David Allen

Connor Hammond

Bryce Comer

CSci 290 Data Mining - Team Project

6 December 2023

Introduction:

The goal of this project is to discover the best indicators used in predicting customer churn. We will accomplish this by collecting a data set of interest and become familiar with it. We will get to know it by researching how and why it was collected and by whom. We will search the data for issues and clean the data, then we will begin gathering statistics and visualizations from the data in order to draw conclusions from it. We will use 80% of the data as training data, and then 20% as test data. With the training data we will create a model to provide insights into the data and use test data to calculate the model's accuracy. The dataset we will be using focuses on customer churn and we will be exploring correlations that exist between the churn and other categories.

Materials:

The Telco customer churn data contains information about a fictional telco company that provided home phone and Internet services to 7043 customers in California in Q3. This data was created by IBM to showcase their software with the goal of allowing the user to "Predict behavior to retain customers. You can analyze all relevant customer data and develop focused customer retention programs." [IBM Sample Data Sets]. Churn is the loss of an existing paying customer. In order for a business to experience growth its amount of new customers must exceed its churn rate. It is also typically more expensive to bring in a new customer than to retain an existing customer. For this reason churn is a very important statistic for relevant businesses where profit and growth are important. This data includes which services were purchased by each unique customer along with monthly and total charges. In addition it includes their relationship status, gender, if they have dependents, if they are a senior citizen and their tenure in years. Using this data we will look for patterns and seek to predict churn. We had only minor issues with missing values. There were 11 missing values in total. They were located in total charges, which we filled in with the mean of total charges.

Methods:

We have used pandas and seaborn to create visualizations of the data. From this, we used seaborn to count the number of customers who have churned, which is just over 5000, and those who have not, which is just under 2000. There are a total of 7043 rows in this dataset, representing customers of this fictional company. We have also discovered that a majority of the customers use a month to month contract at around 4000, compared to one year or two year contracts. According to a histogram of tenure, the distribution is bimodal with the majority of customers having used this company's services from 0 to 4 years or 68 to 72 years. We still plan to predict churn, and we hypothesize that tenure, monthly charges, and total charges will affect churn the most.

One of the challenges we have encountered is how to utilize categorical and binary values in our calculations. Many of the algorithms we have used in class require numerical values, which requires us to convert categorical values into numbers. We plan to use the ID3 algorithm on the dataset rather than the Hunt's algorithm specified in the original proposal. We will also use Categorical Naive Bayes and K nearest neighbor.

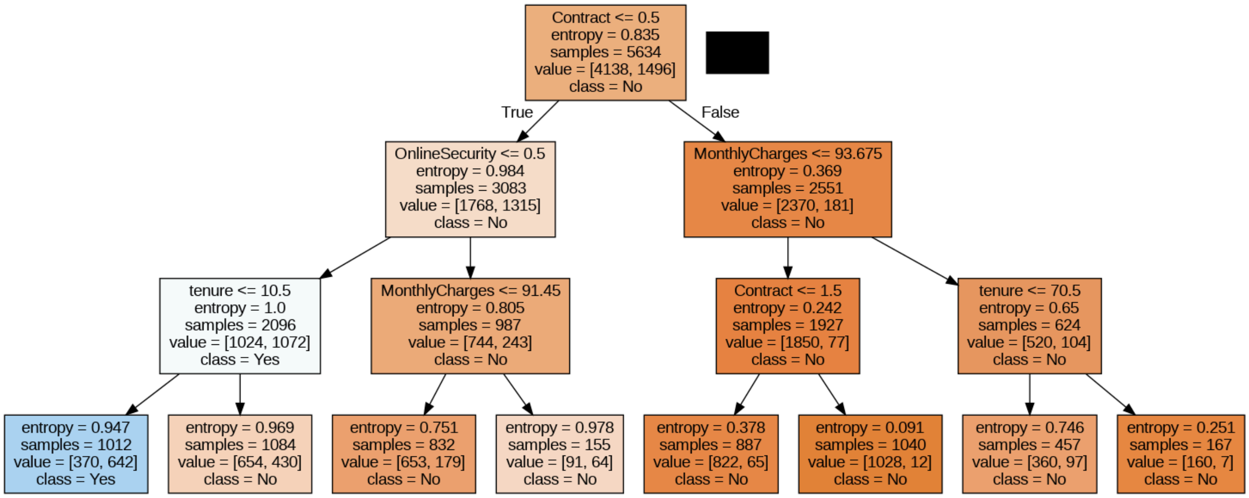
Method 1:

