CS 464 Project

Telecom Customer Churn Rate Prediction

Group 5

Alba Mustafaj	21500009
Alp Ege Baştürk	21501267
Berat Biçer	21503050
Bora Ecer	21501757
H. Buğra Aydın	21501555

Introduction & Project Description

- The objective of this project is predicting user's churn rate, which occurs when a customer stops using the service.
- Predicting whether a customer is leaving or not at the end of the contract term is done by looking at the customer's data according to the features determined at the time of training.
- Prediction accuracy is the main problem

Person in doubt [1]

Dataset Description

- The dataset to be used is that of Telco's Customer Data [2]
- The raw data contains 7043 rows (customers) and 21 columns (features).

customerID	**gender**	**SeniorCitizen**	**Partner**	**Dependents**
Customer ID	Customer gender (female, male)	Whether the customer is a senior citizen or not (1, 0)	Whether the customer has a partner or not (Yes, No)	Whether the customer has dependents or not (Yes, No)

Features

- customerID
- Gender,
- SeniorCitizen
- Partner,
- Dependents
- Tenure
- PhoneService
- MultipleLines
- InternetService
- OnlineSecurity
- OnlineBackup,
- DeviceProtection
- TechSupport
- StreamingTV
- StreamingMovies
- Contract
- PaperlessBilling
- PaymentMethod
- MonthlyCharges
- TotalCharges
- **Churn**

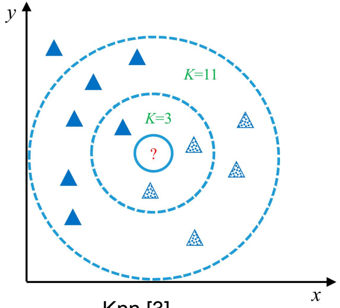
Data Preprocessing

- Initial dataset included non numeric fields like gender and true/false fields, missing data and hashed user ID

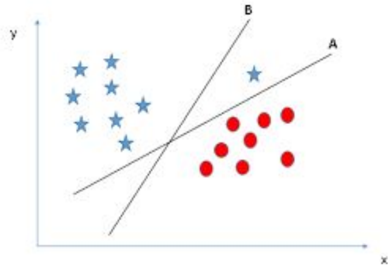
Methods

Following methods were used:

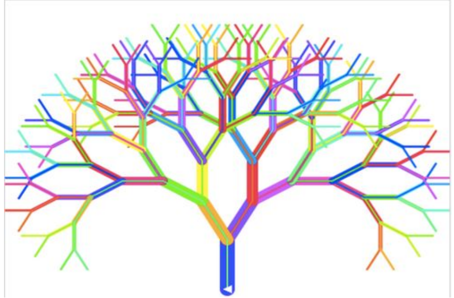
- SVM
- kNN
- logistic regression
- decision tree
- random forest
- Neural Networks



Knn [3]



SVM [4]



Random forest [5]

Success of different methods will be compared to find optimal one(s).

Train/Test Split & Cross Validation

- Used Train/Test split for generalizable models
- Applied cross validation for:
  - SVM
  - Logistic Regression
  - kNN
  - Decision Tree
  - Random Forest

Feature & Parameter Selection

- Used to find optimal parameters and features
- Feature selection was used to improve time
- Parameter selection was used to improve accuracy

Results

- All experiments aimed to find optimal accuracy using that model.
- The experiments we have followed were to answer the following questions:
  - Which are the best hyper parameter values that results with the highest accuracy?
  - Which are the best features that results with the highest accuracy once used with the best hyper parameters?
  - Which train/test split is better in terms of accuracy?
  - What is the relationship between the selected feature amount and the accuracy?
  - What are the ROC, Precision-Recall results?

SVM

- Parameters:
  - Kernel, C, gamma
- C parameter gives regularization. For large C it acts like hard margin.
- Gamma determines influence of further data points on the calculation.
- Recursive feature elimination was used in order to get first i best features.
- Hyperparameter selection was applied in order to choose kernel type, C and gamma parameters.

SVM Results

- Test set ratio = 40%
- Choosing larger or smaller test sets didn’t have significant effects, thus no graphs were drawn since results were inseparable from noise.
- In the case where test set was chosen as 5%, accuracy fluctuated between 78-83%

SVM Results

Accuracy: 80.207%

Confusion Matrix:

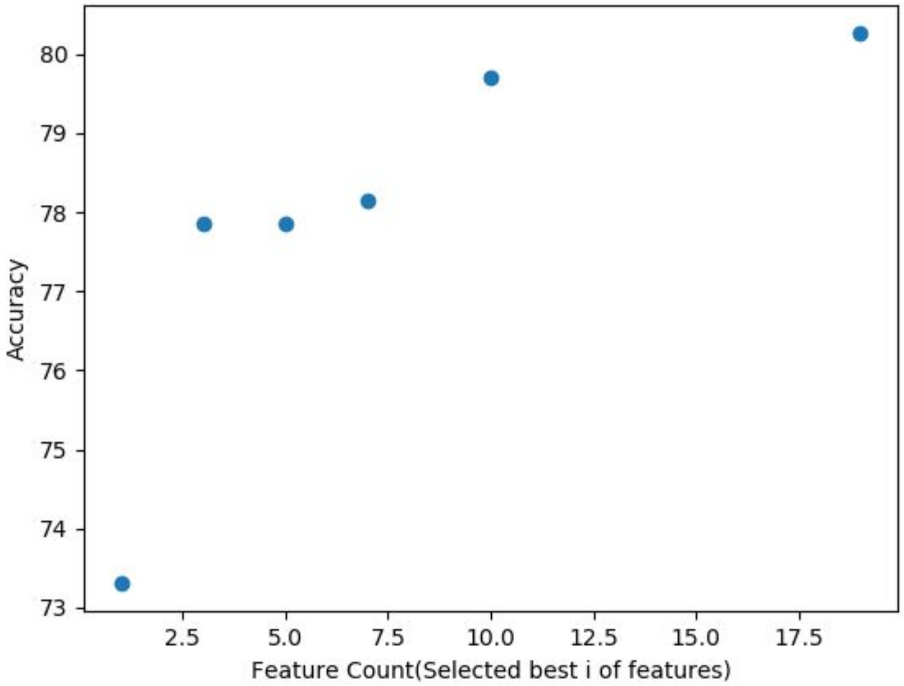
Actual \ Predicted	0	1
0	1906	173
1	386	353

	precision	recall	f1-score	support
0	0.83	0.92	0.87	2079
1	0.67	0.48	0.56	739
micro_avg	0.8	0.8	0.8	2818
macro_avg	0.75	0.7	0.72	2818
weighted_avg	0.79	0.8	0.79	2818

