John Behler Complete Project PDF
Table of Contents

Power Point : Page 2-7

Tableau Dashboard : Page 8-9

Project Overview : Page 10-17

An Analysis of Customer Bank Churn

Machine Learning Capstone Project: DTSC 691

By: John (Jake) Behler

Background

- I looked at banking customer data to predict and analyze customer turnover. This can be important for incentivizing customers who are at risk of leaving to stay at the bank. Retaining customers is important for profit because it is costly to sign up new customers. When customers leave the bank (especially premium customers), it cost money to recover the loss and find new clients. High customer churn puts a big dent into revenue and profitability suffers because of it. I looked at prediction models and visualizations to better understand the data and give insightful information to the bank analyze which clients are likely to leave the bank. This will help the bank implement strategies to keep those customers at the bank.

Steps

- Gather data from Kaggle
- Data preparation, data cleaning, and data transformations
- Machine Learning models to make predictions on data in Jupyter (Python)
- Create visualizations in Tableau to analyze data for finding trends and patterns
- Create a website and PowerPoint to present insight

Insight from Tableau

- Complain is highly correlated with churn rate (99.51%)
- Germany has double the turnover rate compared to Spain and France
- Inactive members have higher churn rate than active members in both males and females
- Churn rate is higher in females (a difference in about 9%)
  - Males tend to stay slightly longer at the bank then females (0.1 years ~ a month)
- Customers who had 3 or 4 products have very high churn rate (82% and 100% respectively) compared to customers who only had 1 or 2 products (27.71% and 7.6%)
- Credit Score didn't have a big impact on customer churn rate
  - Except churn rate increased significantly for client who had credit score under 395
- 45-65 years old is where the highest turnover rate comes from
- Slight decrease in churn rate as satisfaction rates increase (besides France which is opposite)

Takeaway

- From out ML algorithm, Complain has very high predictability for Exiting the banks
  - If our complain variable is known to us before the customer leaves, this is critical and valuable information. This can be predictable to know that a customer will likely leave the bank. However, if this is only recorded at the time of exit, then this will not lead to impactful decisions.
- We have learned from out Tableau Visualizations about which types of groups have a higher churn rate which is important information
  - Complaining, being in Germany, being an inactive member, being a female, having 3 or 4 products, having a credit score under 395, being between the ages 45-65, and having a low satisfaction score all increase the rate of churn.

Challenges & Reflection

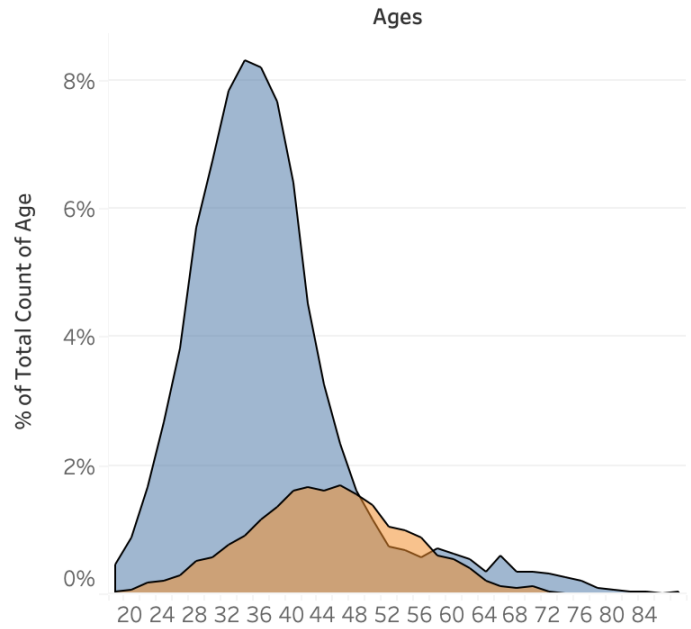
- From the beginning, I could have looked at engineering more features to possibly add to my model
  - I tried different methods such as binning variables, however, I only briefly looked at polynomials and interaction terms between variables
- Poor model performances with Tenure as my response variable
  - I had difficulty finding linear and non-linear relationships with this response variable and choosing the appropriate models
- Grid Search and parameters
  - I did do a grid search of my gradient boosting model, but I could have looked more into tuning the parameters and a grid search for random forest or other models
- Next time for my Tableau dashboard, I would expand the level of interactivity to include more complex filters or additional layers.

