One of the questions posed was how adjusting the test and train set size affects the accuracy. Results are as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Train/Test ratio | Accuracy | Recall | F1-score | Support |
| 2/8 | 0.792 | 0.80 | 0.79 | 5635 |
| 4/6 | 0.801 | 0.80 | 0.80 | 4226 |
| 6/4 | 0.795 | 0.80 | 0.79 | 2818 |
| 8/2 | 0.855 | 0.81 | 0.80 | 1409 |

Table -Measurements for different train and test set sizes

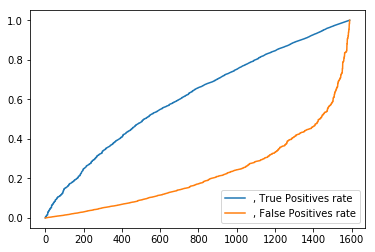
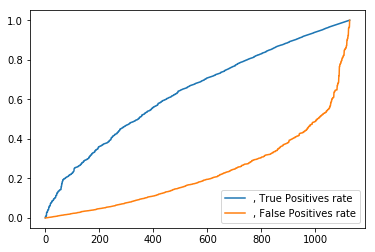


Figure -ROC for ratio 2/8

Figure -ROC for ratio 4/6

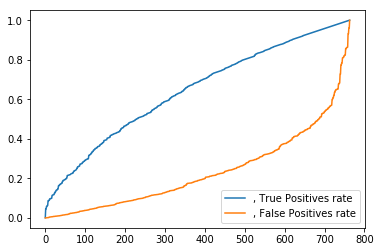
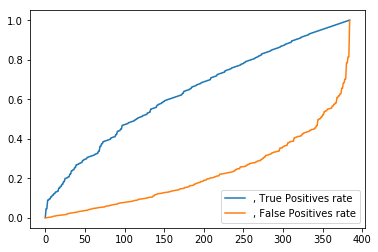


Figure --ROC for ratio 6/4

Figure --ROC for ratio 8/2

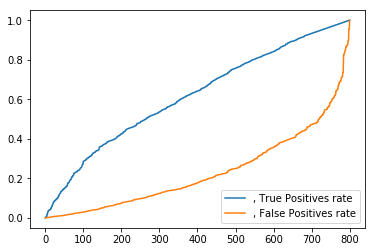
Initially the dataset was shuffled, otherwise it gave an accuracy of approximately 63% with logistic regression. After shuffling the dataset, 4 trials were conducted with different chunks of data for train and test sets. It is observed that although splitting of dataset was made in proportions significantly different from each other, the range of accuracy was between 79%-81%, with the lowest at 79% when train set contained 20% of the whole dataset. ROC curves for different cases do not differ substantially from each other, however the difference becomes clearer in the true and false positive rates in the last case (test set is 20% of dataset) where there is a minor distortion on the plots (for 50<x<100) signifying that the tpr and fpr did not increase proportionally.

Feature selection effect on the results:

Since there were 19 features in total, it is worth considering that not all of them have equal importance in predicting churn. Thus, using SelectKBest with chi score function and choosing to distinguish 5 most important features gave the following results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train/Test ratio | Accuracy | Recall | F1-score | Support | Accuracy difference from using all features |
| 4/6 | 0.786 | 0.79 | 0.78 | 4226 | -0.015 |
| 6/4 | 0.787 | 0.79 | 0.78 | 2818 | -0.008 |
| 8/2 | 0.795 | 0.79 | 0.79 | 1409 | -0.06 |

Table - Table for top 5 features



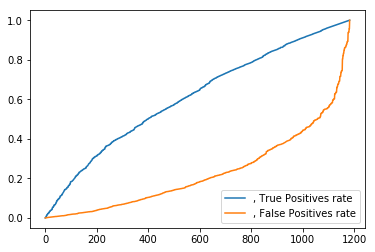


Figure -ROC for ratio 4/6

Figure -ROC for ratio 6/4

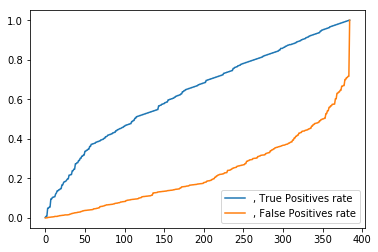


Figure -ROC for ratio 8/2

From the last column of the table it can be deduced that the top 5 features chosen yield results very similar to those recorded while using all 19 features. The top 5 features are respectively: tenure, online security, contract, monthly charge, total charge. Due to this difference the number of features was reduced further to 3 to observe how scores change:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train/Test ratio | Accuracy | Recall | F1-score | Support | Accuracy difference from using all features |
| 4/6 | 0.786 | 0.79 | 0.77 | 4226 | -0.015 |
| 6/4 | 0.785 | 0.79 | 0.77 | 2818 | -0.01 |
| 8/2 | 0.790 | 0.79 | 0.78 | 1409 | -0.065 |

The 3 dominant features in this stage were tenure, monthly charge and total charge. From the obtained results it can be observed that the difference between using 5 and 3 top features is relatively small and seems that online security and contract features do not have fundamental contribution to the results as the accuracy does not vary significantly after discarding those features.