SVM Results

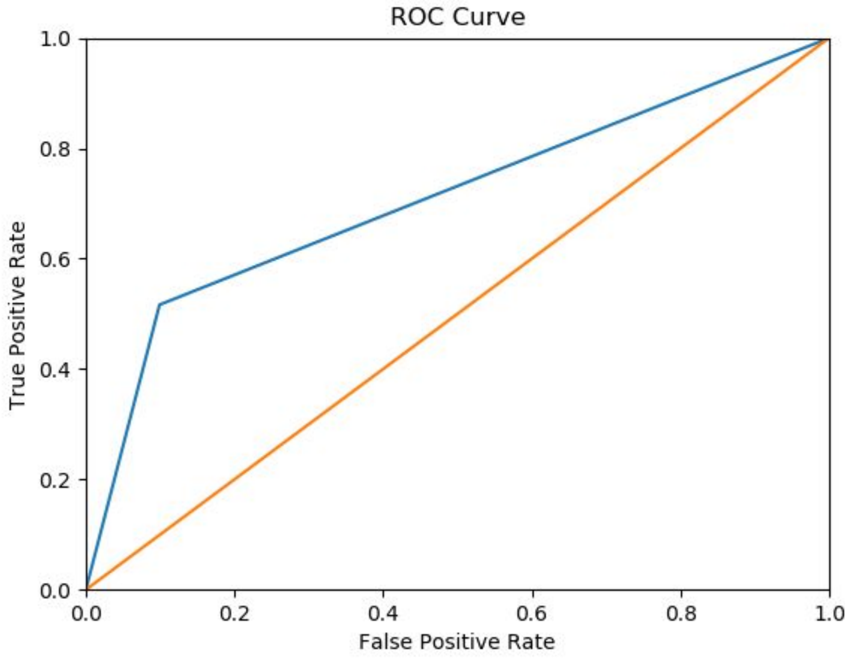
- Best Features:
  - Best feature: tenure
  - Best 3 features: tenure, InternetService, Contract
  - Best 5 features: tenure, InternetService, OnlineSecurity, Contract, TotalCharges
  - Best 7 features: tenure, InternetService, OnlineSecurity, TechSupport, StreamingMovies, Contract, TotalCharges
  - Best 10 features: tenure, InternetService, OnlineSecurity, TechSupport, StreamingTV StreamingMovies, Contract, PaperlessBilling, MonthlyCharges, TotalCharges
- Best Parameters:
  - kernel: rbf
  - C: 100
  - gamma: 0.001

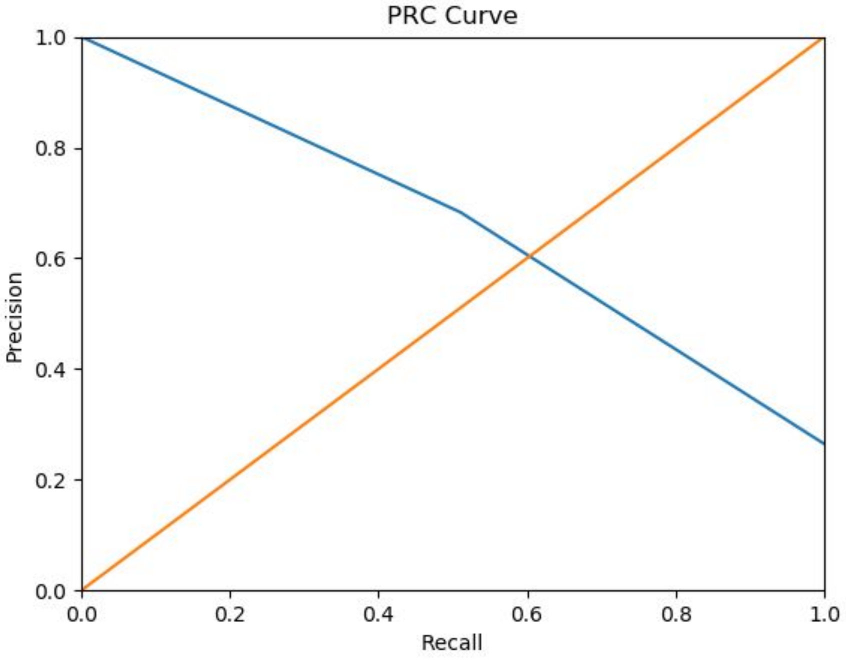
SVM Results



80
79
78
77
76
75
74
73
Accuracy
2.5
5.0
7.5
10.0
12.5
15.0
17.5
Feature Count(Selected best i of features)

SVM Results


ROC Curve
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate


PRC Curve
1.0
0.8
0.6
0.4
0.2
0.0
Precision
0.0
0.2
0.4
0.6
0.8
1.0
Recall

SVM Discussion

- Feature selection was not helpful for the accuracy
- All 19 features were used in order to calculate curves and accuracy since it was not expensive.
- Hyperparameter selection gave rbf kernel as the best choice, which was expected from our dataset.
- SVM is a good model considering it’s simple enough to train and predict and also it provides sufficient accuracy to solve the problem. If necessary, first three features might be used in order to improve performance while sacrificing negligible accuracy.