Customer Bank Churn Dashboard: John Behler

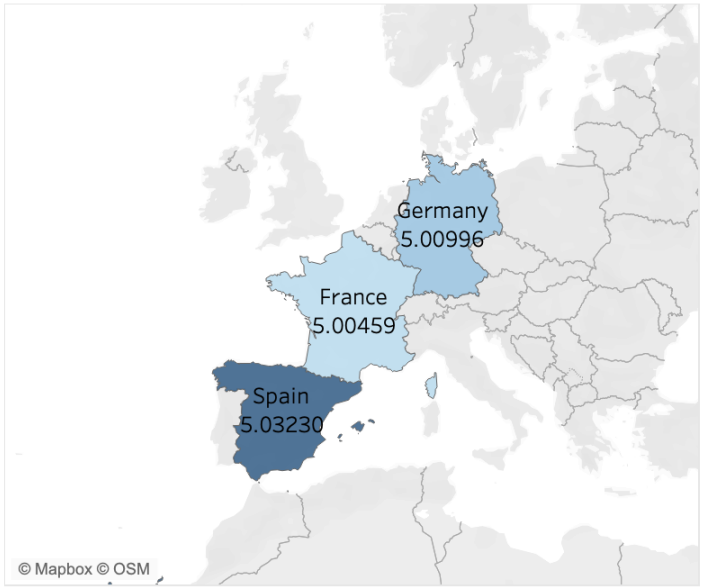
Geography All

Gender All

All Churn Proportion by Age



AVG Number of Years at the Bank by Country



Geography
- France
- Germany
- Spain

Churn Rate
19.638 21.797

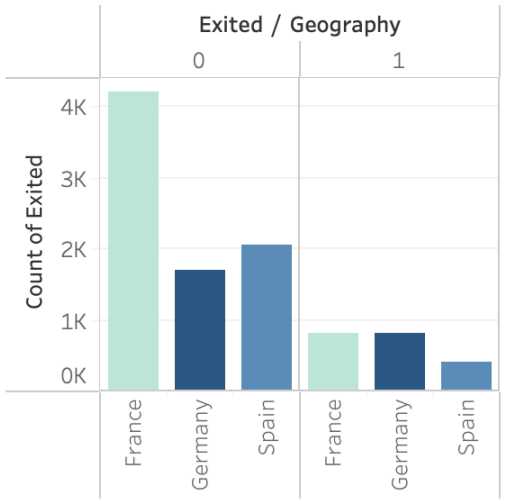
Avg. Tenure
5.00459 5.03230

Exited
- 1
- 0

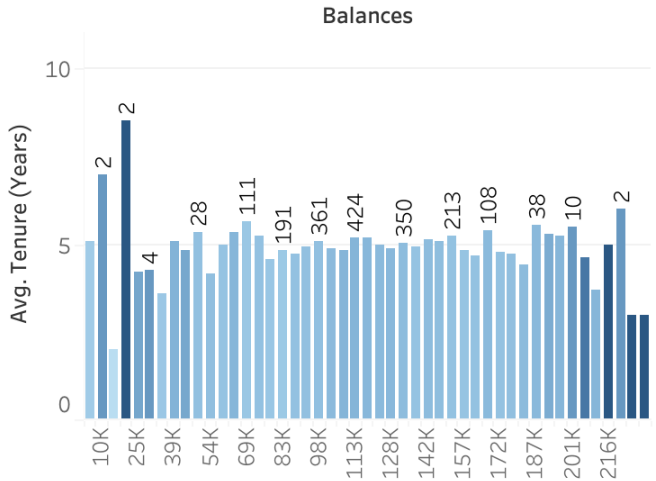
Average Age Exited

Exited	
0	37.408
1	44.836

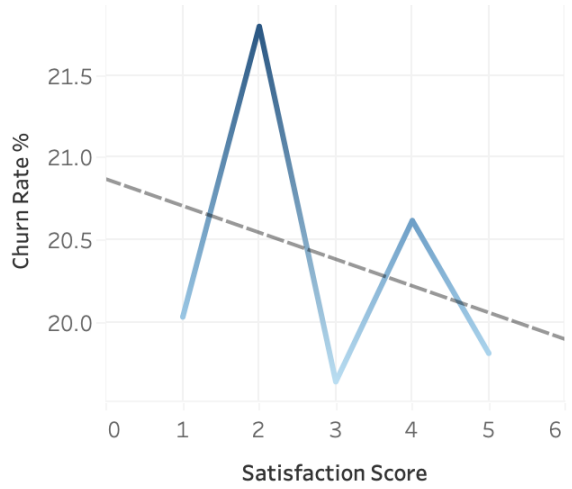
Exited by Geography



Churn Rate By Balances



Churn Rate By Satisfaction Score



Customer Bank Churn Rate Dashboard: John Behler

Exited Count	Stayed Count	Average # of Years	Overall Churn Rate
2,038	7,962	5.013	20.38

