Fulton Bank Datathon

Team 3

Jenny Chen, Katie Lee, Mark Rauschkolb, Kevin Sun, Hanson Wang

Project Description

- **Task: 1) Segment Fulton Bank's customers, 2) Identify customers who are most likely to churn, and 3) Predicts when customers are likely to leave**
- To segment Fulton Bank's customers and understand which customers were most likely to churn based on provided features, we implemented 3 classification approaches: decision tree analysis, logistic regression, and XGBoost
- Following classification, we created churn datasets and identified probabilities of churn for the customer segment most likely to churn using a shifted-Beta-Geometric model

Customer Segmentation & Least likely to churn

- To segment customers based on their likelihood of churn, we used three distinct models: decision tree analysis, logistic regression and xGBoost
- xGBoost does not need feature engineering and scales well, however it is a bit-more time consuming to process than the other models. Decision tree, on the other hand, offers us an easy way to interpret our analysis
- Each model has their pros and cons, and we arrived at slightly different results and accuracy rate from those three models and will discuss our findings in the presentation

When they will churn?

- The sBG model ultimately answers questions relating to customer retention and is applied to a churn dataset.
- Individual customer behavior story: at the end of each period, a customer renews his contract with probability **$1 - \theta$ (geometric distribution)**
- Churn propensities (**θ**) varies across customers based on observable and unobservable characteristics
- To model unobserved heterogeneity, assume:

$$\boldsymbol{\theta} \sim \text{Beta}(\alpha, \beta)$$

Agenda

Member Introduction & Projection Description

Data Processing and Cleaning

Classification Methods

- Decision Tree Analysis
- Logistic Regression
- XGBoost

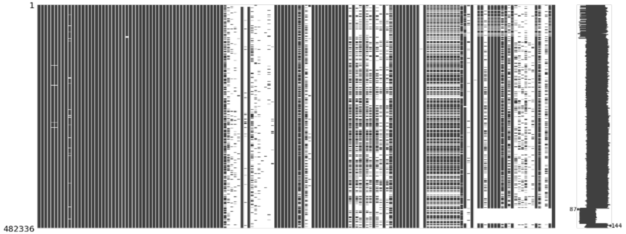
Modeling Time to Churn

Takeaways & Conclusion

Data Preprocessing

Data Filtering

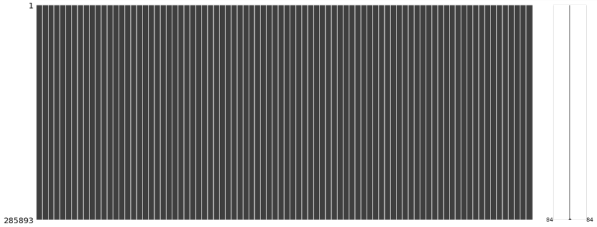
- Just like in our previous model, we only kept the most recent data of consumers (only those that were added after Dec. 2018.)
- This dataset includes 154 variables in total, and I dropped a few repetitive variables, and other variables like loan balance
- However, as shown in the missing data visualization below (msno.matrix), many variables still contain large amounts of missing data



Data Cleaning

- I dropped the columns that have more than 200,000 null values, and then dropped rows that have any null values at all
- The result is a dataset with 84 columns and 258,593 rows

```
droplist=[]
for i in range(len(x.columns)):
  if x.iloc[:,i].isna().sum()>200000:
    droplist.append(i)
x.drop(x.columns[droplist],axis=1,inplace=True)
filtered_x=x.dropna(axis=0,how='any')
```



Dummies Variables

- We believe there are a couple of non-numeric variables (Bank Branch, low / middle / moderate income, customer's lifetime value categorization) that could be significant to predicting a customer likelihood of churn

```
dummies1 = pd.get_dummies(x.BANKID)
dummies2 = pd.get_dummies(x.lmi)
dummies3 = pd.get_dummies(x.quad)
```