DECISION TREE CLASSIFICATION AND REGRESSION

The first algorithm we decided to use was the ID3 algorithm. The ID3 algorithm uses a top down, greedy approach to construct a decision tree from a given dataset. It starts with the entire dataset as the root node, and then recursively partitions the data into smaller subsets based on the feature that maximizes the information gain. The information gain is a measure of how much the entropy of the target variable decreases after a split. However, we learned that the Decision Tree model in sklearn does not use ID3 but instead CART. CART is a modified version of the C4.5 algorithm which is the successor of the ID3 algorithm. The CART algorithm allows for classification and regression therefore we decided to utilize both and compare results to identify which method was the most effective for the dataset.

The decision tree classifier is predicting customer churn for the Telco data. We preprocessed the data by encoding the categorical features into numerical codes, split the data into 80% training and 20% testing sets, using a random state of 42 for reproducibility, and trained a decision tree classifier with entropy criterion and maximum depth of 3. This means the tree can have at most three levels of nodes and splits based on the information gain. This was made to ensure the model was not too complex for analysis. We used the sklearn DecisionTreeClassifer to fit the model on the training data and predict on the test data. We evaluated the model performance using three metrics: mean absolute error (MAE), mean squared error (MSE), and R2 score. MAE measures the average absolute difference between the actual and predicted values, MSE measures the average squared difference, and R2 score measures how well the model fits the data, ranging from -1 to 1. All of the above was also applied to the decision tree regression model.

Decision Tree Classification



Breakdown of the decision-making process of the classifier:

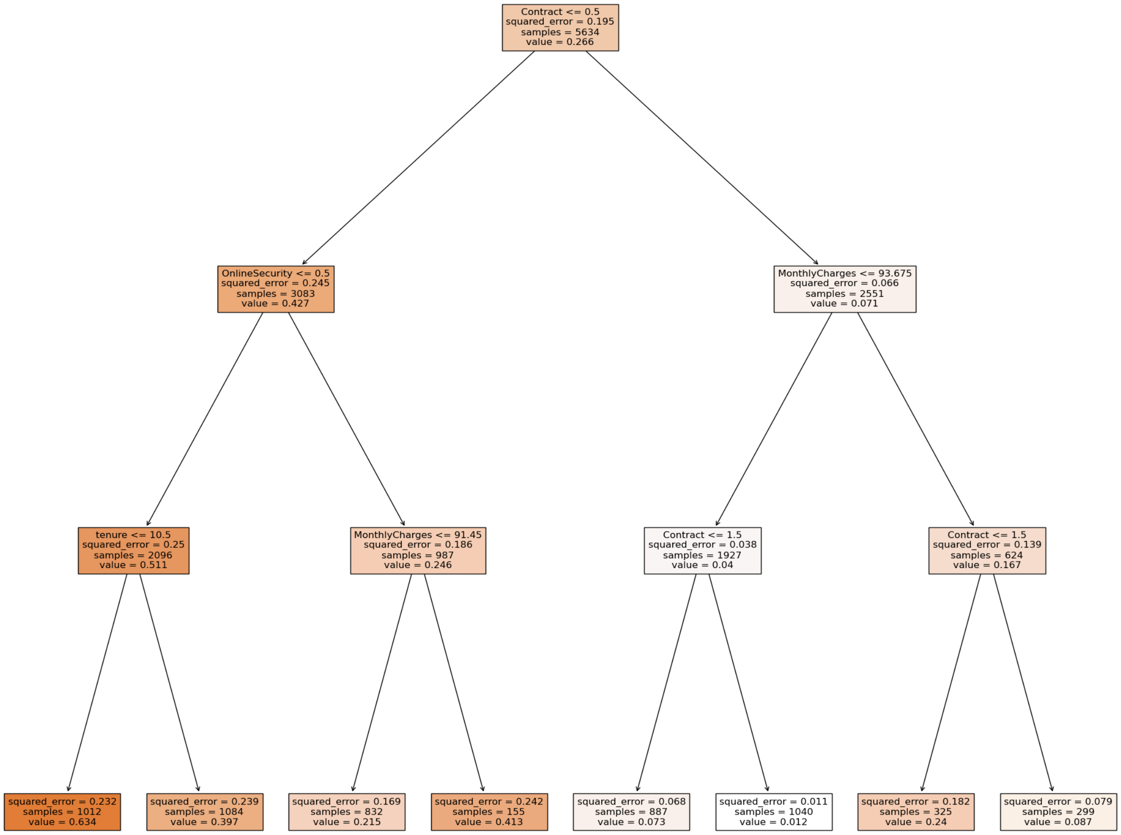
1. The first condition checks if the ‘Contract’ value is less than or equal to 0.50. If it is:
    a. It then checks if ‘OnlineSecurity’ is less than or equal to 0.50. If it is:
        i. It checks if ‘tenure’ is less than or equal to 7.50. If it is, the predicted class is 1 (Churn: Yes). If ‘tenure’ is greater than 7.50, the predicted class is 0 (Churn: No).
    b. If ‘OnlineSecurity’ is greater than 0.50:
        i. It checks if ‘MonthlyCharges’ is less than or equal to 91.45. Regardless of the ‘MonthlyCharges’, the predicted class is 0 (Churn: No).
2. If the ‘Contract’ value is greater than 0.50:
    a. It checks if ‘MonthlyCharges’ is less than or equal to 93.67. If it is:
        i. It checks if ‘Contract’ is less than or equal to 1.50. Regardless of the ‘Contract’, the predicted class is 0 (Churn: No).
    b. If ‘MonthlyCharges’ is greater than 93.67:
        i. It checks if ‘tenure’ is less than or equal to 70.50. Regardless of the ‘tenure’, the predicted class is 0 (Churn: No).

The value 0.5 represents the midpoint between the month-to-month and the one year contract. This means that the decision tree is using a numerical encoding of the contract feature, where month-to-month is 0, one year is 1, and two years is 2. The value 0.5 is the threshold that is used to split the data into two subsets based on the contract feature. Customers who have a contract value less than or equal to 0.5 are classified as having a month-to-month contract, while customers who have a contract value greater than 0.5 are classified as having either a one year or a two year contract. According to the decision tree, churn is most likely to happen when the customer has a month-to-month contract, no online security, and a tenure of 10 months or less. These were the results of MAE, MSE, and R2:

- MAE: 0.2186
- MSE: 0.2186
- R2: -0.1230

The lower the MAE, the better the prediction. In this case, the MAE is 0.2186, which means that on average, the prediction is off by 0.2186 units. The MSE is also 0.2186, which means that on average, the prediction is off by 0.2186 squared units. The R2 score ranges from -1 to 1, where 1 means a perfect fit, 0 means no fit, and negative values mean worse than no fit. The higher the R2, the better the prediction. The R2 is -0.1230, which means that the prediction is worse than no fit. This indicates that the decision tree model is not a good predictor of the class labels because it fails to capture the relationship between the features and the target variable (churn).

Decision Tree Regression



Breakdown of the decision-making process of the regression:

1. The first condition checks if the ‘Contract’ value is less than or equal to 0.50. If it is:
    a. It then checks if ‘OnlineSecurity’ is less than or equal to 0.50. If it is:
        i. It checks if ‘tenure’ is less than or equal to 10.50. If it is, the predicted value is 0.63. If ‘tenure’ is greater than 10.50, the predicted value is 0.40.
    b. If ‘OnlineSecurity’ is greater than 0.50:
        i. It checks if ‘MonthlyCharges’ is less than or equal to 91.45. If it is, the predicted value is 0.22. If ‘MonthlyCharges’ is greater than 91.45, the predicted value is 0.41.

