\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

[[62 4]

[ 5 44]]

precision recall f1-score support

False 0.93 0.94 0.93 66

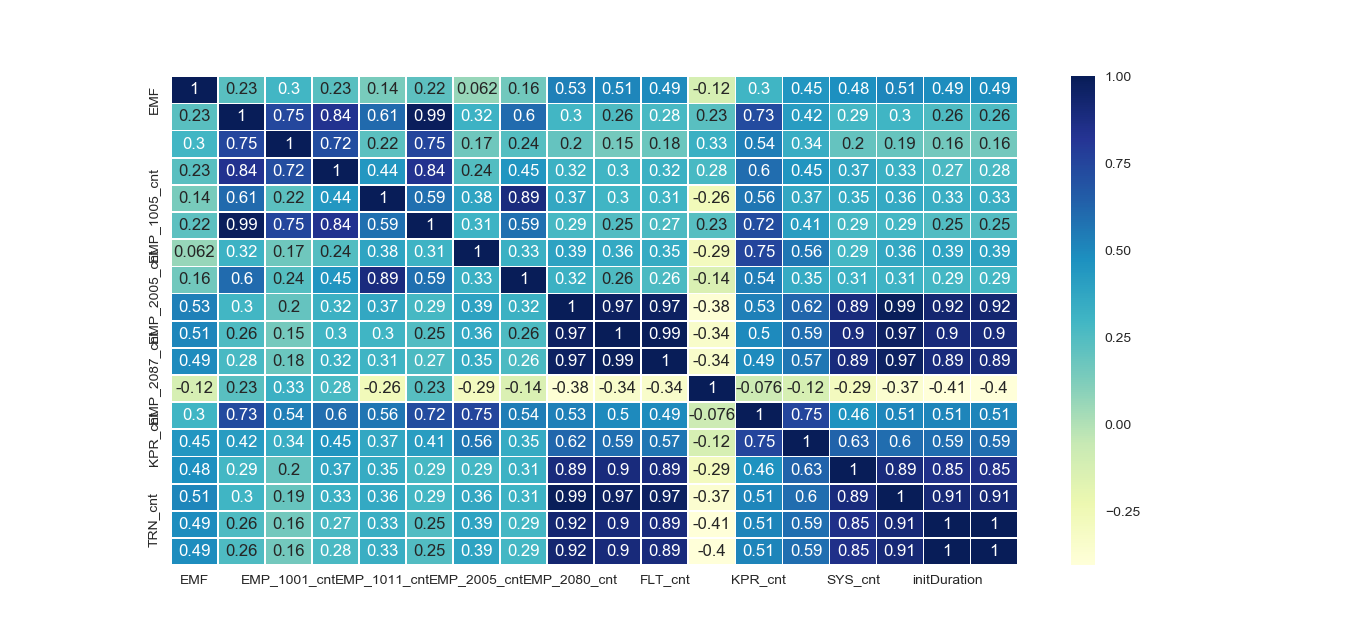
True 0.92 0.90 0.91 49

avg / total 0.92 0.92 0.92 115

* As we can see from the above metrics we achieved accuracy of 94% on training data and 92% on testing data.

**Logistic Regression Classifier**

* Below is the heatmap to learn about the correlation between dependant variable and independent variable as well as between two independent variables.
* One of the assumptions of logistic regression algorithm is there should not be any multicollinearity.
* Based on the data description the columns ‘Init\_st\_ts’,’LocoID’,’Ops\_Id’,’init\_end\_ts’ are of object type and hence dropping those colums from the dataframe and we have ‘initDuration’ column which provides information on time hence ‘init\_st\_ts’ and ‘init\_end\_ts’ can be dropped and not much information is lost and there’s no fear of underfitting.
* Based on the heatmap and correlation matrix we can find out that there is a strong correlation between EMP\_2000\_cnt and EMP\_1000\_cnt, EMP\_2080\_cnt and EMP\_2087\_cnt,FLT\_cnt with EMP\_2080\_cnt and EMP\_2087\_cnt ,TRN\_cnt with FLT\_cnt, EMP\_2080\_cnt and EMP\_2087\_cnt ,initDuration with FLT\_cnt, EMP\_2080\_cnt and EMP\_2087\_cnt ,Init\_time\_taken with FLT\_cnt, EMP\_2080\_cnt and EMP\_2087\_cnt.

Experimenting with dropping several combinations of independent variabales without losing much information for a model to learn.

* We have df1 dataframe with 17 independent variables dropping columns of object type and df2 dataframe with 11 independent variables after dropping columns to get rid of multicollinearity,and data in all the columns looks to be on the same scale and no scaling is needed.
* The logistic regression model trained on df1 gives training accuracy of 99.77% and test accuracy of 98.19%.
* The logistic regression model trained on df2 gives training accuracy of 100% and test accuracy of 99.07%.

**LSTM Algorithm**

* The dataset is also trained using LSTM algorithm.
* In the LSTM we have used LabelEncoder but OneHotEncoder can also be used,the code is commented.
* We have used ‘Adam’ optimizer,’ReLU’ activation function at lstm cells and ‘sigmoid’ activation function at output layer.
* Dropout rate of 0.1 is used to avoid overfitting.
* The LSTM on our dataset gives 98.91% accuracy at the end of 100 epochs during training and 95% accuracy on test dataset