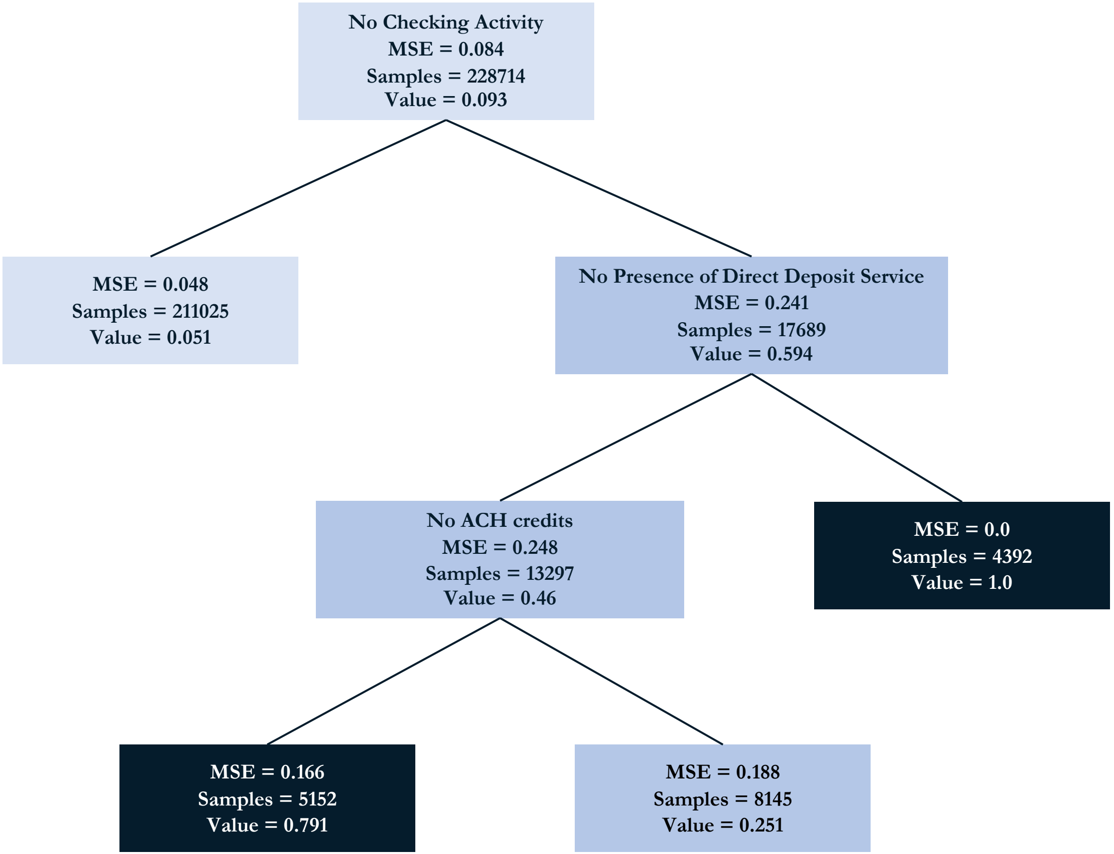
Train Test Split

- We split the data into train test with a test size of 0.2 in order to evaluate the model

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(
  filtered_x, y, test_size=0.2, random_state=42)
```

Agenda

Decision Tree Analysis



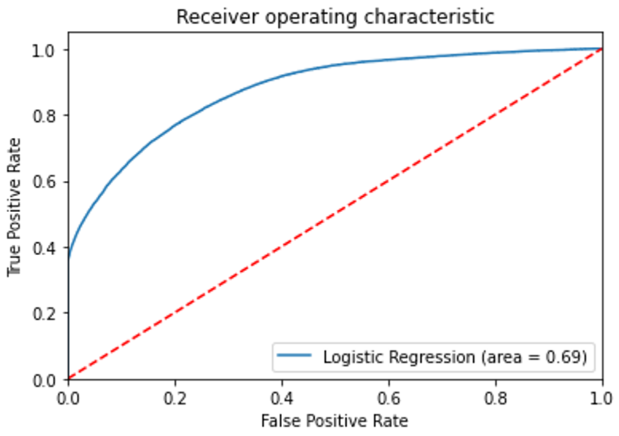
Method and Interpretation

- We ran a decision tree regressor on the y-train and x-train data with a maximum **leaf nodes of 4,** for simplicity's sake.
- Our model yields an **R-squared value of 0.48,** while this is not necessarily an ideal accuracy score, it does paint a basic picture of the kinds of customers that would be likely to churn
- As shown in the decision tree on the left, the key determining factor for customer churn, according to the decision tree regressor, is whether a customer customer has had any checking activity