Neural Network

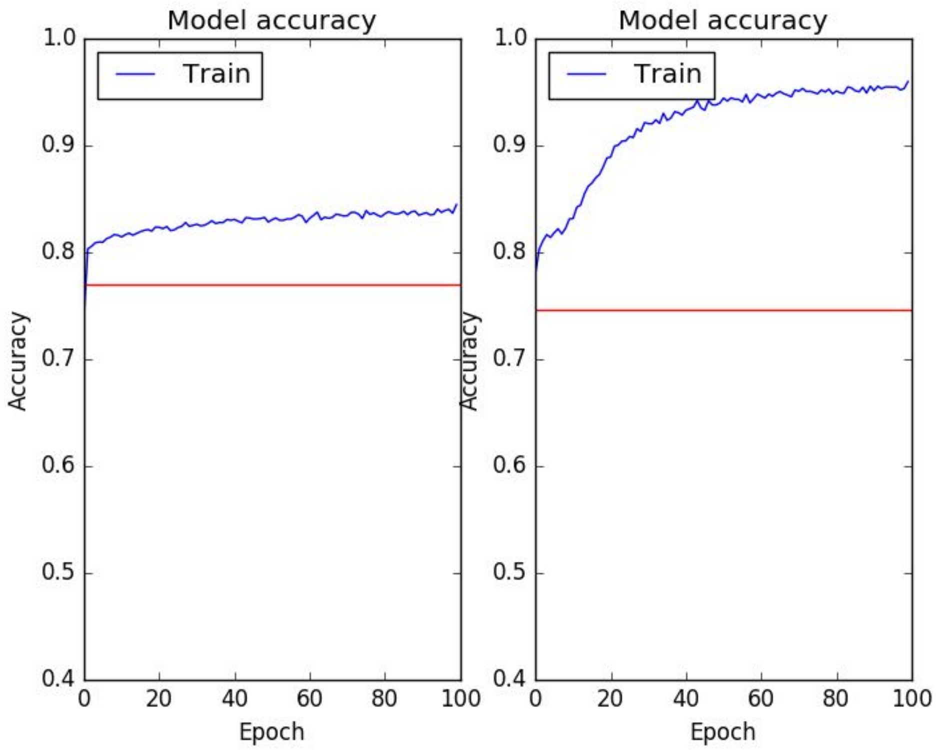
- 2 NNs were used, one simple and one complex one.
- Feature selection was applied
- No parameter selection was applied

Neural Network Results

- Test set ratio = 40%
- Simple one with three layers with 20, 5 and 1 nodes.
- More complex one with 6 layers of 200, 150, 100, 50, 25, 1 nodes.
- Purpose of this experiment is finding a network which is fast and does not overfit the data.

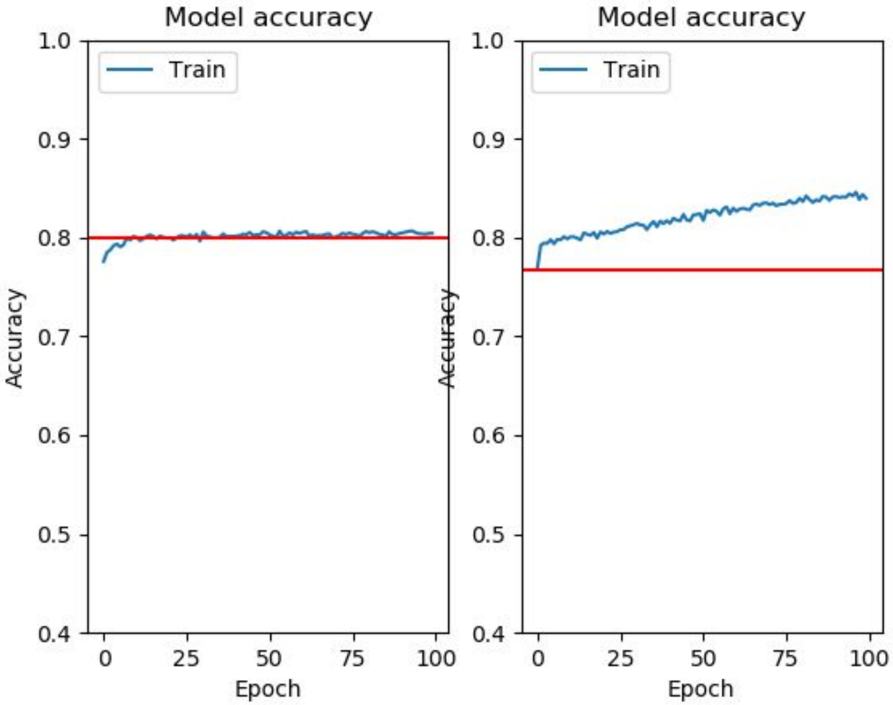
Neural Neural Network Results

- 19 Features



Neural Network Results

- Best 10 Features



Neural Network Discussion

- Both models overfit without feature selection
- Only complex network overfits when using best 10 features
- Test set accuracy is capped around 80%, which is also reached by other methods.
- Thus NN is not optimal because of training time

Decision Tree

Parameters:

- Impurity criterion: gini or information gain using entropy
- Maximum depth of the decision tree
- Minimum number of samples required to be at a leaf node
- Minimum number of samples required to split an internal node
- Splitter strategy: best or random

Decision Tree

- Hyper parameter selection with 5-fold cross validation applied to different combinations of the parameter values.
- Then feature selection for different test/train ratios were conducted using the best hyper parameter values to find best accuracy.