Geography All

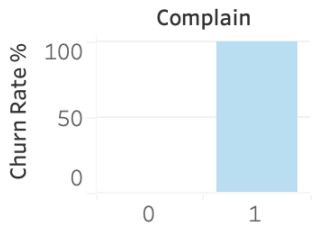
Gender All

Geography
France
Spain
Germany

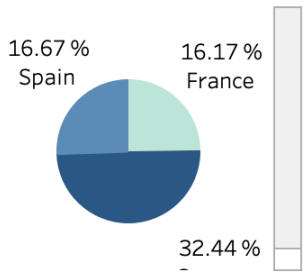
Churn Rate 17.56 100.00

Exited 1 0

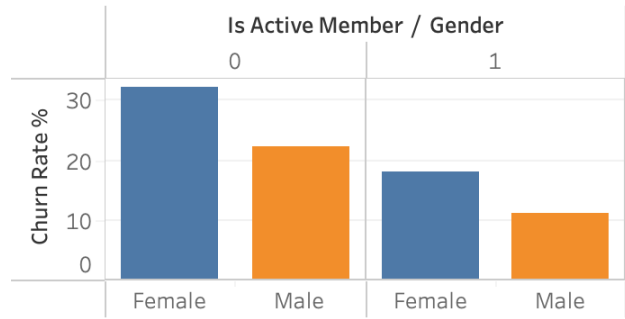
Churn Rate By Complain



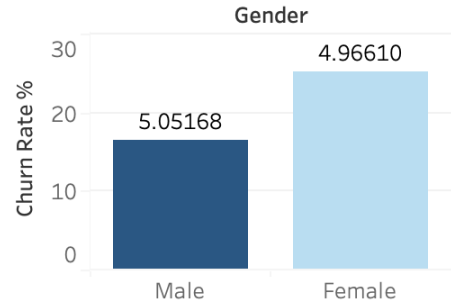
Churn Rate by Geography



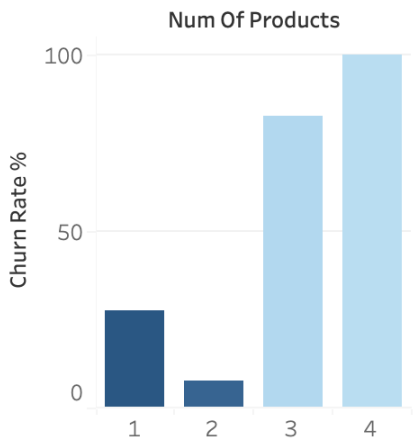
Churn Rate By IsActiveMember



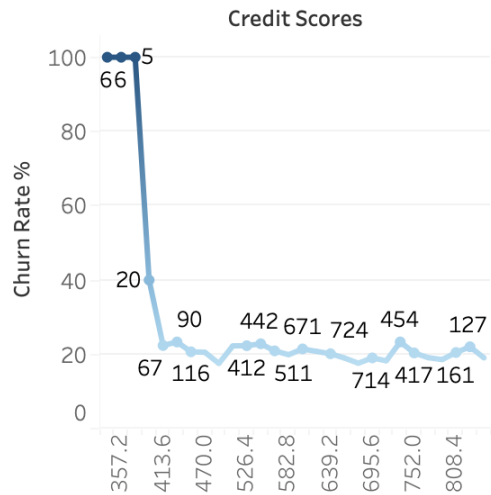
Churn Rate By Gender



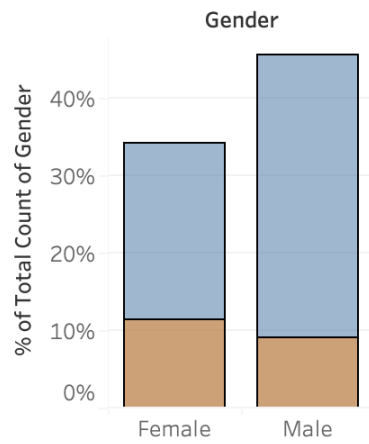
Churn Rate By Num of Products



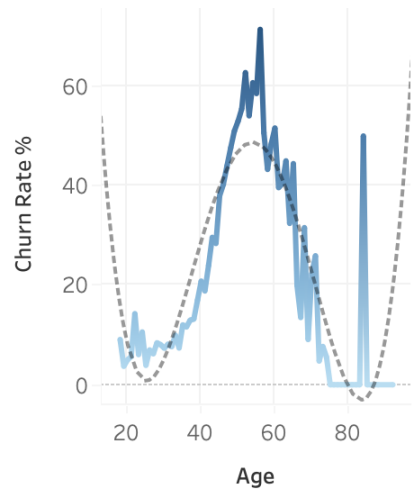
Churn Rate By Credit Scores



Churn Proportion Gender



Churn Rate By Age



John Behler
DTSC 691
Project Overview
ML Capstone Project
An Analysis of Customer Bank Churn

Project Objective and Scope

In this project, I looked at customer banking data and analyzed customer churn. The domain was customers at a specific bank and other banks would not be able to implement this model because different banks have different types of customers. The objective and aim of the project was to create a machine learning algorithm to predict the number of years a specific customer would be at the bank based on their user profile, banking information, and demographics. The expectation was to create a solid model that a bank would be able to make predictions on current clients based on customer information. If the bank could accurately predict the length of time it would take a customer to leave the bank, they may be able to intervene to keep the customer before it is too late. This specific model would be predicting customers' tenure (# of years, 0-10) which doesn't give us the full picture of trends and analysis but would be helpful in identifying which individual customers leave the bank sooner. This is important information and insight so the bank can retain their customers which leads to profit. Retaining customers (especially premium customers), is very important so revenue and profitability don't take a hit so the bank doesn't have to worry about signing up new customers which leads to loss of profit. When clients leave the bank, it costs the bank money which could be preventable if strategies are implemented to incentivize customers who are at risk of leaving to stay at the bank. We just need to learn more about which customers are leaving and why and I used prediction models to look at that in Jupyter. I also looked at Tableau to visualize patterns and trends across different categories to see if there was a trend in what is causing customers to leave the bank.

Data Acquisition

