BCG PowerCo. Churn Analysis

1st Parkash Meghwar
MS Data Science
Stevens Institute of Technology
Jersey City, USA
pparkash@stevens.edu

2nd Asad Sajid Awan
MS Business Intelligence and Analytics
Stevens Institute of Technology
Jersey City, USA
aawan2@stevens.edu

***Abstract*—The objective of this project is to help PowerCo., a gas and electricity distribution company, anticipate customer churn that would ultimately help them save those customers. After analyzing the data and applying around 7 machine learning algorithms, we achieved an accuracy of 90% and a recall of 76% on the test data. This indicates our models' ability to predict customer churn and offers an opportunity to explore a discounted rate of 20% as a potential solution.**

I. Introduction

The objective of this project is to anticipate the likelihood of customer churn for PowerCo., one of BCG's clients operating in the gas and electricity distribution sector for residential neighborhoods and SMEs. With the power sector undergoing liberalization, companies are under pressure to reduce their costs to stay competitive. PowerCo. is seeking a new, feasible, and efficient pricing strategy to combat customer churn. Our task involves predicting which customers are likely to churn so PowerCo. can take some measures to retain them. PowerCo. is interested in testing the hypothesis that offering a 20% discounted rate could prevent customer churn. However, we can also explore alternative solutions to address this issue and provide additional recommendations to the client.

After completing exploratory data analysis and normalizing the data, we have used Logistic Regression, K-Nearest Neighbors, Support Vector Machines, Decision Tree, Gaussian Naïve Bayes (GNB), Gradient Boosting Classifier (GBC), and Random Forest algorithms to predict which customers would churn, achieving the accuracy of 90% and recall of 76% on testing data.

II. Related Work

All the methods that we have used are famous for classification problems, such as this one of customer churn analysis, and have been used extensively in the past. This dataset, however, is a brand-new dataset and has just been made available at Kaggle so people are yet to work on it and publish their findings. Initially, we were thinking of going ahead with logistic regression, SVM and KNNs, but our primary research revealed that, for the problems that suffer from imbalance classes, decision trees, random forest and gaussian naive bayes have proven to give accurate results for classification tasks.

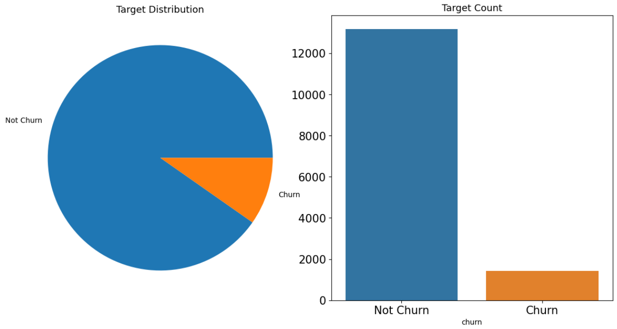
III. Our Solution

A. Description of Dataset

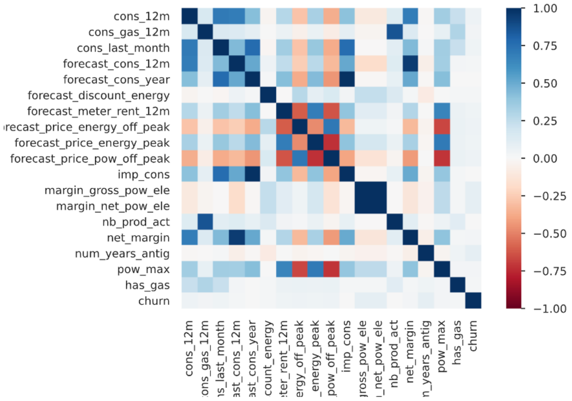
Customer data such as usage, sign up date, forecasted usage etc. ID is the unique identifier of each client in the dataset. Churn indicator: whether each customer has churned or not. It has 14000+ rows and 26 variables.

We have used pandas profiling library to do EDA of the given dataset. Target value, Churn, is highly skewed therefore the appropriate metric the judge the performance of the classifiers should be precision and recall. From the correlation matrix, it seems that only gross margin and net margin variables are correlated with the target variable.

Target value Distribution:



Correlation Matrix:



Normalization of Features: While most of the machine learning algorithms that we have used do not suffer from non-normalized features, KNNs and Logistic regression require that we use this pre-processing technique to bring all the variables on the same scale. We used ‘StandardScaler‘ command from ‘sciket-learn‘ library so that every variable has a mean of 0 and standard deviation of 1. This is also referred to as z-score normalization.

