Name: cognizanti1 Rank : 30 F1 Score: 0.50

I have tried 4 classification methods learned: Logistic Regression, SVM, Decision Tree and Random Forests.

**Feature Selection**:

1)I converted the data into Sparse binary matrix using Scipy csr\_matrix function and performed feature selection during model development. For Decision Tree Classifier and RandomForests Classifier, I used the Entropy Criterion inbuilt of Sklearn package using the parameter ‘criterion’. I followed the feature selection from Sklearn’s Select From Model to reduce the features for SVM and Logistic Regression Model.

2)The dataset was imbalanced with disproportionally large class labels of 1’s (722 ) than 0’s (78). So I used **Imbalanced-learn python** package to oversample the binary class label of 1 as It had only 78 samples with contrast to class label of 0( [('0', 722), ('1', 78)]) ) in the training set , I oversampled the minority class label 1 using SMOTE method and trained the datasets using 4 classification methods and performed cross validation. After resampling , I got the following count for the class labels :([('1', 293), ('0', 507)]). SMOTE method generates new samples by changing the values of attributes. It works by creating synthetic samples from the minor class instead of creating copies. The algorithm selects two or more similar instances (using a distance measure) and perturbing an instance one attribute at a time by a random amount within the difference to the neighboring instances.

3) I followed the following methods to select the model and ran the program and got SVM performing better than other models as number of features >> number of samples

* I constructed **Decision Tree** with the following parameters:
  + Entropy is the best criterion to select features for this dataset. There is significant increase in F1 Score when criterion is Entropy than when it was Gini.
    - * Criterion: Entropy, Class\_weight: Balanced and Splitter: Best
      * F1 Score : [0.87645688 0.87703016]
* I constructed **Support Vector Machine** (Linear SVM) with Following parameters and did feature selection using SelectFromModel in sklearn. With SVMs and logistic-regression, the parameter C controls the sparsity: the smaller C the fewer features selected.
  + SVM is useful especially when the number of features(components/columns ) >> number of samples in the training set. SVM performs better with more iterations and the result is different for each iteration. Dual should be “True” here since no. of columns >> no. of samples
    - * C=0.01, penalty='l2', dual=True
      * F1 Score : [0.96196868 0.95883777]
* I constructed **LogisticRegression** with the following parameters and did feature selection using SelectFromModel in sklearn. Class-weight param gives weight to class labels inversely proportional to their frequency. Here, C controls the no. of features, as C is less only fewer features are selected. It follows “one over the rest” (ovr) scheme and this can be used for Binary classification. RandomState can be set to None or zero .To obtain a deterministic behaviour during fitting, random\_state must be fixed
  + - * C=0.01, random\_state=0, class\_weight = "balanced", solver='liblinear', multi\_class='ovr'
      * F1 Score : [0.96196868 0.95883777]
* I constructed **RandomForests** with following parameters and did feature selection Gini in Sklearn by default.
  + - * criterion = "entropy",n\_estimators=10, max\_depth=10, min\_samples\_split=2, random\_state=0
      * F1 [0.92344498 0.92760181]

I chose Support Vector Machine as I found it gave good F1 score when using cross validation of Training Data set. Similar to SVC with parameter kernel=’linear’, but implemented in terms of liblinear rather than libsvm, so it has more flexibility in the choice of penalties and loss functions and should scale better to large numbers of samples and it gave me a F1 score of 0.50 at the time of submission. This class supports both dense and sparse input.

I did cross\_validation using the function test\_train\_split function() with test size of 0.3 to get F1 scores of Training Dataset after cross validation.