2. If the ‘Contract’ value is greater than 0.50:
    a. It checks if ‘MonthlyCharges’ is less than or equal to 93.67. If it is:
        i. It checks if ‘Contract’ is less than or equal to 1.50. If it is, the predicted value is 0.07. If ‘Contract’ is greater than 1.50, the predicted value is 0.01.
    b. If ‘MonthlyCharges’ is greater than 93.67:
        i. It checks if ‘Contract’ is less than or equal to 1.50. If it is, the predicted value is 0.24. If ‘Contract’ is greater than 1.50, the predicted value is 0.09.

The tree does not say explicitly if the customer churns or not because a regression model predicts a numerical value, not a categorical label. However, we can interpret the value as the likelihood of churn, and set a threshold to decide if the customer churns or not. If we set the threshold of churn to 0.5, then any customer with a value greater than 0.5 is considered to churn, and any customer with a value less than or equal to 0.5 is considered to not churn. With the information provided, churn is most likely to happen when the customer has a month-to-month contract, no online security, and a tenure of 10 months or less.

These were the results of MAE, MSE, and R2:

- MAE: 0.2938
- MSE: 0.1467
- R2: 0.2463

The MAE is 0.2938, which means that on average, the prediction is off by 0.2938 units. The MSE is 0.1467, which means that on average, the prediction is off by 0.1467 squared units. The R2 is 0.2463, which means that the prediction is moderately fit. This indicates that the decision tree regression is a fair predictor of the values.

Result 1:

Comparison: Classification vs Regression

Looking at the trees themselves, the main difference between the two models is that the classification model can directly tell if the customer churns or not, while the regression model can only give an estimate of the likelihood of churn which must be interpreted. Interpretation of the MAE, MSE, and R2 scores comes down to what we are looking for. If we care more about the magnitude of the error, then the classifier model may be more accurate, as it has a lower MAE. If we care more about the squared magnitude of the error, then the regression model may be more accurate, as it has a lower MSE. If we care more about the fit of the prediction, then the regression model may be more accurate, as it has a higher R2. However, none of the models have a confidently high accuracy and have moderate error and low R2 scores.

Method 2:

Categorical Naive Bayes

In our exploration of the Telco customer dataset, one of the methodologies we used was the Naive Bayes algorithm, specifically choosing the Categorical Naive Bayes variant. This choice was made because of the categorical distribution observed in the dataset, where discrete categories are present. In addition, the supervised nature of the algorithm aligns nicely with the dataset's structure, as it includes class labels for each training tuple. The term "Naive" in Naive Bayes comes from the assumption of class-conditional independence, implying that features are independent given the class label. This assumption lessens the computational complexity involved in the algorithm. Naive Bayes has distinct training and prediction phases. During the training phase, the algorithm estimates the probabilities of different categories for each feature, taking into account the provided class labels. To address potential issues arising from zero probability calculations, Laplacian correction is used. These probabilities become important in the prediction phase, where the algorithm calculates the likelihood of encountering a specific set of feature values for a given class. The class with the highest calculated likelihood is assigned as the predicted class.

This algorithm in combination with scikit-learn was used to construct a model trained on our Telco customer dataset. Through the use of this model it is possible to provide historical analysis, future scenario planning, and resource allocation. However, due to the scope of this project it will be limited to historical analysis. This is possible through pattern identification. The predictive model, having been trained on historical data, excels at identifying patterns and trends within the dataset. By using its ability to recognize historical relationships between features and outcomes, the model can highlight recurring patterns in customer behavior, service usage, or other relevant variables. In our case churn prediction. A feature importance analysis can be created which identifies which variables have the most significant impact on predicting customer churn. Features with higher importance are indicative of patterns that strongly influence the target variable.

