Telco Customer Churn Prediction
A comparison study of Logistic Regression and Random Forest

Supplementary Materials

Elnara Mammadova
Department of Computer Science
INM 431 – Machine Learning Coursework

Glossary

Accuracy - The fraction of predictions that a classification model got right. In a binary classification accuracy has the following definition:

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Number Of Examples}}$$

Binary classification - A type of classification task that outputs one of two mutually exclusive classes.

Class - One of a set of enumerated target values for a label.

Classification Model - A type of machine learning model for distinguishing among two or more discrete classes.

Confusion Matrix - An (N x N) table that summarizes how successful a classification model's predictions were; that is, the correlation between the label and the model's classification. One axis of a confusion matrix is the label that the model predicted, and the other axis is the actual label. N represents the number of classes.

Continuous Feature - A floating-point feature with an infinite range of possible values.

Cross-validation - A mechanism for estimating how well a model will generalize to new data by testing the model against one or more non-overlapping data subsets withheld from the training set.

Dataset - A collection of examples

Discrete Feature - A feature with a finite set of possible values. For example, a feature whose values may only be *animal*, *vegetable*, or *mineral* is a discrete (or categorical) feature.

Ensemble - A merger of the predictions of multiple models.

Hyperparameter - The "knobs" that you tweak during successive runs of training a model.

Label - In supervised learning, the "answer" or "result" portion of an example. Each example in a labelled dataset consists of one or more features and a label.

Logistic Regression - A classification model that uses a sigmoid function to convert a linear model's raw prediction (y′) into a value between 0 and 1.

Loss - A measure of how far a model's predictions is from its label. Or, to phrase it more pessimistically, a measure of how bad the model is.

Loss curve - A graph of loss as a function of training iterations.

Machine Learning - A program or system that builds (trains) a predictive model from input data. The system uses the learned model to make useful predictions from new (neverbefore-seen) data drawn from the same distribution as the one used to train the model. Machine learning also refers to the field of study concerned with these programs or systems.

Model - The representation of what a machine learning system has learned from the training data

Overfitting - Creating a model that matches the training data so closely that the model fails to make correct predictions on new data.

Precision - A metric for classification models. Precision identifies the frequency with which a model was correct when predicting the positive class. That is:

$$\text{Accuracy} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positive}}$$

Precision Recall Curve - A curve of precision vs. recall at different classification thresholds.

Prediction - A model's output when provided with an input example.

Random Forest - An ensemble approach to finding the decision tree that best fits the training data by creating many decision trees and then determining the "average" one. The "random" part of the term refers to building each of the decision trees from a random selection of features; the "forest" refers to the set of decision trees.

Recall - A metric for classification model that answers the following question: Out of all the possible positive labels, how many did the model correctly identify? That is:

$$\text{Accuracy} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negative}}$$

Test set - The subset of the dataset that you use to test your model after the model has gone through initial vetting by the validation set.

Training - The process of determining the ideal parameters comprising a model.

Training set - The subset of the dataset used to train a model.

Validation - A process used, as part of training to evaluate the quality of a machine learning model using the validation set. Because the validation set is disjoint from the training set, validation helps ensure that the model’s performance generalizes beyond the training set

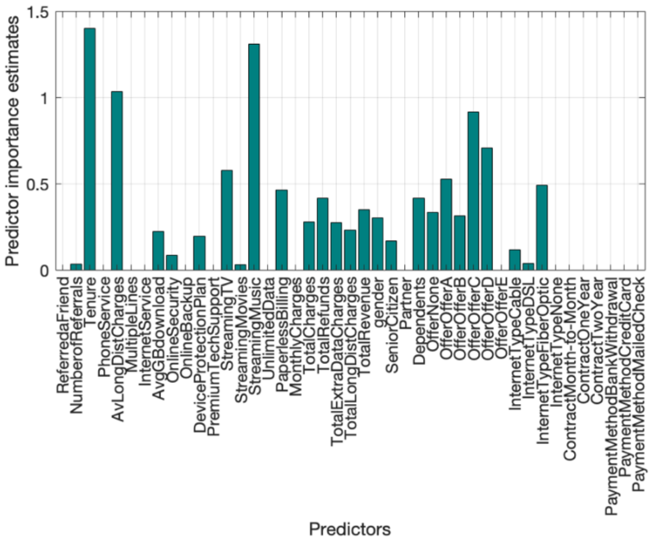
Definitions taken from Google Developer’s Machine Learning Glossary [1]