```
regtree = tree.DecisionTreeRegressor(max_leaf_nodes = 4)
regtree = regtree.fit(x_train, y_train)
fig = plt.figure(figsize = (20, 12))
tree.plot_tree(regtree, filled = True, feature_names = x.columns, fontsize = 15)
plt.show()
```

Logistic Regression

```
[[81829   275]
 [ 8916  5448]]
              precision    recall  f1-score   support

           0       0.90      1.00      0.95     82104
           1       0.95      0.38      0.54     14364

    accuracy                           0.90     96468
   macro avg       0.93      0.69      0.74     96468
weighted avg       0.91      0.90      0.89     96468
```



Method and Interpretation

- Because we are looking to standardize across *when* people opened their account, a logistic regression was the first way to start.
- After fitting the logistic regression model, we see that the accuracy of logistic regression classifier on test set is 0.90
- (ROC) curve is a common tool used with binary classifiers. The dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).

```python
# StandardScaler() for PCA and regression and any methods using gradient descent
# MinMaxScaler() for images (pixel intensities in a scale of 0-255)

model_df_scaled = model_df.copy()
model_df_feature_cols = model_df.columns.to_list()
model_df_feature_cols.remove('closed_HH')
model_df_scaled[model_df_feature_cols] = StandardScaler().fit_transform(
    model_df_scaled[model_df_feature_cols])
```

```python
model = LogisticRegression()

model.fit(x_train, y_train)