Decision Tree Results

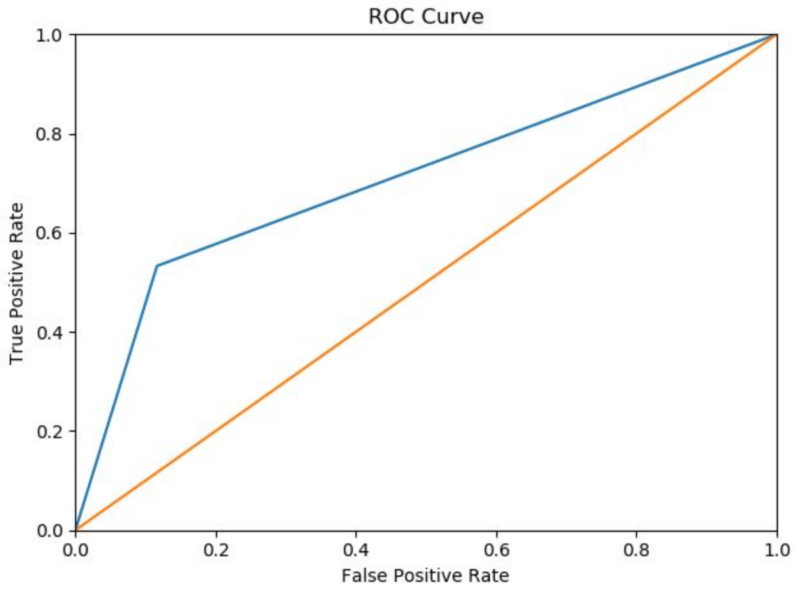
- Best Parameters
  - Maximum Depth: None
  - Min. number of samples required to be at a leaf node: 40
  - Min. number of samples required to split an internal node: 68
  - Splitter Strategy: Best
- Best feature set = Contract, StreamingMovies, TechSupport, InternetService, MultipleLines, Dependents.
- Best train/test ratio: 6/4
- Best overall accuracy: % 79

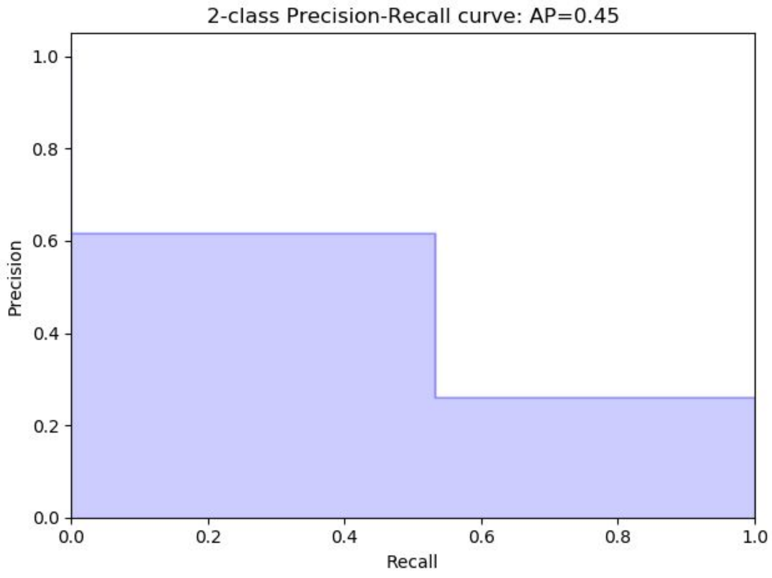
Decision Tree: Results

- Confusion Matrix:

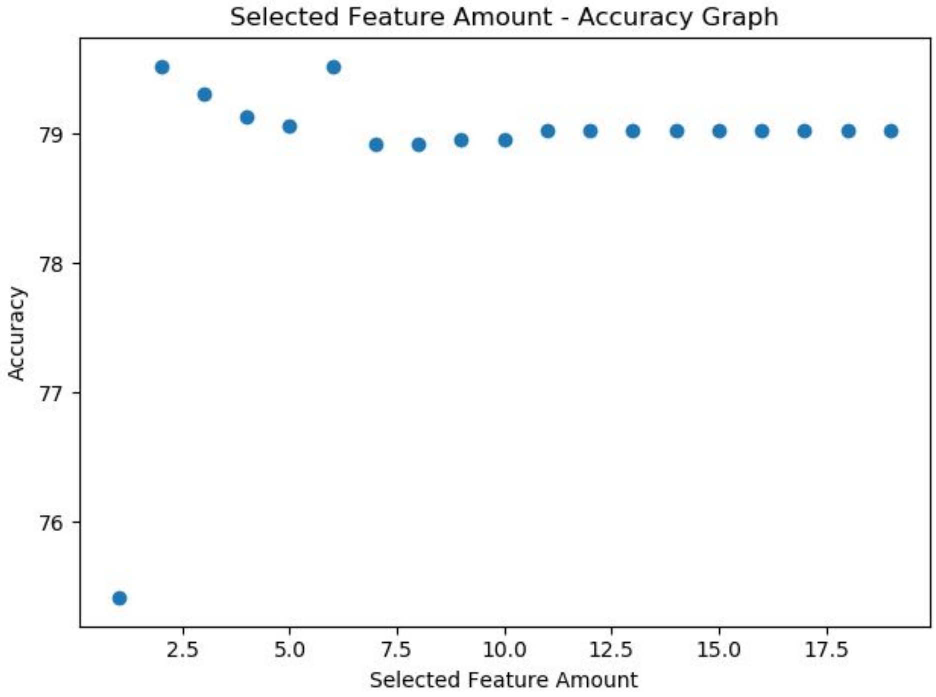
Actual \ Predicted	0	1
0	1903	165
1	419	331

Decision Tree: Results





Decision Tree: Results



Decision Tree: Discussion

- Hyper parameter selection was helpful.
- However the accuracies with the best parameters and with the default values was close to each other.
- Feature Selection did slightly improved the accuracy.
- In terms of overall performance, Decision Tree model is stable. It performs similarly in most of the cases.