Appropriate Data Types: Nearly all the variables had appropriate data types except ‘hasgas‘ column with values ‘t‘ for true and ‘f‘ for false. We converted it into Boolean.

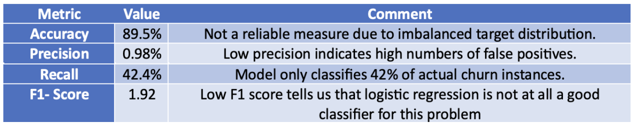
Train-Test Split: We used random sampling to divide the dataset into training and testing based on 80-20 rule.

B. Machine Learning Algorithms

Our approach was to implement as many ML algorithms as we could that we have studied in this class and compare their performance as a classifier. For this purpose, we chose some of the algorithms that have already been used on this dataset and some of them, which we discovered in our primary research to solve the challenges we faced during implementation phase. All the results shared are based on Test dataset.

C. Implementation Details

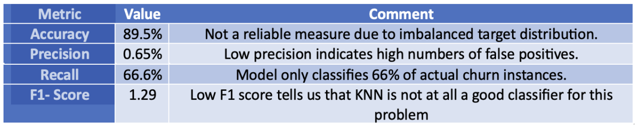
Logistic Regression:



Metric	Value	Comment
Accuracy	89.5%	Not a reliable measure due to imbalanced target distribution.
Precision	0.98%	Low precision indicates high numbers of false positives.
Recall	42.4%	Model only classifies 42% of actual churn instances.
F1- Score	1.92	Low F1 score tells us that logistic regression is not at all a good classifier for this problem

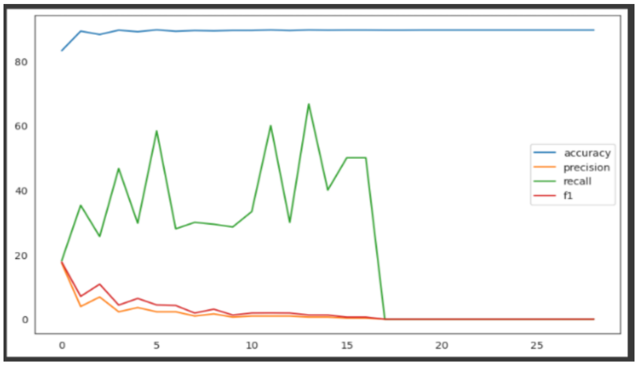
Challenges: Logistic regression was fairly simple to implement on this dataset, but it did not yield satisfactory results even after normalization of the features. Since logistic regression gets its inspiration from linear regression, we assume this is because of the absence of the highly correlated variables with the target variables.

K-Nearest Neighbor: N = 14

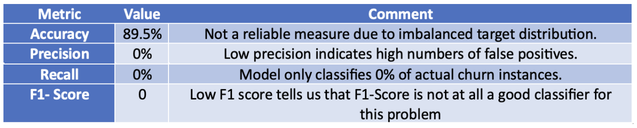


Metric	Value	Comment
Accuracy	89.5%	Not a reliable measure due to imbalanced target distribution.
Precision	0.65%	Low precision indicates high numbers of false positives.
Recall	66.6%	Model only classifies 66% of actual churn instances.
F1- Score	1.29	Low F1 score tells us that KNN is not at all a good classifier for this problem

Challenges: In problems involving KNN, it is often key to select the number of neighbors that give us the most optimal results. We obtained different values based on different values of N showed below.



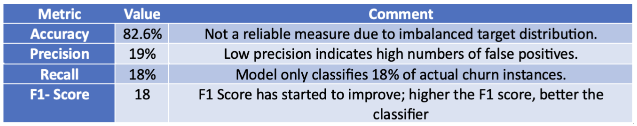
Support Vector Machines (SVM):



Metric	Value	Comment
Accuracy	89.5%	Not a reliable measure due to imbalanced target distribution.
Precision	0%	Low precision indicates high numbers of false positives.
Recall	0%	Model only classifies 0% of actual churn instances.
F1- Score	0	Low F1 score tells us that F1-Score is not at all a good classifier for this problem

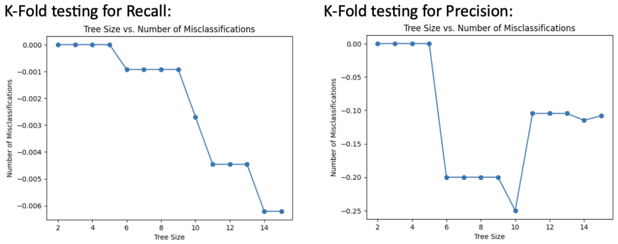
Challenges:SVM does not work on this data at all. One of the reasons for that is high target class imbalance between churn and not churn clients. To battle the high class imbalance issue, we have tried decision trees and random forest ML algorithms below.