Intermediate Results

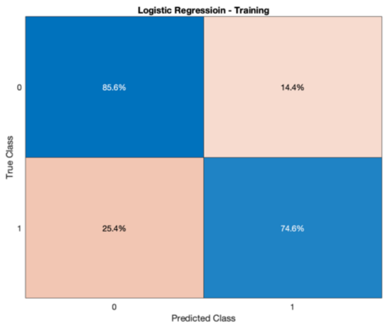
Logistic Regression

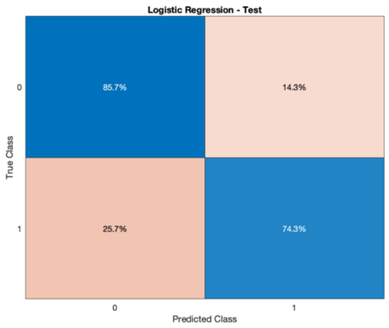
- Initial model was constructed using cross-validated lasso regularization of a generalized linear model using 25 Lambda and 10-fold cross-validation on balanced training data.

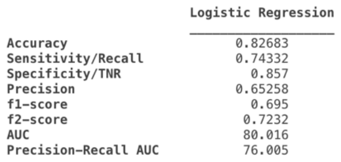
- 27 nonzero predictors were identified



- Trained the model using the minimum coefficients acquired and Confusion matrix were constructed for Train and Test set. As seen in the figure below, the because Logistic regression was trained on biased data, when it encounters considerable negative class in the test set, it ends up predict a greater proportion of false negatives and a lower proportion of false positive instances, resulting in lower accuracies.

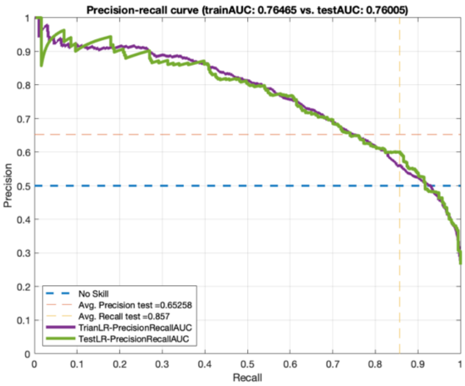






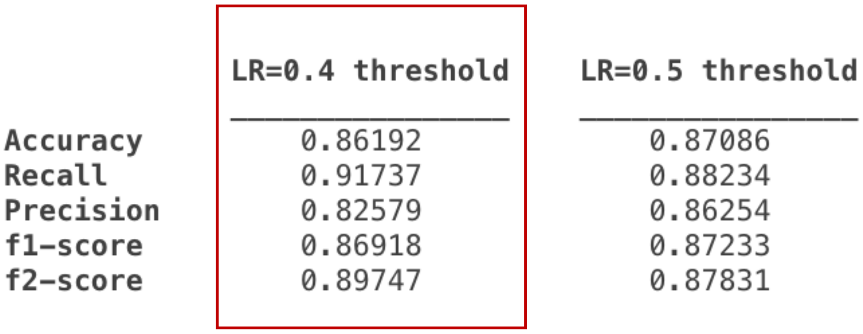
```
                          Logistic Regression
                          ___________________
Accuracy                        0.82683
Sensitivity/Recall              0.74332
Specificity/TNR                   0.857
Precision                       0.65258
f1-score                          0.695
f2-score                         0.7232
AUC                              80.016
Precision-Recall AUC             76.005
```

- Precision Recall AUC curves shown below shows **the tradeoff between precision and recall for different threshold**. A low area under the curve represents both low recall and low precision, where low precision relates to a high false positive rate, and low recall relates to a high false negative rate.



- It was concluded that Logistic regression did not perform well with biased dataset. So next, model was run on balanced dataset (unbiased), resulting in much better results presented in the poster.