In the banking customer churn data there are 10,000 entries and 18 features. The first three columns, row number, ID number, and Surname, were not important in our analysis because those likely do not have any correlation or patterns with how long a customer would stay at the bank. Since that is clearly not important data for our ML algorithm, if we keep it in our analysis it will create noise and make a less predictable model. The other variables were included in the model because we wanted to see if there was a correlation or trend between them and the length of time a customer stays at the bank. Some of the variables in the dataset that were included in my model are Age (18-92), Credit Score (350-850), Complain (0-No, 1-Yes), Balance (0-250898.09) and Satisfaction Score (0-5) because there is a possibility that people leave the bank because of one or more of these reasons. The data didn't contain any dates or timestamps, so analysis couldn't be done on trends and time series. I chose Tenure (0-10 years) as our response variable but after trying many different types of models, binning

the response variable, and deleting certain rows I could not come up with any good models to implement. I tried linear regression, logistic regression, SVM, random forest, gradient boosting models, and performed a grid search, but no models performed well in predicting Tenure. I switched my response variable to “Exited” and got different results and better predictable results. The results gave much better accuracy which had a lot to do with the “Complain” variable and its high correlation with the response variable. This is important to take note of. If the complaint happens at the time of leaving the bank, then this won’t be helpful in being able to predict customers that leave. However, if they complain days, weeks, months, or years before leaving, this would be a very helpful model for a business as there is high predictability. This dataset was from Kaggle and I selected it because it is a real world scenario. I chose to pick a project that is a real possible situation to be able to add to my portfolio and I could showcase my data science skills across multiple industries. This dataset provides multiple features that allow me to run regression and analyze the data at a high level trying multiple different models. It allowed me to critically think about different ways to look at the data and come up with models that could be implemented as well as thinking about different ways to look at the data and feature engineer variables. Banks want to know the insights and reasoning for what causes customers to stay at a bank longer, and also what motivates a customer to leave the bank. If they can understand the data and the customer's incentive to leave, they might be able to intervene and come up with strategies to keep them longer. Sometimes certain groups of people or demographics yield different motives, and it is my job to look at the data, and see if there is any correlation between the customers in the dataset and them leaving the bank. High customer churn and turnover rate is costly for banks, so they would want to implement methods to retain those customers, maybe with incentives or new policies and this can be done better with data analysis.