Logistic Regression

Parameters:

- penalty
- Tol
- Dual
- fit_intercept
- solver
- multi_class

Logistic Regression

Parameters:

multi_class : ‘ovr’ for binary classification

Solver : ‘Liblinear’ as it is more suitable for the size of our dataset

Dual : False, as the number of samples > number of features

fit_intercept, tolerance and penalty tuned experimentally

Logistic Regression

Penalty ‘l1’ vs ‘l2’:

Slight accuracy improvement of 1% and tpr over fpr for penalty

Tolerance set to 1e-4 was better with:

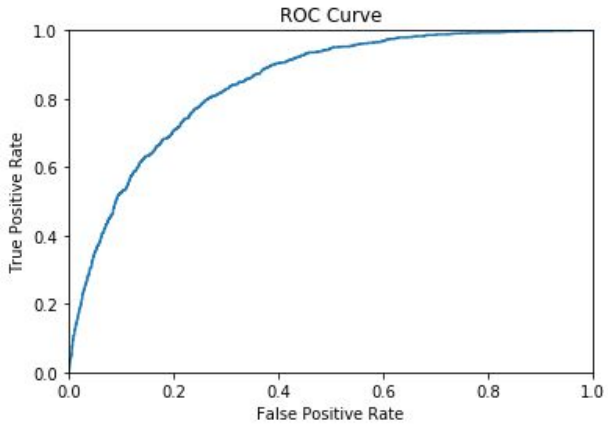
Accuracy: 0.804

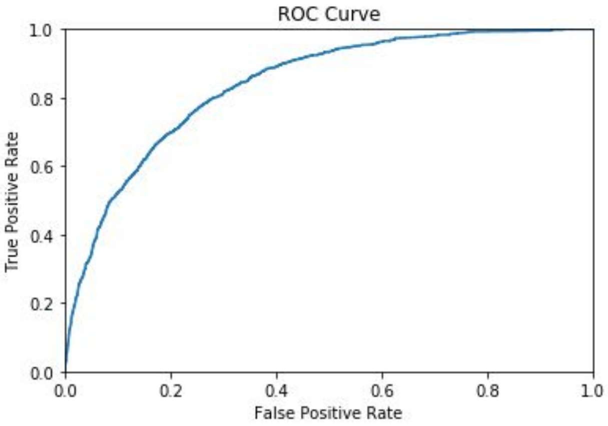
Precision: 0.79

Recall: 0.80

fit_intercept False vs “True”

Adding constant improved accuracy by 9.2% as well as tpr ove





Logistic Regression

Train/Test split:

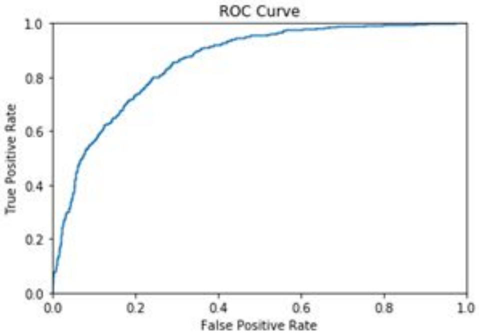
Train/Test ratio	Accuracy	Recall	F1-score	Support
2/8	0.792	0.80	0.79	5635
4/6	0.801	0.80	0.80	4226
6/4	0.795	0.80	0.79	2818
8/2	0.855	0.81	0.80	1409

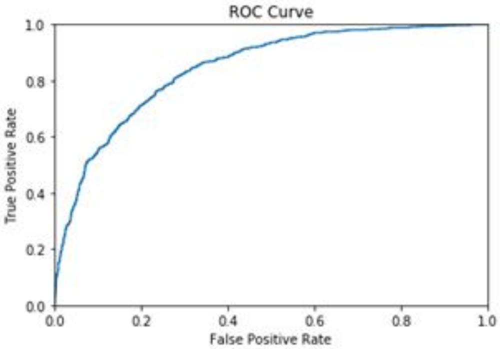
Accuracy range 79%-81%

79% :traineset 20% of the whole dataset.

81% traineset 80% of the whole dataset.

ROC curves for different cases do not differ substantially from each other (left 4/6, right 2/8)





Logistic Regression

Feature Selection:

SelectKBest with chi score function and 5 top features:

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

Top Features:***tenure, online security, contract, monthly charge, total charge.***

Reducing to 3 gave almost identical results and ***tenure, monthly charge, total charge*** as top features.

Logistic regression

Cross validation:

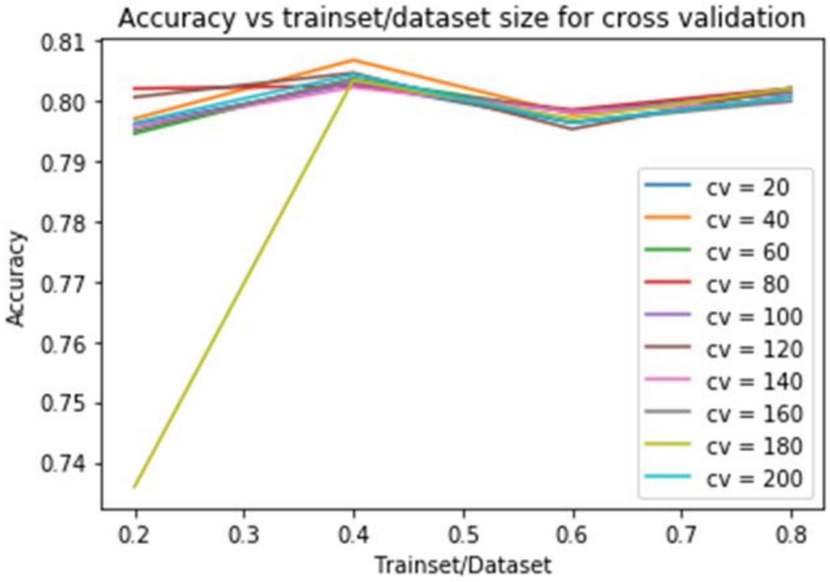
Trials for seven train/test sizes and number of folds were conducted.