Result 2:

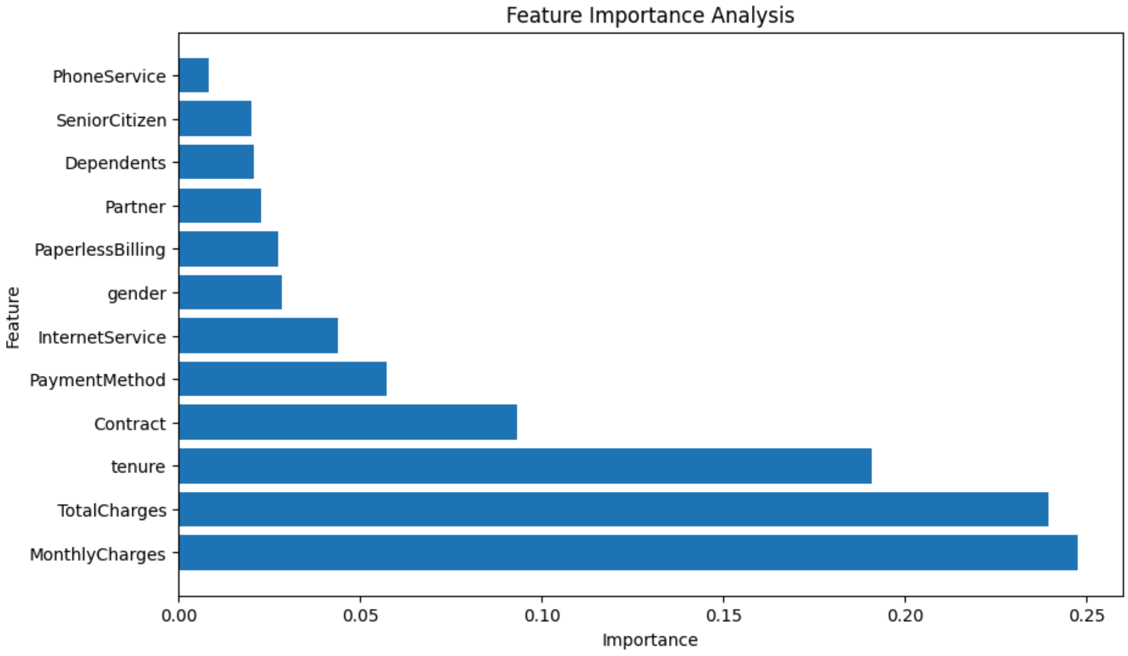
Categorical Naive Bayes

As there are some class-dependent variables present in the dataset two models were trained. One was trained with them present and the other with them absent to assess the change in results. The accuracy scores and classification report were generated for both models and a difference was observed. When class-dependent variables were dropped an increase in the model's accuracy of .04 was observed. This model was subsequently chosen.

Classification Report:

	precision	recall	f1-score	support
False	0.90	0.80	0.84	1036
True	0.57	0.74	0.64	373
accuracy			0.78	1409
Macro avg	0.73	0.77	0.74	1409
Weighted avg	0.81	0.78	0.79	1409

A Feature Importance Analysis was then done which revealed the most important features in deciding customer churn to be monthly charges, total charges, and tenure in that order. This aligns with our original predictions.



Method 3:

K Nearest Neighbor

The last algorithm we tried on the Telco customer dataset was the K Nearest Neighbor algorithm. This algorithm uses a value k for the number of neighbors and calculates the euclidean distance between each of the test points and each of the training points in the set. The model then finds the k number of points with the lowest euclidean distances to the training point, and counts the number of occurrences of each class it is predicting in the group of closest points and predicts the class of the test point with the class that occurred most.

In the case of our dataset, our models are attempting to predict customer churn for Telco, and we have reserved 80% of data for training and 20% for testing. We predicted churn of a customer based on monthly charges and total charges, so the model found the training points with the closest values of monthly and total charges to the test values, and predicted whether the test customer would churn based on whether those who had similar monthly and total charges did. We also tested this with the combinations of tenure and monthly charges, and tenure and total charges. The purpose of this was to determine which of these models had the highest accuracy. We accomplished this using the KNearestNeighborsClassifier module from sci kit learn. We tested the classifier with multiple different values of k to determine which value would yield the highest training and test accuracy.