Exploratory Analysis

First, I uploaded the dataset into Jupyter and imported the necessary libraries. Then I checked the shape of the data and noticed 10,000 observations and 18 features. I applied .head() to my dataset to look at the first 5 rows of the data. This gives us a quick look of the column names and what sort of data we are dealing with. I noticed right away that the first three columns (Row Number, Customer ID, and Surname) were going to be useless in predicting customer churn so I knew I was going to delete them before doing analysis because we do not want to incorporate them in the models because that would negatively impact our model. I noticed three columns had categorical data which was in string format, so I was going to have to encode those values so I could use them in my regression models by converting them to numerical values. I also noticed that we had numerical and ordinal data. Then I looked at the categorical columns to ensure that everything was spelled correctly by using .unique() and noticed there were no inconsistencies in the spelling. There are three countries, two genders, and 4 card types, and there were no typos. I wanted to make sure that there were no duplicated rows so used duplicated().sum() and found no copies. If there were, I would have to investigate and most likely delete that row. Next I looked at .info() to see if there were any null values which our dataset did not have and checked the datatypes of our features. We can see the three categorical variables are “object” and the rest are numerical (int64 and float64) which is

expected. I then checked the statistical summary of the variables we are dealing with by using .describe(). This function gives us the mean, standard deviation, and other basic statistical info such as minimum and maximum values which can be useful looking for outliers even though a boxplot will also give us this information. I looked at the correlation matrix and didn't see any correlation between the features and my response variable (tenure). I did feature engineer two variables for my correlation matrix, called "new_balance" which created a binary variable whether or not the balance was 0. If the balance was zero, the new variable was coded as 0, and if the customer had any balance at all, the new variable was coded as 1. I also created a new variable called "new_age" that split the age variable into young customers and older customers. There was still no correlation between these and the response variable so it wasn't worth adding to my models.There is a chance they may have had positive benefit in my models but since it wasn't in my initial dataset I decided to leave that out. I then looked at a pairplot and histogram of the features to see if there were any patterns but since the variables were mainly categorical, binary, and uniformly distributed there was no reason to do data transformation on most of these features besides age which was right skewed. I took the square root of this column to make it more normally distributed for our model so it would perform better. I looked at the boxplot of the numerical features to check for outliers and there were outliers for old age and low credit scores. They were technically extremes and further from the center of the data but they were normal variations in real life customers and it didn't look like they were coded for something or mistyped so I kept them for my analysis. There are some customers who actually have low credit scores, and are older than normal. Also since I was scaling the data, the effect of these "outliers" would be mitigated. I looked at an OLS regression model to look at p-values for feature selection. High p-values mean there is no statistically significant relationship between the response and the predictor so I removed them. Then I created multiple visualizations in Tableau. I created a calculated field to look at the churn rate, and compared that to many different variables such as Gender, Age, Geography, Balance, etc. I was able to come up with many key findings that show differences in churn rates in certain groups. As mentioned in my powerpoint, Complain, being in Germany, being an inactive member, being a female, having 3 or 4 products, having a credit score under 395, being between the ages of 45-65, and having a low satisfaction score, all increase the risk of a customer leaving the bank. This is crucial to be able to target these groups to find ways to keep them at the bank longer. During my analysis in Jupyter I ran linear regression, logistic regression, SVM, random forest, gradient boosting models and performed a grid search while tuning the parameters. I will go further into detail later in the model training and model evaluation section. However, the main takeaway is I struggled with creating a model that would accurately predict the number of years a customer would stay at the bank. I tried different methods and the only main conclusion from my machine learning models is that Complain variable is highly correlated and predictable in determining if a customer will leave the bank or not (however, not good at predicting the number of years at the bank).