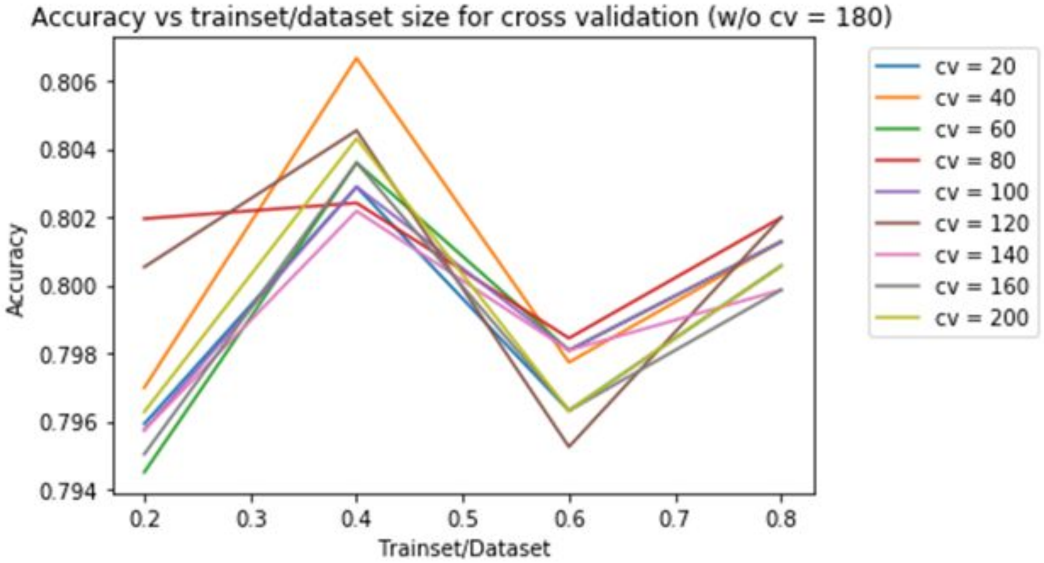
Having 4 number of folds yielded worse results than not using CV.

Best results if the trainset contains about 40% of the whole dataset and the nr of folds = 40.

Icreasing trainset beyond 40% does not yield better accuracy, as it did in the very first case with no cross validation.

Logistic Regression





Random Forest Classification

Parameters:

- Decision Tree Amount
- Maximum Depth
- Minimum Samples to Form A Leaf
- Maximum Leaf Nodes

Random Forest Classification

- Hyper parameter selection with 5-fold cross validation applied to different combinations of the parameter values.
- Then feature selection for different test/train ratios were conducted using the best hyper parameter values to find best accuracy.

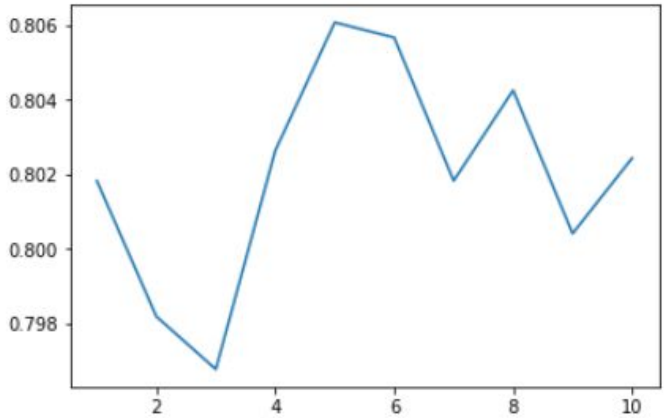
Random Forest Results

- Best Parameters
  - **Number of Decision Trees:** [1, 5, 10, 100]
  - **Maximum Leaf Nodes:** [10, 100, 1000, None]
  - **Max Depth:** [1, 10, 100, None]
  - **Min Samples Required In Leaf:** [1, 2, 5, 10]
- Best feature set = Tenure, Contract, Monthly Charges, Online Security, Internet Service
- Best test/train ratio: 0.3
- Best overall accuracy: 0.7955

Random Forest Results

- Confusion Matrix:

```
[[1418  136]
 [ 267  292]]
```

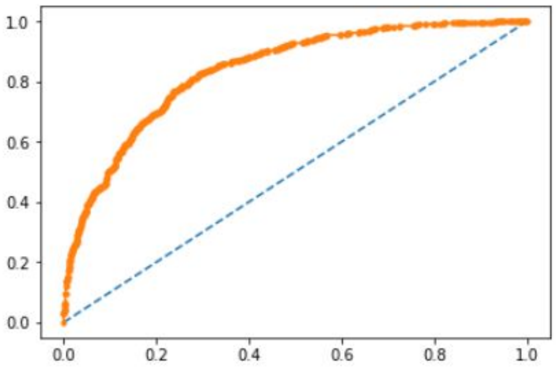


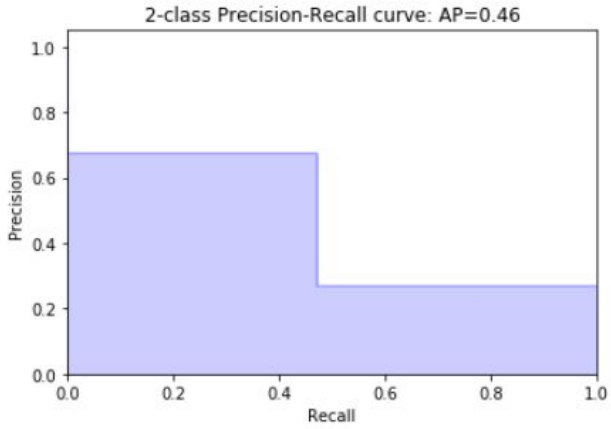
The highest accuracy is obtained with 5 of the most important features.

