Logistic Regression and ANN Solutions for Churn Rate

Contributors: Jason Spaw

Objective:

- Assess the potential losses due to churn.
- Provide solutions for churn rate.
- Observe potential modeling for predicting churn of customers in a given time period.
  - Logistic Regression
  - ANN

What is churn rate, and why is this important?

Churn Rate: The percentage of customers or subscribers that stop doing business with a company over a specific period of time.

Importance:

- Banks make their money off loans and other services provided to existing customers.
- Free checking accounts act as an initial relationship between banks and customers.
- Losing customers means losing from your pool of sales, or potential leads.

Other Key Terms

Net Churn Rate: The measure of lost revenue month over month,due to cancellations and account downgrades, after factoring in revenue from existing customers.

Expansion Revenue: Any revenue that is generated in excess from customer's initial purchasing price or contact.

Churned Revenue: Measure the monthly recurring revenue (MRR) loss from existing customers over a specific period.

ARPA: Average Revenue Per Account (assuming stock market annual returns of 10% for this example)

Net Churn Rate

Expansion Revenue = Assume no excess profits
Churned Revenue = (Capital in active churned accounts) * (Average rate of return (10%))
Revenue, Beginning of Period = (Capital in active accounts) * (Average rate of return

$$\text{Net Churn Rate} = \frac{(\text{Expansion Revenue} - \text{Churned Revenue})}{\text{Revenue, Beginning of Period}}$$

Your Estimated Net Churn Rate

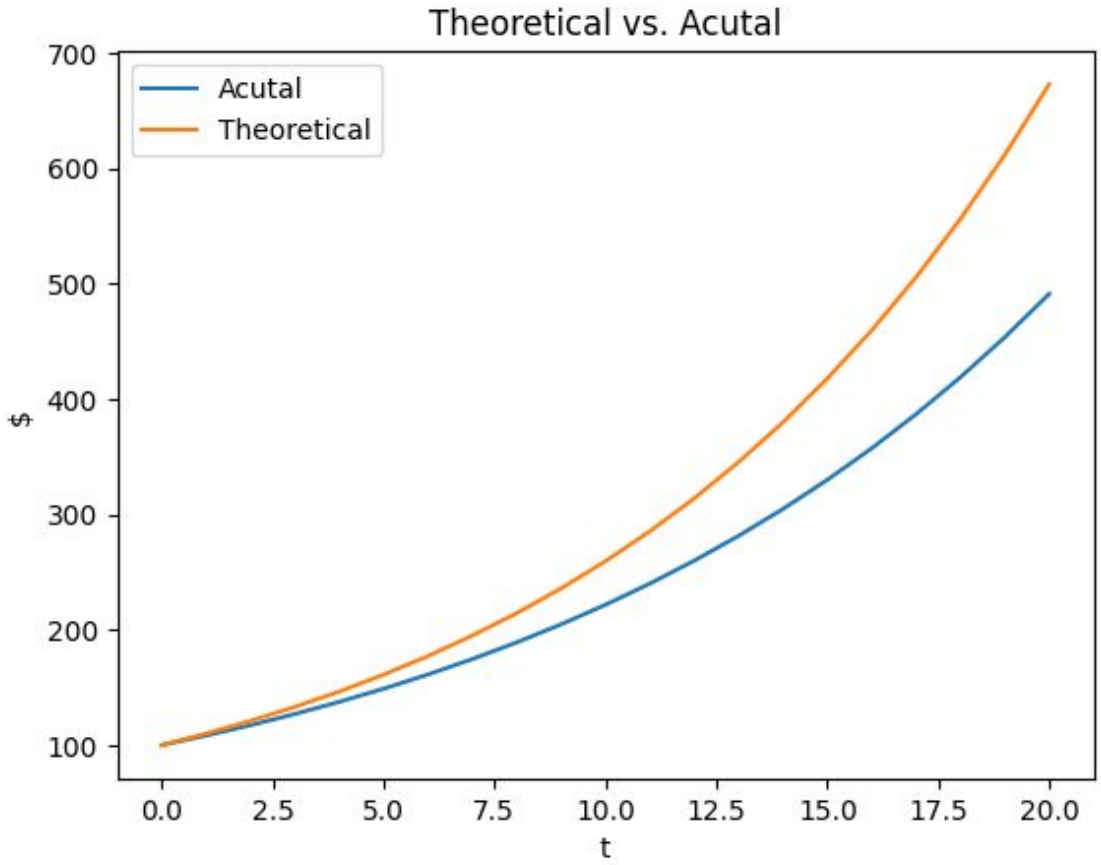
```
# Calculate Churn Rate for this month

active_accounts = churn_data2[churn_data2['IsActiveMember'] == 1]
exited_accounts = active_accounts[active_accounts['Exited'] == 1]

churn_rate = (0 - (np.sum(exited_accounts['Balance'])*0.1))/(np.sum(active_accounts['Balance'])*0.1)
actual_rate = ((np.sum(active_accounts['Balance'])*0.1) - (np.sum(exited_accounts['Balance'])*0.1))/(np.sum(active_accounts['Balance'])*0.1)
print(f"Churn Rate: {churn_rate}")
print(f"Actual Rate: {actual_rate}")
```

```
Churn Rate: -0.17173689182607707
Actual Rate: 0.828263108173923
```