Data Preparation

Like I mentioned above, in my dataset, there were no missing values, so fortunately I didn't have to decide whether or not to delete those rows or impute the median/average value. I

also encoded my categorical columns to numerical values so they would be ready for machine learning models and sorted my features based on numerical, ordinal, or categorical data. Then I created a pipeline and column transformer that scaled and encoded the features based on if they were numerical, ordinal, or categorical data. I had variables that had different ranges in numerical values so, scaling the data is important to make sure the model is balanced and not one feature is “more important” than the others. When looking at the histograms of the data, I noticed the age variable was right skewed so I took the square root of it to make it more normally distributed. The other variables were normally distributed, categorical, binary, uniformly distributed, or had few options so there was no need to transform this data. I did look for multicollinearity between my features by looking at a correlation matrix. There were two that were highly correlated, Exited and Complain, that could affect the model. However, after speaking with the TA (and I may have misunderstood exactly what was said), I believe I was told to leave all variables in my model for the analysis. I then looked at the value counts of my categorical data and the splits were balanced which is important.

Model Training

The intent of my model training was to come up with a model with good predictability of the number of years a customer would stay at the bank. I explained earlier, the particular reason for this is to identify churn before it happens. If we can identify when a customer is about to leave the bank, we can try to save that customer from leaving. I left all of my features in this model (age, gender, geography, etc) and after preprocessing them (scaling, transforming, encoding, etc), the training set was ready for our model. I set the response variable to ‘Tenure’ which is a numerical variable with a range of 0-10. Even though there was no correlation between the variables and tenure, I started with a basic linear regression model. The model performed very poorly. I will explain in the next section, “Model Evaluation”, how I knew the model was poor. The next step I took was I looked at an OLS regression model to get a sense of how impactful the features were in terms of predicting “tenure” based on the p-values. I evaluated the p-values and removed some of the features that had high p-values because that indicated no statistically significant linear correlation between the variable and the response and might create noise and make our model less predictable. I then ran another linear regression model with less features but that did not improve the model. I decided to run a lasso model to shrink coefficients of less importance and penalize the coefficient and is used when there are many features. Lasso helps to apply this regularization and identify features that are not valuable to our model and according to our correlation matrix, there were many features that didn’t seem significant. That also delivered poor results. I then created an SVR model to look for non-linear patterns and it is more flexible. SVR is good at capturing complex trends in our data that linear regression models cannot. Our linear regression models didn't provide good results and maybe the models were too simplistic. I got a poor model from that as well. I decided to run a random forest regression model which is a tree based algorithm to capture feature interactions and reduce overfitting to improve prediction accuracy. With poor results from that as well I created a gradient boosting regression model which is a tree based model to look for nonlinearities and complex patterns. That also got poor results so our data for predicting tenure is messy and not predictable. Our original linear regression model performed the best of all of

our models, however I looked at two regression models, and a lasso model (with all coefficients of 0), and none of them improved our model significantly so I figured there was no linear relationship between the variables I chose. So I decided to run a grid search with my gradient boosting model because it performed better than the random forest and SVM model. I was hoping maybe it would be able to understand the more complex data and minimize the error. A grid search will optimize the hyperparameters to create a better model to be able to generalize to unseen data even though it's unlikely to significantly give us better results. Since our gradient boosting model was underfitting, I selected parameters that would help correct ('n_estimators', and 'max_depth'). After running the grid search to our gradient boosting model, we did get slight improvement, however, it wasn't significant which was expected. It seemed our models were clearly not able to find any pattern in the data to predict our continuous response variable. I tried to bin 'Tenure' and see if there is any way to predict long vs short term customers in a logistic regression model (classification). Even though there wasn't any predictive power in trying to guess the exact number of years, I was hoping maybe there was some trend in the data to see if a customer's demographics would be able to predict a short vs long term client. After running this model, we still didn't get the results we wanted with precision slightly over 0.50. I then deleted the rows where the customer hadn't left the bank yet. If they haven't left yet then maybe those customers that are still at the bank are affecting our analysis. Our model performed just as badly so after trying multiple different things (binning data, deleting rows, checking different models), I have come to the conclusion that from our data it is very hard to predict tenure from our features even after trying multiple different options. Our models haven't accomplished predictive performance, which makes sense because of weak feature correlation. One suggestion would be to include other features and new customer demographics in the dataset to see if there is something else we can use to predict the number of years a customer will stay at the bank. In order to still gain some insight from the data, I continued to try to learn more about the data by looking at different variables for the response variable. I changed the response variable to "Exited". We have data on if the customer has left the bank or not, so I wanted to look to implement a model that can predict whether a customer will leave or not. This model gave us great results with metrics over 0.997 which suggests a great model however our variable "Complain" had a 0.996 correlation with this response variable so it's important to keep that in mind for why our model performed so well. I quickly looked at other models but SVC, gradient boost, and random forest did not perform any better. Since "Complain" had such a high correlation I removed that variable from our model and looked at the metrics again. Our Gradient Boosting Regression Model performed the best with good accuracy and precision, but still a poor recall and f1 score. The information I took from these models is that "Complain" is really the biggest predictor of if a customer will leave the bank or not and the data isn't easily able to predict the number of years a customer will stay for. Maybe something to try next time would be to use a grid search and change the parameters for the model without "Complain" and with "Exited" as the response variable.