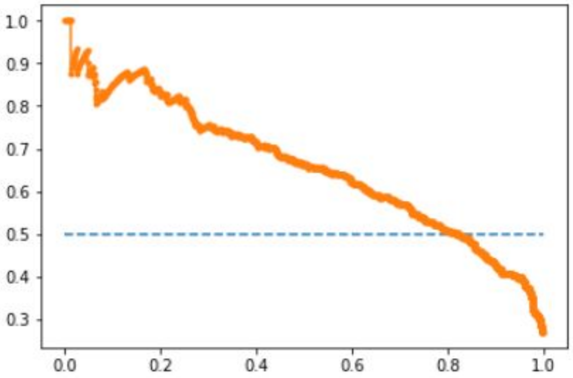
```
[[-14.566561809785105, 'tenure'],
 [-8.34395288133322, 'Contract'],
 [-3.959842045378472, 'OnlineSecurity'],
 [-3.959842045378472, 'TechSupport'],
 [-3.535573254802211, 'MonthlyCharges'],
 [-2.2627668830734144, 'OnlineBackup'],
 [-1.979921022689236, 'InternetService'],
 [-1.4142293019208791, 'SeniorCitizen'],
 [-0.989960511344618, 'Dependents'],
 [-0.8485375811525222, 'DeviceProtection'],
 [-0.8485375811525222, 'MultipleLines'],
 [-0.8485375811525222, 'StreamingMovies'],
 [-0.7071146509604396, 'StreamingTV'],
 [-0.5656917207683436, 'Partner'],
 [-0.4242687905762611, 'PhoneService'],
 [-0.0, 'PaperlessBilling'],
 [-0.0, 'PaymentMethod'],
 [-0.0, 'TotalCharges'],
 [0.1414229301920959, 'gender']]
```

Random Forest Results

AUC: 0.839







Random Forest Discussion

- Tenure is the most important feature
- Feature selection reduced complexity
- It is hard to make comments on the hyper parameters since different decision trees generated in each iteration
- Unstable approach with good results
- If we don’t limit the maximum number of leaf nodes, the model overfits

kNN

Parameters

- Number of closest neighbors
- Weights assigned to samples
- Selection of algorithm for computing nearest neighbors
- Distance metric for any 2 points

kNN

Hyperparameter selection with experiments for K = [1, 25]

Feature Count selection with experiments for feature count = [1, 19]

kNN Results

Best Parameters

- Number of closest neighbors K = 4
- Uniform weights assigned to samples (default case)
- Automatic selection of algorithm for computing nearest neighbors (default case)
- Distance metric Minkowski with p = 2, equivalent to Euclidean L2 distance (distance)

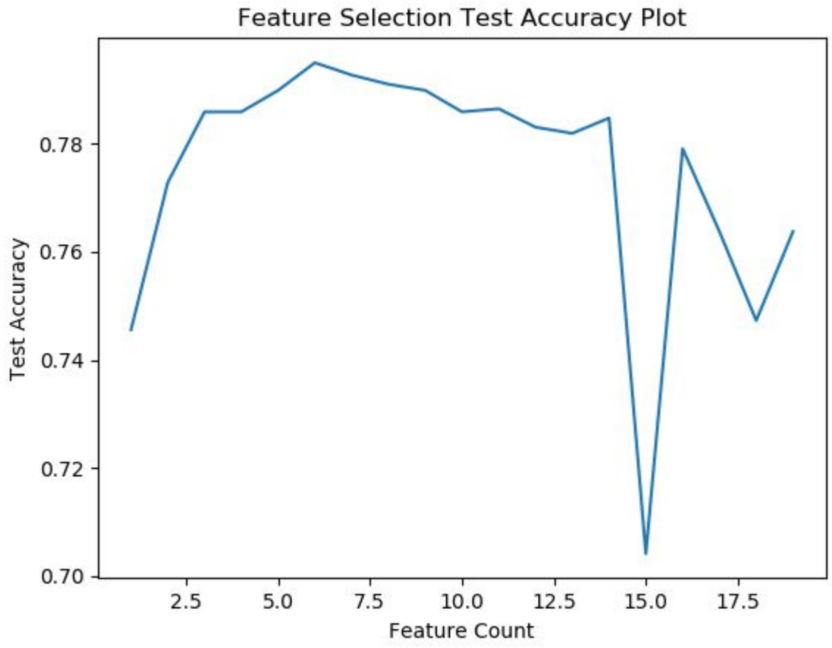
kNN Results

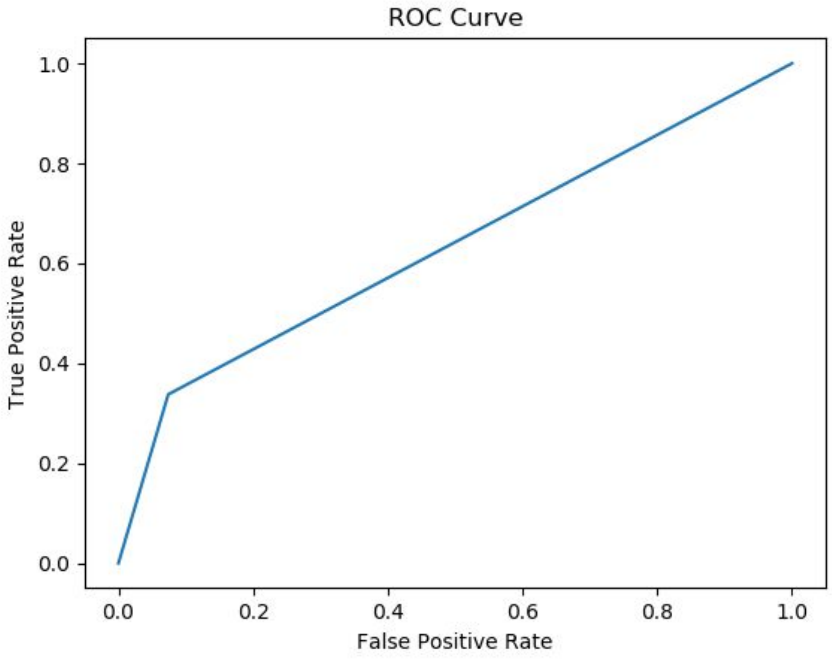
Best Feature Set: **['tenure', 'OnlineSecurity', 'TechSupport', 'Contract', 'MonthlyCharges', 'TotalCharges'].**