y_pred = model.predict(x_test)
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import classification_report
print(classification_report(y_test,y_pred))
```

Gradient Boosting

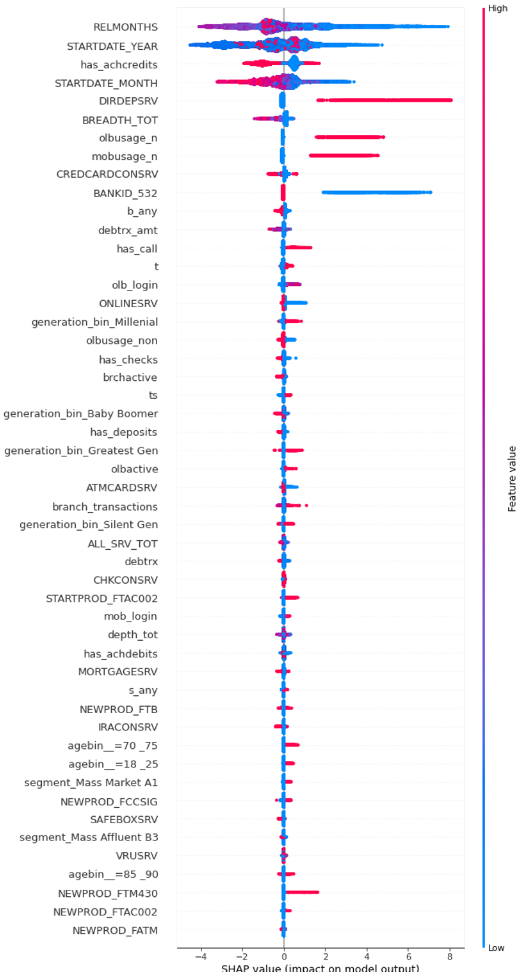
Extreme Gradient Boosting

- Speed and performance on top of traditional Random Forests
- Can handle sparse data (good for our sparse dataset for our one-hot encoded categorical features)
- Boosting allows weak learners to learn from past mistakes into a strong learner

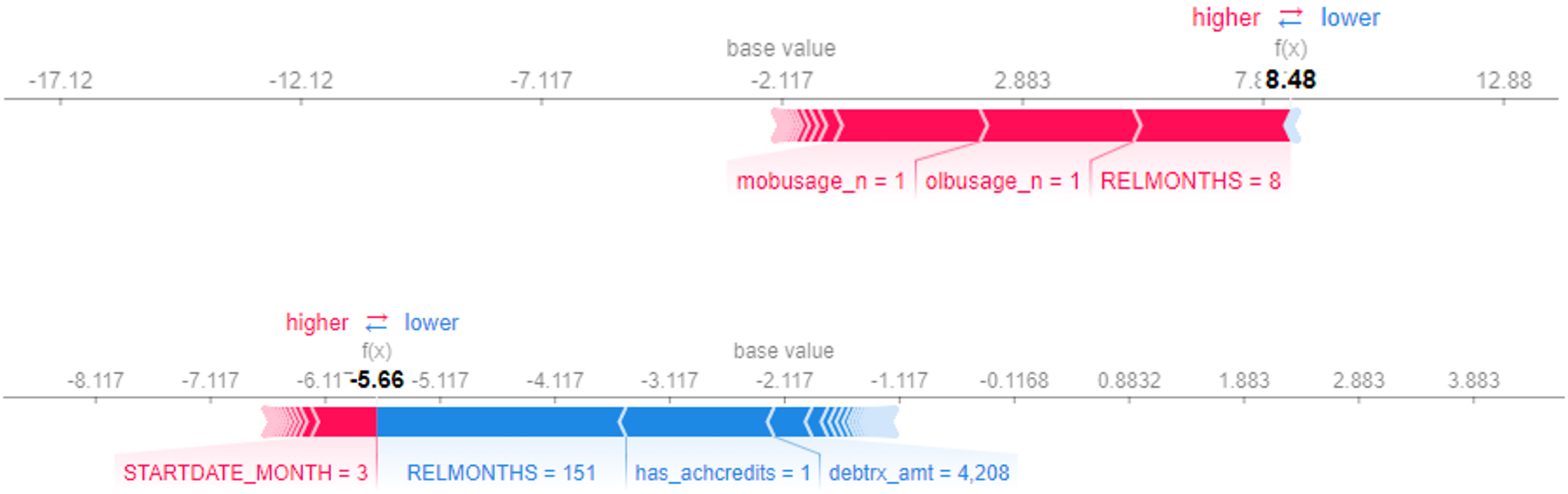
```
F1 score and accuracy score for training set: 0.9918 , 0.9859.
F1 score and accuracy score for test set: 0.9908 , 0.9843.
              precision    recall  f1-score   support

           0       0.98      1.00      0.99     82050
           1       1.00      0.89      0.94     14418