Above, I explained individually why I chose each model to run. Initially I ran Linear regression (one with all the features, and then another without features with a high p-values) and Lasso model to capture linear relationships. When those models didn't work, I looked at other models such as random forest, gradient boosting, and SVC models to look for non linear

relationships because these models are good at interpreting complex trends in the data that linear regressions can't. When these models were poor, I binned my response variable and looked at classification models to see if that would change my predictions and produce a better model. However, I was unable to get a good model with this as my response variable.

Once again, the models I used were linear regression, lasso, SVR, random forest regression, gradient boosting, logistic regression, random forest classification and gradient boosting classification. I didn't do a whole lot of hyperparameter tuning because my models weren't providing good results, and I decided to focus on different models to capture non-linearity over changing the parameters. For the grid search that I did apply to my gradient boosting model, I did include hyperparameters that would help with my model that was underfitting the data. These are the parameters I used, pg_gbr = {'n_estimators': [150, 250, 350], 'max_depth': [2, 4, 6]}. If I were to do this project again, I would look more at tuning the parameters for this model as well as creating parameter grids for different models.

Model Evaluation

In my models, I used 5 fold cross validation which splits the data multiple times and runs different regressions. This helps to ensure that the model isn't overfitting and helps generalize better to unseen data. While running my linear regression, lasso, SVR, random forest, and gradient boosting models my output was very poor for all the models. I explained above in the model training section the reason why I decided to run each model. In order to look at the effectiveness of the model, I looked at the metrics MAE, RMSE, and r^2 for both the training set and the test set which help determine how well the model did. The reason I did both was to make sure the training set wasn't overfitting the data. So if the metrics were very good for the training set, but poor for the testing set, then I know that the model actually just learned the patterns for the training set and wont generalize well for the population or our target audience. MAE stands for the mean absolute error, which calculates the difference between the predicted values and the actual values. You want this value to be minimized but interpreting it depends on your response variable. In my instance, the range of my MAE in the models I ran were between around 2.54-2.59, which means our predictions were off by an average of about 2.54 years. This isn't very helpful because our response variable is on a scale of 0 -10 years which is an average of 25% off. RMSE stands for the root mean squared error. This is similar to the MAE as it measures the difference between the expected and actual values. However the RMSE penalizes more when there is a larger residual, so RMSE will always be bigger than MAE. r^2 is the coefficient of determination. A higher r^2 means your model is doing a good job of making accurate predictions. In all of our models our r^2 was not very good and ranged from -0.05 to 0.0006. Most of our r^2 from the testing set were negative which indicates our model performs worse than just selecting the average every time.