Best Test/Dataset Ratio: **0.25**

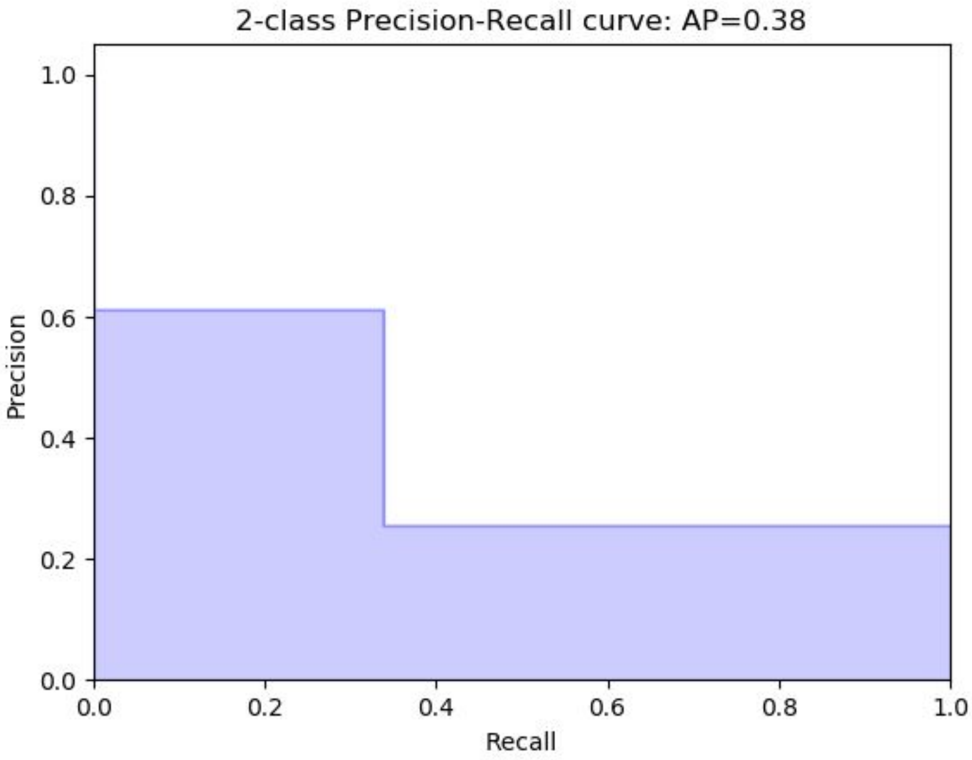
Best Accuracy: **0.792**

kNN Results


Feature Selection Test Accuracy Plot
Test Accuracy
Feature Count
0.70
0.72
0.74
0.76
0.78
2.5
5.0
7.5
10.0
12.5
15.0
17.5


ROC Curve
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.2
0.4
0.6
0.8
1.0

kNN Results



2-class Precision-Recall curve: AP=0.38
Precision
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Recall

kNN Discussion

- Feature selection reduced overfitting & model complexity
- At large nearest neighbor count, K, noise increases & accuracy goes down
- Fast to compute, relatively high accuracy
- Alternative metrics of distance that are context sensitive might improve accuracy
- Most frequent class dominates classification if data is skewed

6. Conclusion

- Accuracy of most of the algorithms capped around 80%
- Answer to the prediction problem is one of the classifiers, preferably the easiest the compute ones
- In the future, better datasets can be used in order to increase prediction accuracy
- Also our code for an experimental commercial use is extremely primitive. This code can be improved so that a user can interact with a GUI

References

[1] Person in doubt, [Online]. Available: https://www.psychologytoday.com/au/blog/children-the-table/201806/when-child-tells. [Accessed: Oct. 11, 2018]

[2] Blast Char ”Telco Customer Churn”. Kaggle. [Online]. Available: https://www.kaggle.com/blastchar/telco-customer-churn. [Accessed: Oct. 2, 2018]

[3] knn Graph, [Online]. Available: https://www.google.com.tr/search?q=knn&source=lnms&tbm=isch&sa=X&ved=0ahUKEwi2xc-fgf_dAhXF_CoKHcdcAfoQ_AUIDygC&biw=1366&bih=613#imgrc=zWukqNn05iZXEM: [Accessed: Oct. 11, 2018]

References

[4] SVM Graph, [Online]. Available: https://www.analyticsvidhya.com/wp-content/uploads/2015/10/SVM_1.png. [Accessed: Oct. 11, 2018]

[5] Random forest animation, [Online]. Available: https://www.google.com.tr/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=2ahUKEwim3KjIgv_dAhUFzaQKHd0XCJ0QjRx6BAgBEAU&url=http%3A%2F%2Fcagriemreakin.com%2Fveri-bilimi%2Frandom-forest-classification-10.html&psig=AOvVaw3CyZGO5GS--DYwxZ2S1bZL&ust=1539369042818892. [Accessed: Oct. 11, 2018]