    accuracy                           0.98     96468
   macro avg       0.99      0.95      0.97     96468
weighted avg       0.98      0.98      0.98     96468
```

Results

- Feature attribution should have **consistency and accuracy**
- By plotting the impact of a feature on every sample we can also see important outlier effects. For example, while DIRDEPSRV(Binary field showing presence (1) or absence (0) of direct deposit service in the household) is not the most important feature globally, it is by far the most important feature for a subset of customers
- **Most product related features were not impact in our model, but the top ones included: STARTPROD_FTAC002 (Simply Checking) and NEWPROD FCCSIG (Signature Credit Card)**



Gradient Boosting Feature Attribution



Agenda

Member Introduction & Projection Description

Data Processing and Cleaning

Classification Methods

- Decision Tree Analysis
- Logistic Regression
- XGBoost

Modeling Time to Churn

Takeaways & Conclusion

sBG Methodology (credit to Peter Fader)

Churn Model

- The sBG model ultimately answers questions relating to customer retention and is applied to a churn dataset.
- Individual customer behavior story: at the end of each period, a customer renews his contract with probability **1 - θ (geometric distribution)**
- Churn propensities (**θ**) varies across customers based on observable and unobservable characteristics
- To model unobserved heterogeneity, assume:

  θ ~ Beta(α, β)

Continuous Mixture Model: Shifted-Beta-Geometric (sBG) Distribution

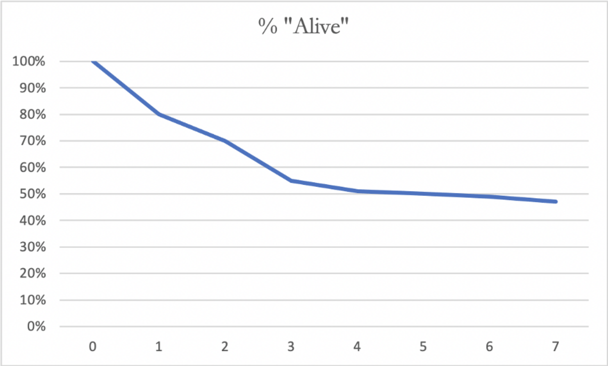
$$P(T = t \mid \alpha, \beta) = \int_0^1 P(T = t \mid \theta) f(\theta \mid \alpha, \beta)\, d\theta$$
$$= \frac{B(\alpha + 1, \beta + t - 1)}{B(\alpha, \beta)}.$$

$$P(T > t \mid \alpha, \beta) = \int_0^1 P(T > t \mid \theta) f(\theta \mid \alpha, \beta)\, d\theta$$
$$= \frac{B(\alpha, \beta + t)}{B(\alpha, \beta)}.$$

 # Customers Surviving At Least 0-7 Years

Year	# Customers	% "Alive"
0	1000	100%
1	800	80%
2	700	70%
3	550	55%
4	510	51%
5	500	50%
6	490	49%
7	471	47%

Churn Dataset Example



sBG Methodology (credit to Peter Fader)

Methodology

1. Construct churn datasets for each account start date, using the feature relmonths
2. Identify customers within the two identified segments from the decision tree classification
3. For the segment of customers most likely to churn, apply an sBG model to understand probabilities of churn in the months following account opening
4. Interpret the Alpha Beta hyperparameters to understand when current customers will churn

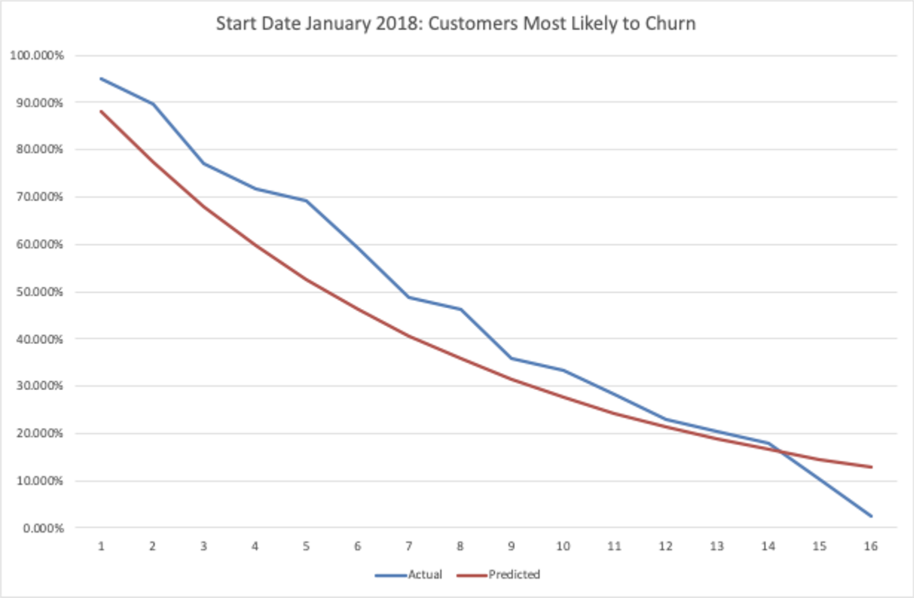
	RELMONTHS	Count	% Survived
0	10	4	0.98473282
1	11	1	0.98091603
2	12	2	0.97328244
3	13	4	0.95801527
4	14	7	0.93129771
5	15	3	0.91984733
6	16	3	0.90839695
7	17	2	0.90076336
8	19	1	0.89694656