Theoretical vs. Actual Growth Over Time



Other Products Missed Due to Churn

- Loans
- Monthly Maintenance Fees
- Overdraft Fees
- Insufficient Funds Fees
- Out-of-network ATM Fees

Our Solution

- Provide you with a model that effectively predicts churn rate for the given month.
- This provides target customers
- Bank may provide incentives to target customers to keep them.
- Incentives may include lower interest rates
- This enables you to keep legacy customers on higher interest rates while only targeting those at risk of churning to provide lower rates for retention.
- Optimize profits

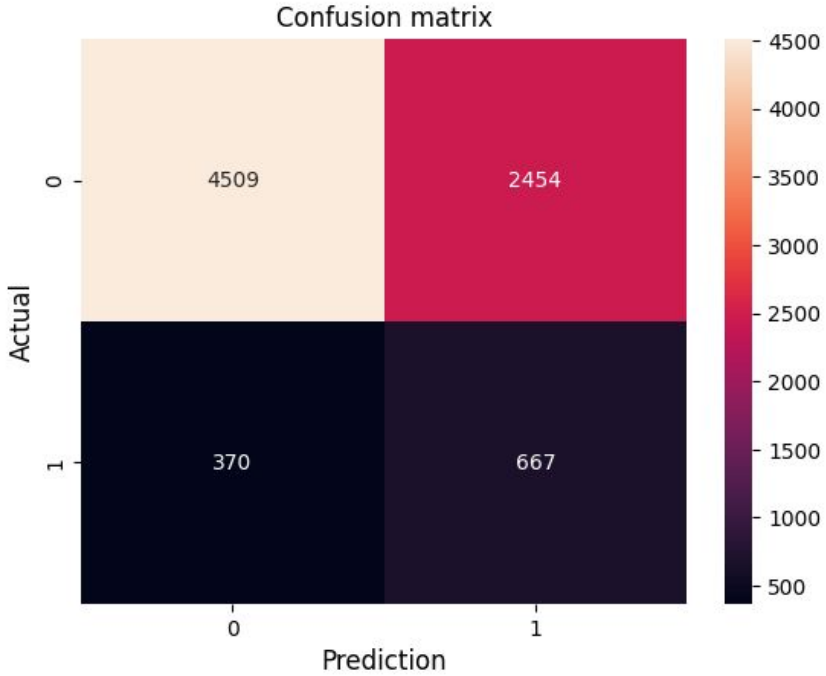
Cleaning and Feature Engineering

- No NaN’s present
- Gender column was converted to binary (Male = 1, Female = 0)
- Geography was split into dummy columns reflecting each individual country
- Data was normalized

