AWS Build-A-Thon

Telecom Customer Churn Prediction Powered By AWS Sagemaker

-Deepthi V N

Github
(https://github.com/SmartPracticeschool/SPS-1610-Telecom-Customer-Churn-Prediction-powered-by-AWS-Sagemaker)

Video
https://youtu.be/BnfjPWFROFk

Objective

Customer churn occurs when customers stop doing business with a company. As the cost of retaining an existing customer is far less than acquiring a new one, maintaining a healthy customer base is important for the success of any business. As customers have multiple options in the telecom industry, the churn rate is particularly high in this industry. Individualized customer retention is difficult because businesses usually have a lot of customers and cannot afford to spend much time on one. The costs would be too high and would outweigh the extra revenue. But if a company could predict in advance which customers are at risk of leaving, they could focus customer retention efforts by directing them solely toward such "high risk" customers.

The main objective of my project was to build a predictive model to generate a prioritized list of customers most vulnerable to churn.

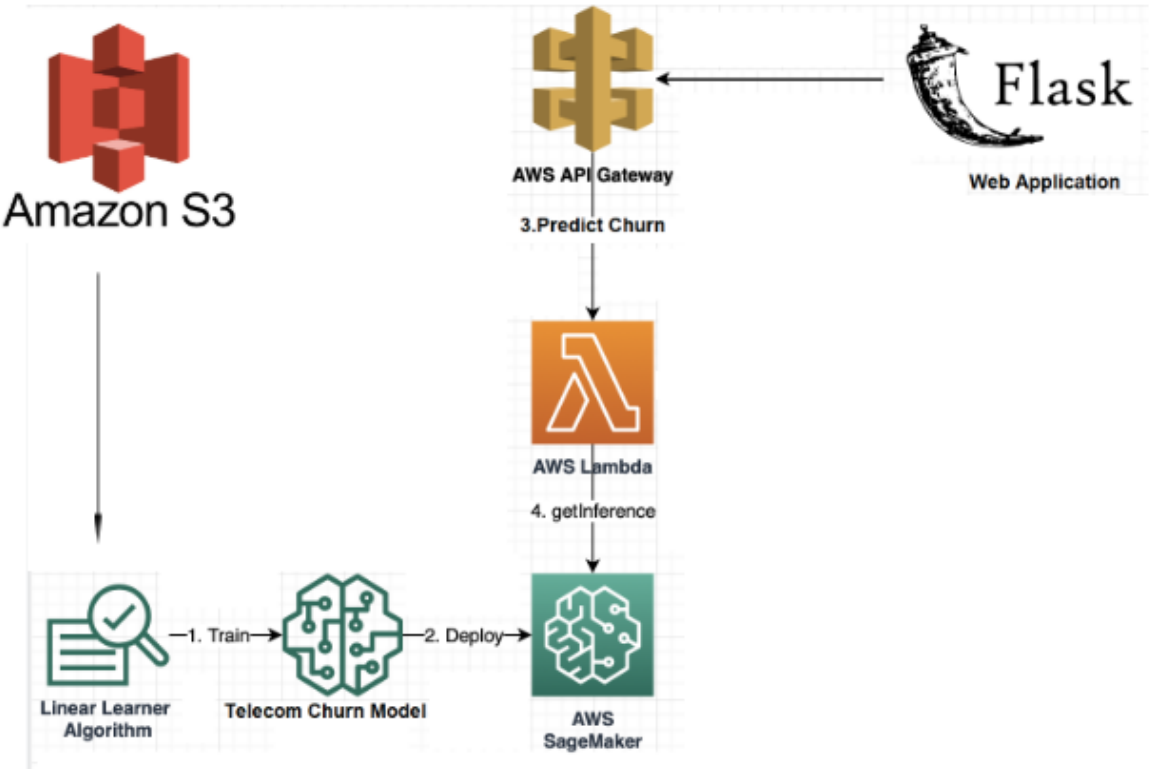
Method

I used the bank dataset available from kaggle.com that contains details of the bank customers and the target variable reflecting the fact whether the customer left the bank or not. The raw data contained 10000 rows (customers) and 14 columns (features). The Exited column was target variable. The classification goal was to predict whether the customer will churn(1/0).

Predict variable (desired target): y — has the customer churned? (binary: "1", means "Yes", "0" means "No)

I cleaned the dataset to remove the unnecessary attributes and hot encoded several categorical variables. I explored the correlation between several features and the target variable. I included all features to predict the target variable (customer churn).

Then I have built and deployed a Machine Learning model to predict the customer churn using Amazon SageMaker and predictions can be obtained by using its Endpoint.
A python - flask application is also created that interacts with the model deployed on AWS Sagemaker with the help of AWS API Gateway and AWS Lambda Services.



Model

Amazon SageMaker provides an XGBoost container that we can use to train in a managed, distributed setting, and then host as a real-time prediction endpoint. XGBoost uses gradient boosted trees which naturally account for non-linear relationships between features and the target variable, as well as accommodating complex interactions between features.

Amazon SageMaker XGBoost can train on data in either a CSV or LibSVM format. Also it should

- Have the predictor variable in the first column
- Not have a header row

Using this technique I have got Overall Classification Rate(accuracy) of 84.4%

Cost Evaluation

I was interested in exploring the cost implications of implementing, vs. not implementing a predictive model. There were costs associated with the model erroneously assigning false positives and false negatives. It was important to look at similar costs associated with correct predictions of true positives and true negatives. Because the choice of the threshold affects all four of these statistics, it was important to consider the relative costs to the business for each of these four outcomes for each prediction.

I made the following cost assumptions to explore the cost implications of implementing the model.

1. My model essentially correctly identified a happy customer in this case, and in this case, we don't need to do anything.
2. False negatives were the most problematic, because they incorrectly predict that a churning customer will stay. We will lose the customer and will have to pay all the costs of acquiring a replacement customer, including foregone revenue, advertising costs, administrative costs, etc.
3. Finally, for customers that my model identifies as churning, this is the cost of

both true positive and false positive outcomes. In the case of false positives (the customer is happy, but the model mistakenly predicted churn).

It’s clear that false negatives are substantially more costly than false positives.

Conclusion

In order to maintain their current customer base, using the current model will save the financial status of the bank. So, it’s worth investing in optimizing the model further to increase cost savings.

Future Work

One can optimize the sensitivity of the model (further decrease the number of false negatives). Also one can invest more in feature engineering and try and include additional features from other datasets and make it more generalized.