Then I switched my response variable from continuous to binary and ran a logistic regression model. The metrics for determining the effectiveness of these models are different. The metrics I looked at were accuracy, precision, recall, and f1 score. Accuracy is the number of predictions made correctly (since it is binary) out of the number of total predictions. Of course you want a

high accuracy. Precision is the accuracy of your positive predictions. Recall is the percentage of the positives that you caught out of all the positives. F1 score is a balance between recall and precision. For all of these metrics, the lowest value possible is 0, the highest possible score is 1, and the closer to 1, the better the model. For our first logistic regression model all four of our metrics were higher than 0.99 which means our model predicted very well. This is mainly because one of our variables has a high correlation of 0.996 with the response variable. Since about 20% of our population from our dataset exited and 80% stayed, a baseline model would just predict stayed 80% of the time with an 80% accuracy but 0% recall for exited. The fact that all of our model metrics show higher than 0.997 indicates good results. When I removed the highly correlated variable “Complain” and looked at the metrics for multiple models, I saw a decline in model performance. Our accuracy was good at 0.866 with our gradient boosting model but recall and f1 score weren’t too good. In the future, looking at a grid search to find tune these models would be a start to improve the models. This suggests that “Complain” was a key predictor for our model. I mentioned this in my PowerPoint, that if “Complain” is only found out at the time of exiting the bank, this model would be useless because at that point, it is too late. However, if not this would indicate that if this bank wanted to try to decrease churn, they could implement this model to predict who is going to leave the bank to customize strategies to retain these customers.

I have submitted to brightspace, uploaded to google drive, and emailed a .ipynb of my jupyter code. This includes EDA, data preparation, model training, and model evaluation which was described in this Project Overview

Link to Website:
https://johnbehler.wixsite.com/capstoneproject

Data can be Found:
https://www.kaggle.com/datasets/radheshyamkollipara/bank-customer-churn/data

Sources

- https://docs.google.com/document/d/1DdEklVPbkJHaxiDHygGbs4dpCpJkQGlV8zuNKUFbiG0/edit?tab=t.0#heading=h.2o7tbfebd65q
- https://docs.google.com/document/d/1jWjIrHvKg5SUlu0FROZICppLdeJfu6BscY66xwbQe1U/edit?tab=t.0
- scikit-learn. “3.2.4.3.2. Sklearn.ensemble.RandomForestRegressor — Scikit-Learn 0.20.3 Documentation.” *Scikit-Learn.org*, 2018, scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html.
- Rendyk. “Distinguish between Tree-Based Machine Learning Models.” *Analytics Vidhya*, 28 Sept. 2024, www.analyticsvidhya.com/blog/2021/04/distinguish-between-tree-based-machine-learning-algorithms/.
- “ROC Curve with Visualization API.” *Scikit-Learn*, 2025, scikit-learn.org/stable/auto_examples/miscellaneous/plot_roc_curve_visualization_api.html.
- “Statsmodels.regression.linear_model.OLS — Statsmodels.” *Www.statsmodels.org*, www.statsmodels.org/dev/generated/statsmodels.regression.linear_model.OLS.html.

- “Machine Learning: Linear Regression — Data Analysis with Python - 2020 Documentation.” *Github.io*, 2020, csmastersuh.github.io/data_analysis_with_python_2020/linear_regression.html. Accessed 28 Apr. 2025.
- Seong, Soonmo. “Random Forest with Grid Search.” *Cloud Villains*, 9 Apr. 2024, medium.com/cloudvillains/random-forest-with-grid-search-b739fb0da311.
- “MS in Data Science | Online | $9,900 | Eastern University.” *Eastern.edu*, 2022, www.eastern.edu/academics/graduate-programs/ms-data-science. Accessed 28 Apr. 2025.
- “Box Plot in Python Using Matplotlib.” *GeeksforGeeks*, 10 Apr. 2020, www.geeksforgeeks.org/box-plot-in-python-using-matplotlib/.
- Waskom, Michael. “Seaborn.heatmap — Seaborn 0.10.1 Documentation.” *Seaborn.pydata.org*, 2024, seaborn.pydata.org/generated/seaborn.heatmap.html.
- “Seaborn.histplot — Seaborn 0.11.2 Documentation.” *Seaborn.pydata.org*, seaborn.pydata.org/generated/seaborn.histplot.html.
- “RocCurveDisplay.” *Scikit-Learn*, 2025, scikit-learn.org/stable/modules/generated/sklearn.metrics.RocCurveDisplay.html#sklearn.metrics.RocCurveDisplay.from_estimator. Accessed 28 Apr. 2025.
- S_Naghiyeva. “What Is Predict_proba and [:,1] after (X_test) in Code?” *Stack Overflow*, 2 Sept. 2022, stackoverflow.com/questions/73582838/what-is-predict-proba-and-1-after-x-test-in-code.