Doing cross validation:

Using LogisticRegressionCV with different number of fold values affected the model and accuracy. Initially cv = 4 was tried with different train/dataset size ratios and the following results were obtained:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Train/Test ratio | Accuracy | Recall | F1-score | Support | Accuracy difference w/o cross validation |
| 2/8 | 0.736 | 0.34 | 0.62 | 5635 | -0.056 |
| 4/6 | 0.801 | 0.80 | 0.80 | 4226 | 0 |
| 6/4 | 0.793 | 0.79 | 0.79 | 2818 | -0.002 |
| 8/2 | 0.801 | 0.80 | 0.79 | 1409 | 0.054 |

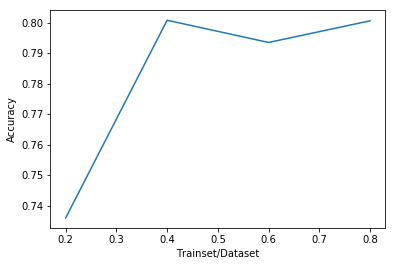
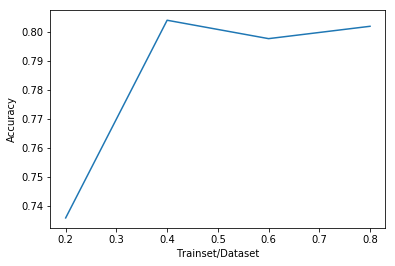


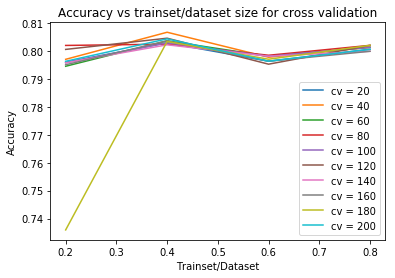
Figure 8-Accuracy results for CV =4

While separating the trainset into 10 folds, no significant difference in results were obtained. The following plot contains the accuracies obtained from the same process as that conducted formerly, but with parameter cv = 10:

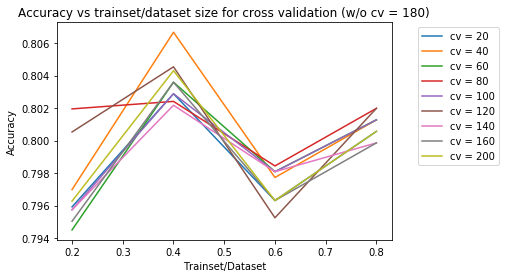


However it was observed that, as the number of folds increased, the better was the accuracy obtained. This can be attributed to the fact that, if the number of folds is small then each of them will have a big chunk of data, thus leading to a smaller amount of data for training and a big one for validation. For instance in the case of having 4 folds: 5634/4 = 1408 rows would belong to the validation fold on each iteration and there would be less data for training. However, as the number of folds is increased, say 100, on which the accuracy is significantly higher (see table below), the number of rows in the validation fold is much lower 5634/100 = 56 and there are better chances to obtain a better performing model.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ratio  Nr Folds | Train/Test = 2/8 | Train/Test = 4/6 | Train/Test = 6/4 | Train/Test = 8/2 |
| 20 | 0.795918 | 0.802887 | 0.796309 | 0.800568 |
| 40 | 0.796983 | 0.806673 | 0.797729 | 0.801278 |
| 60 | 0.794499 | 0.803597 | 0.798084 | 0.801278 |
| 80 | 0.801952 | 0.802414 | 0.798439 | 0.801987 |
| 100 | 0.795741 | 0.802887 | 0.798084 | 0.801278 |
| 120 | 0.800532 | 0.804543 | 0.795245 | 0.801987 |
| 140 | 0.795741 | 0.802177 | 0.798084 | 0.799858 |
| 160 | 0.795031 | 0.803597 | 0.796309 | 0.799858 |
| 180 | 0.735936 | 0.80336 | 0.797019 | 0.801987 |
| 200 | 0.796273 | 0.804307 | 0.796309 | 0.800568 |



From the data and the plot it can be observed that the best accuracy is obtained when the trainset contains about 40% of the whole dataset and the nr of folds = 40. It’s observed that increasing the size of trainset beyond 40% does not yield better accuracy, as it did in the very first case with no cross validation. Considering the above discussion about the reasons why increasing the number of folds increases the accuracy and the recent results, we can deduce that the former reasoning does not always hold. For instance, the performance does increase as cv increases gradually for most of the cases, however when cv = 180 a strange result is observed where the accuracy is approximately 0.73 which is similar to the case above when cv = 4. Furthermore, the best performance does not correspond to the case with the largest number of folds. If we discard the penultimate case where cv = 180 and re-plot the graph to distinguish the accuracies relationship of those cases which start above 0.79, then we get the following:



The results become clearer in this graph and support the above claim that increasing the nr of folds improves the performance generally, with some exceptions. However, considering the accuracy presented in the y axis, it can be observed that the change between all cases occurs in the second decimal digit, thus meaning around 1%. This is not an essential change in performance as to completely dispute the argument made related to the number of folds. It should also be considered that there might be biased portion of data while training or validating which also affect the parameters and lead to over or under fitting.