9	20	2	0.88931298
10	22	2	0.88167939
11	23	2	0.8740458
12	24	1	0.87022901
13	26	1	0.86641221
14	28	4	0.85114504
15	29	2	0.84351145
16	30	2	0.83587786
17	32	219	0.83587786

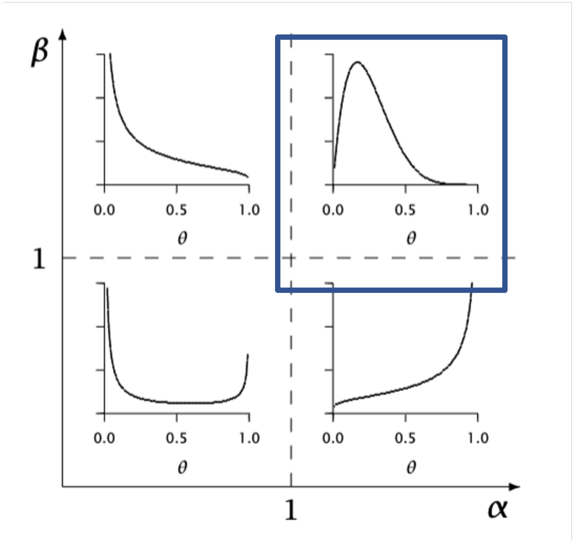
Churn Dataset for Accounts Started March 2018

sBG Model

Takeaways

- Given the distribution of theta, which represents propensity to churn, we see that there is very little *unobserved heterogeneity* amongst the of customers in the most likely to churn segment
- This implies that most of the customers who are most likely to churn have similar motivations for opening/closing accounts





alpha	25454410.88
beta	185360316.1

Average Theta: 12%

Summary

Gradient Boost	• On a randomly sampled test set, our XGBoost model could predict whether they churn (closed_HH=1) with 98% accuracy • Additionally, we identified the feature importance of features such as RELMONTHS and product features such as the presence of the Signature Credit Card. We also found that the higher their Mobile Usage, Online Usage the less the likelihood of churn • Generally, customers with smaller months with Fulton and do not have a Signature Credit Card are more likely to churn.
Decision Tree	• For the decision-tree analysis, we refiltered the data by dropping columns that have more than 300,000 null or missing values • We also dropped columns with significant collinearity that would impact the accuracy of our model, and dropped any rows with NA • A lack of checking account or direct deposit is a clear warning that a customer is likely to churn
Logistic Regression	• Because we are looking to standardize across *when* people opened their account, a logistic regression was the first way to start. • After fitting the logistic regression model, we see that the accuracy of logistic regression classifier on test set is 0.90 • (ROC) curve is a common tool used with binary classifiers. The dotted line represents the ROC curve of a purely random classifier; a good classifier stays as far away from that line as possible (toward the top-left corner).
sBG Model	• From the sBG methodology, we gain a deeper understanding of the segment of customers who are most likely to churn and can calculate approximate probabilities of churn at each time period following account opening • We expect all customers classified as least loyal to churn after ~2 years
Suggestions	• Improve mobile app interface and website, convince customers to join mobile app as part of marketing • Product-wise, the signature credit card program has the best performance in reducing churn, Fulton could have more similar programs to increase customer loyalty to reduce churn • Overall, be wary of customer churn in the initial two years because eventually customers will become loyal and generate high value

Q&A