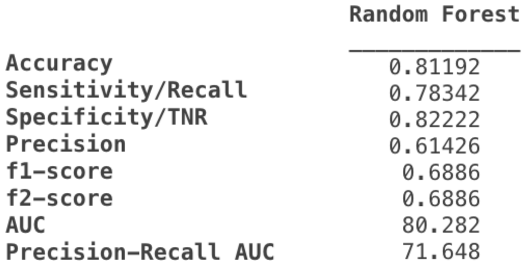
- Another additional tuning was done by applying the threshold to the cut-off point in probability between the positive and negative classes, which by default for any classifier would be set at 0.5. However, setting the threshold to >=0.4, biased the predictive behaviour of a classification model. That way the model produced imbalanced binary prediction that is not biased to the majority class. *Table 1* shows the comparison of performance measures for Logistics regression model with thresholds set to >=0.5 vs. >=0.4. As seen in the table below, I decided to go with 0.4 threshold as it presented much better performance measures.



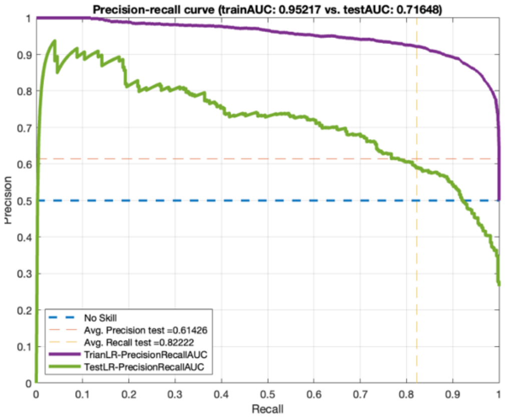
```
               LR=0.4 threshold    LR=0.5 threshold
               ________________    ________________
Accuracy           0.86192             0.87086
Recall             0.91737             0.88234
Precision          0.82579             0.86254
f1-score           0.86918             0.87233
f2-score           0.89747             0.87831
```

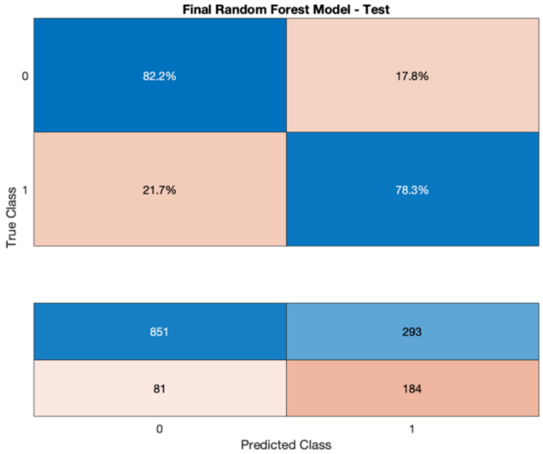
Random Forest

- After observing good results from Logistic Regression using unbiased data, I decided to train Random Forest on balanced training data from the get-go, without using any prior or custom misclassification cost matrix. We ran home-mode grid search for hyperparameter tuning using balanced training data, and 10-fold cross validation (maxNumSplit-200, MinLeafSize-5). We acquired the optimal tree size of 181 through a cross validation. However, I achieved poor performance from this experiment compared to the final model presented in the poster. Refer to the confusion matrix and AUC curves below, as well the loss curves for cross-validation vs. test set.



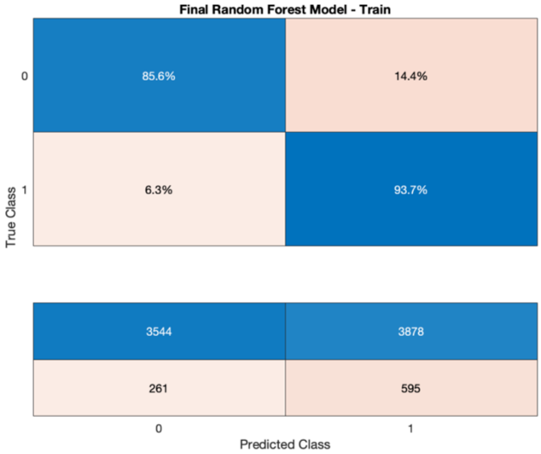
```
                          Random Forest
                          _____________
Accuracy                     0.81192
Sensitivity/Recall           0.78342
Specificity/TNR              0.82222
Precision                    0.61426
f1-score                      0.6886
f2-score                      0.6886
AUC                           80.282
Precision-Recall AUC          71.648
```