Model Performance

Logistic Regression

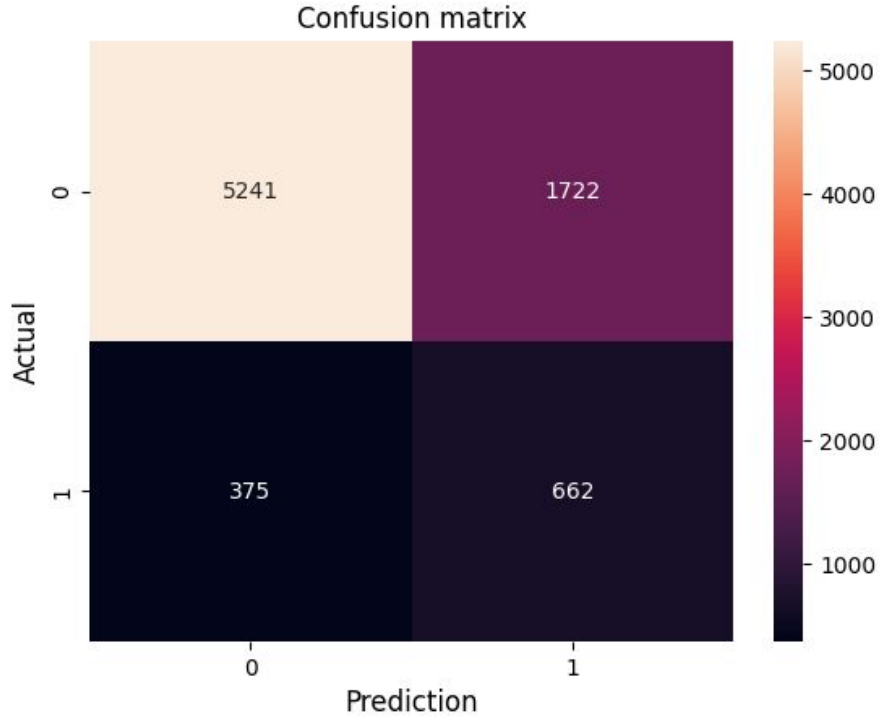
- Accuracy: 0.6470
- Precision = TP/(TP + FP) = 0.214
- Recall = TP/(TP + FN) = 0.643



Model Performance

Shallow ANN (neurons = 14)

- Accuracy: 0.738
- Precision = TP/(TP + FP) = 0.278
- Recall = TP/(TP + FN) = 0.638



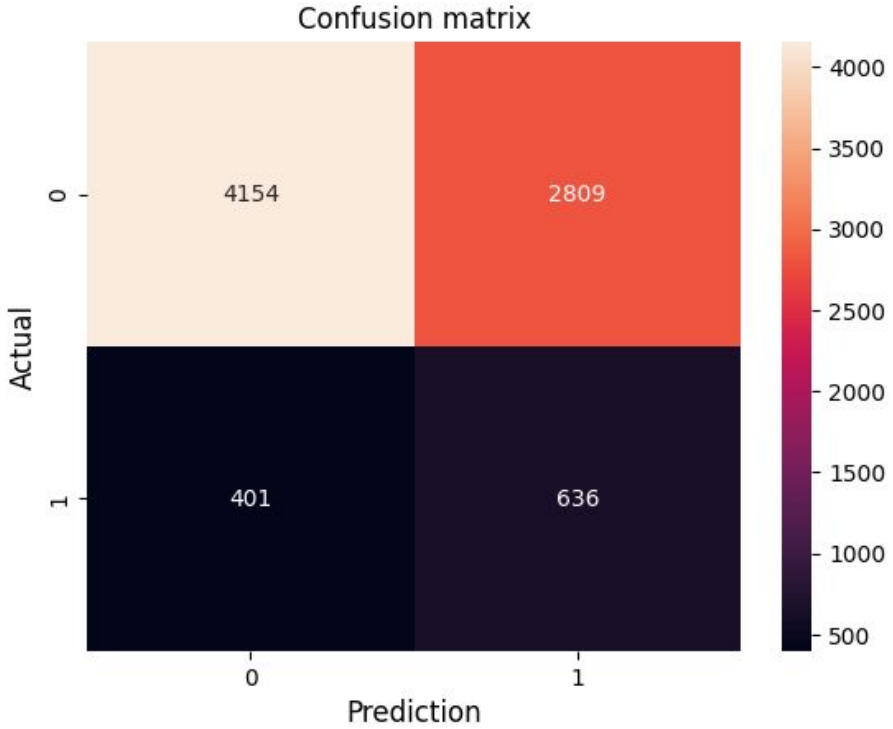
Model Performance

ANN Classifier

architecture=[20,16,14],

activations=[sigmoid, np.tanh, sigmoid]

- Accuracy: 0.599
- Precision = TP/(TP + FP) = 0.185
- Recall = TP/(TP + FN) = 0.613



Conclusion

- Churning of customers accounts for noticeable losses in the company
- Solutions:
  - Churn Predictor
  - Incentives (possibly lower interest rates)
  - Will optimize the retention efforts by minimizing resources (profits) used to retain
- All models performed similarly but Shallow ANN had better precision and recall, with recall being our greatest concern.

Questions?

Thank You