Decision Tree:

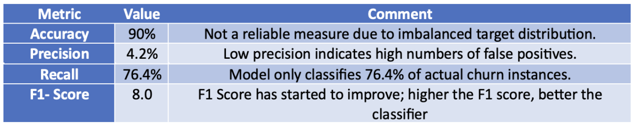


Metric	Value	Comment
Accuracy	82.6%	Not a reliable measure due to imbalanced target distribution.
Precision	19%	Low precision indicates high numbers of false positives.
Recall	18%	Model only classifies 18% of actual churn instances.
F1- Score	18	F1 Score has started to improve; higher the F1 score, better the classifier

Challenges: Since Decision Trees are not susceptible to high class imbalance, we have started to get a better precision, recall and F1-score.



Random Forest:

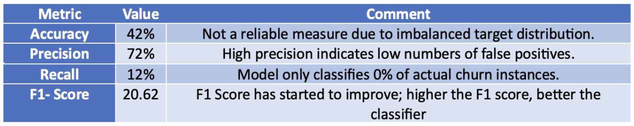


Metric	Value	Comment
Accuracy	90%	Not a reliable measure due to imbalanced target distribution.
Precision	4.2%	Low precision indicates high numbers of false positives.
Recall	76.4%	Model only classifies 76.4% of actual churn instances.
F1- Score	8.0	F1 Score has started to improve; higher the F1 score, better the classifier

Challenges: Since we know that it is a tradeoff between Precision and Recall. We can change the hyperparameters of this model to increase precision, but it will decrease recall. In the next model of GNB, you will see that we have increased precision, but recall has dropped. If we prefer Recall over

Precision, Random Forest seems to be the best model for this problem.

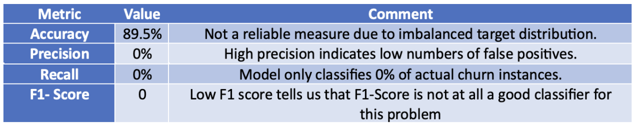
Gaussian Naive Bayes (GNB):



Metric	Value	Comment
Accuracy	42%	Not a reliable measure due to imbalanced target distribution.
Precision	72%	High precision indicates low numbers of false positives.
Recall	12%	Model only classifies 0% of actual churn instances.
F1- Score	20.62	F1 Score has started to improve; higher the F1 score, better the classifier

Challenges: GNB is a probabilistic classifier based on Bayes' theorem. It assumes that features are conditionally independent given the class label and follows a Gaussian (normal) distribution. For this model, we have high precision, but low recall. And this model gives us the highest F1-score as well. It is a strong contender to be the best model if we prefer precision over recall.

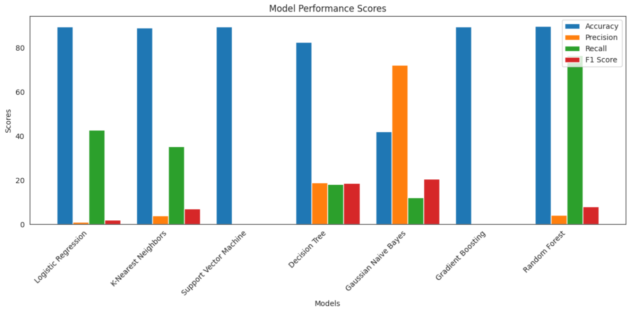
Gradient Boosting Classifier (GBC):



Metric	Value	Comment
Accuracy	89.5%	Not a reliable measure due to imbalanced target distribution.
Precision	0%	High precision indicates low numbers of false positives.
Recall	0%	Model only classifies 0% of actual churn instances.
F1- Score	0	Low F1 score tells us that F1-Score is not at all a good classifier for this problem

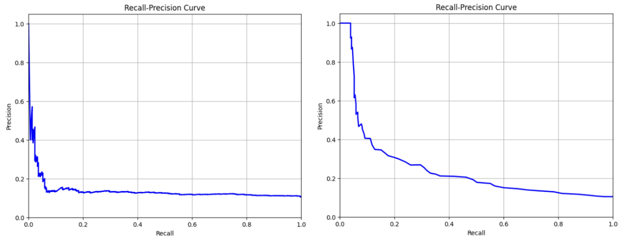
Challenges: We tested GBC out of curiosity to see how XGBoost variants perform on our dataset. It didn't give us any promising results.