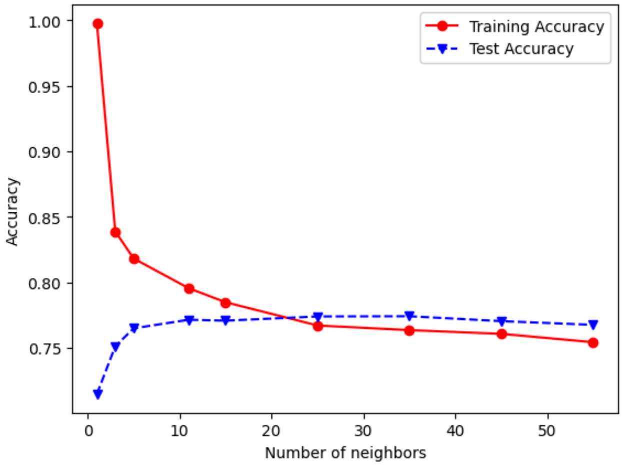
One of the main reasons we decided to use the K Nearest Neighbor algorithm is because of the fact that the model may accurately predict human behavior. Many of the customers for Telco may have the same reasons for churning, which would be represented by short euclidean differences, due to them having similar circumstances. Because of this, we predict that the K

Nearest Neighbor algorithm will have a high accuracy when predicting that customers will churn.

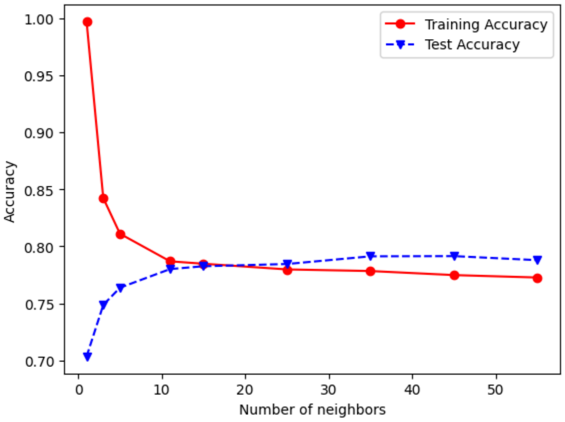
Result 3:

K Nearest Neighbor

The numbers we chose for k were 1, 3, 5, 11, 15, 25, 35, 45, and 55. We specifically chose odd numbers here to make sure that there were no ties for the predictions. Our first test of the model was with monthly charges and total charges. Here, we see that, for the first few values of k, the training accuracy dramatically falls and the test accuracy also increases very quickly, although not to the same extent. After about k=5, the improvements to the test accuracy are fairly marginal, and the training accuracy continues to decrease. Therefore for this model, it seems like k=5 gives us the most optimal results since if we increase k from here, we encounter an issue of overfitting, where the training and test accuracy both start to fall.



Our second test of the model uses monthly charges and tenure to predict churn. Here, we see that both curves level off around k=11. This point seems to be the best for keeping both training and test values accurate. This model also seems to have performed better than the first, since



both curves leveled off slightly higher around 78% accuracy as opposed to the first model, which had around 76% accuracy, according to the y axis of these graphs.

Conclusion:

We have succeeded in creating accurate models to explore our Telco customer churn dataset. We have found the most important categories within the data for predicting customer churn to be customers' monthly charges, total charges, and tenure. This aligns with our original predictions. Along the way we have become experienced with multiple algorithms and techniques for manipulating and cleaning data. We have also gained valuable experience in the creation of models to leverage the power of computers for the process of data mining. Customer churn is a very important metric for businesses which live and die by their ability to recognize historical and future relationships between features and outcomes. Our models were able to highlight correlations existing within the dataset. With future data our models are capable of making important predictions that could help businesses point their resources toward making investments with the maximum possible return. Data mining when properly done is an incredibly powerful tool.