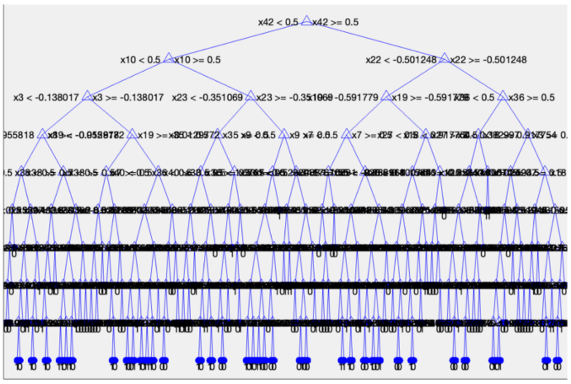
Final Random Forest Model - Test

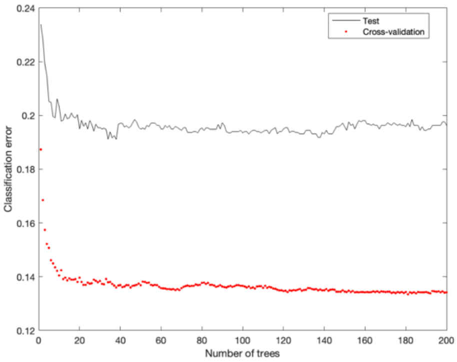
True Class	Predicted Class 0	Predicted Class 1
0	82.2%	17.8%
1	21.7%	78.3%
	851	293
	81	184



Final Random Forest Model - Train

True Class	Predicted Class 0	Predicted Class 1
0	85.6%	14.4%
1	6.3%	93.7%
	3544	3878
	261	595





Implementation Details

Logistic Regression and Random Forest were chosen for this study. The dataset contains continuous and discrete features with binary responses. From these properties it becomes apparent that this is a classification problem and that regression model do not apply. Logistic Regression was especially applied as it supports binary classification. Implementation was done using single live script for ease of communication and understanding of the process steps. Home-made grid search was run on a different live script for ease, since each grid search (total of 14 were run) took between 1-3 hrs.

For *Logistic Regression* Lasso (L1) regularization was used and removed redundant predictors by using cross-validated fits. During training coefficients from the minimum-plus-one standard error point were used. Applied the threshold to the cut-off point in probability between the positive and negative classes to 0.4. Confusion matrix, ranking metrics and ROC-AUC and PR-AUC were generated for comparison with the subsequent model.

For *Random Forest*, we chose to use unbalanced dataset as it achieved much better accuracies. Positive classes were quite a low percentage of the data. This imbalance indicates that RUSBoost is an appropriate algorithm to use since the data is skewed, but it is not within the scope of this course therefore Bag (Random Forests) was used.
Initial model was trained on bagged ensemble of regression trees using hyperparameters acquired from 10-fold cross validated home-made grid search on a different script. Grid Search results were plotted in 2D and 3D using different metrics, to find the optimal hyperparameter thresholds (MaxNumSplit – 150, MinLeafSize-4). Priors [26.5, 73.5] and misclassification cost matrix [0 3; 1 0] were used (using several tries to acquire the optimal numbers for cost matrix). From that model the NumLearningCycle was chosen as a subsequent hyperparameter to be optimized. The loss and error of the model at varying learning cycles was determined, and the number of trees yielding the lowest associated error was chosen (Optimal number of trees = 68). The model was validated using k-fold cross validation. Finally, the final classifier was trained with all the optimized hyperparameters and later implemented on the test set.

References:

[1] Machine Learning Glossary, Google Developers.
https://developers.google.com/machine-learning/glossary