IV. Comparison

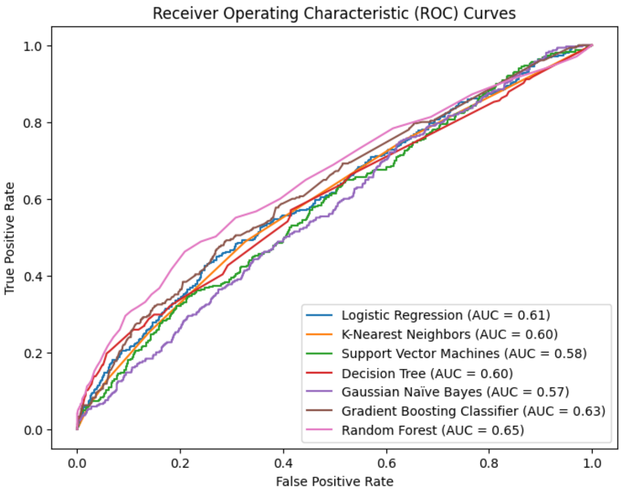


If we prefer recall (proportion of churned clients that the model correctly identifies), Random Forest seems to be the best performing. If we prefer precision (proportion of predicted churned clients who are actually churned), GNB seems to be the best classifier for this problem. Decision Tree seems to be an average model, but certainly better than logistic regression and KNN, who suffer from high class imbalance in the data. Actually, the reason decision tree performs slightly better, and random forest and GNB performs much better, is because these models are not much affected by the high class imbalance problem we faced in this dataset.

Recall-Precision Curve for GNB:　Recall-Precision Curve for Random Forest:



ROC Curves:



V. Future Directions

Feature Engineering: In our on-the-job training, we used to study the variables of the models extensively and do feature engineering to come up with variables which had high predictive power for the target class variable. This often involved tracking the account of customer for over a period of one year or so to see how the importance of the customer changed overtime. The features given to us in the dataset can be studied and monitored throughout a period by maintain slowly changing dimensions. This is one potential future direction that can be fruitful for our task.

Class Imbalance: For this problem, instead of solving the problem of data imbalance by repeatedly sampling the class which is low in number, we worked with ML models that are not much susceptible to this problem. We can approach the problem the other way by pseudo sampling the disadvantaged class to increase its proportion.

Neural Networks: We were introduced to neural networks in this course and were fascinated by their power for prediction and classification. Where traditional ML techniques underperform, neural networks give the best results. But they need extensive exploration and study at our part to be comfortable enough and to apply those cutting-edge techniques to solve this problem.

VI. Conclusion

We implemented around 7 ML models in hopes to counter the challenges we were facing in the problem and to improve the predictive power. We conclude that Random Forest, with the highest ROC, is our shot in solving this problem. Based on the Recall-Precision graph, it offers the best tradeoff between recall and precision. An argument can be made for Gaussian Naïve Bayes (GNB) as well because it has high F1 score, and it offers slightly better precision on the go as compared to random forest. With these models, we can identify with 76% recall, alternatively, with 72% precision which customers are likely to leave PowerCo. so that the company can take extensive measures to retain them.

We believe we have some room for improvement especially when we haven't experimented with neural networks to solve the problem.

References

Dataset : https://drive.google.com/drive/folders/1a2rtXe7-t5ijh64hcwx2Ic31r4-dcjj1?usp=sharing

Sheth, V. (2022, February 18). Implementing gradient descent in python from scratch. Medium. https://towardsdatascience.com/implementing-gradient-descent-in-python-from-scratch-760a8556c31f

Sharma, P. (2023) Implementation of gaussian naive Bayes in python sklearn, Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/11/implementation-of-gaussian-naive-bayes-in-python-sklearn/ (Accessed: 08 July 2023).

Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. (2019, September 1). O'Reilly Online Learning. https://www.oreilly.com/library/view/hands-on-machine-learning/9781492032632/