CS 6375
Machine Learning - F24

Exploratory Group Project

Customer Churn Prediction

By Model Masters

Team Members

Shiva Dhanush Konuru(SXK230027)
Manikanta Sai Kommireddy(MXK220132)
Mayuresh Bhangale(MXB240017)
Chetan Rahane(CRR220000)

Instructed By
Prof. Crystal Maung

Customer Churn Prediction in the Telecom Industry

1. Introduction

The predictions have now become a cornerstone in customer retention strategies across industries. Churn is a scenario wherein a **customer discontinues business** dealings with a company. The case of telecom companies is different since the competition is high, and getting new customers often costs more than retaining existing ones.

The aim of this project is to bring customer churn prediction into life, develop actionable insights, and help in strategizing ways to improve retention. We decided to use the **Telco Customer Churn** dataset available on Kaggle by IBM because of its rich structure and relevance to the telecom domain.

2. Explanation

2.1 Why We Chose Customer Churn?

Customer churn prediction is critical for reducing revenue losses, especially in subscription-based businesses. By identifying at-risk customers, companies can take targeted actions to retain them, boosting profitability and customer satisfaction.

2.2 Why the Telecom Industry

It faces one of the highest churn rates in the industry, with aggressive competition and analogous service offerings. Added to this is the cost sensitivity of customers, making it an ideal use case for predictive analytics and strategies to prevent churn.

We explored multiple datasets available for telecom industry customer churn and decided to go ahead with Telco by IBM.

2.3 Why the Telco Dataset from IBM Kaggle

The **Telco Customer Churn dataset** was chosen for its relevance and completeness:

- It contains a mix of numerical, categorical, and binary features that allow us to explore a variety of machine learning models.
- The dataset includes a well-defined target variable, making it suitable for binary classification tasks.

- It represents real-world customer data in a telecom context, allowing insights to be directly applicable.

The dataset had 7041 customer data, and 21 columns which could be used as features. The columns are as follows: customerID, gender, SeniorCitizen, Partner, Dependents, tenure, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, MonthlyCharges, TotalCharges, Churn.

It contained binary columns like SeniorCitizen, Partner, and PhoneService; numeric columns including tenure, monthly charges, and total charges; and also multi-valued categorical columns such as the type of contract.

2.4 Preprocessing and Analysis of the Dataset

To ensure the dataset was ready for modeling, we performed the following steps:

Since, the column customerID would not provide insight on why the customer decided to leave, this column needed to be dropped, so that the training could be faster.

The categorical data needed to be handled. We used label encoding and one-hot encoding to convert categorical features into numerical formats.

We scaled the numerical columns, so that it would fit in a scale which could be applied to all the data. Applied standardization to ensure features like MonthlyCharges and TotalCharges were on comparable scales.

The rows that contained empty values and incomplete data were dropped.

The data contained 5174 non churn customers and 1869 churn customers data. Now, we were ready to start with the training.
Following is the details of dataset:

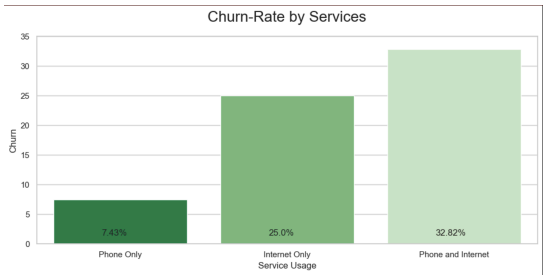
```
onlinesecurity
 onlinesecurity
No                     3497
Yes                    2015
No internet service    1520
Name: count, dtype: int64


onlinebackup
 onlinebackup
No                     3087
Yes                    2425
No internet service    1520
Name: count, dtype: int64


deviceprotection
 deviceprotection
No                     3094
Yes                    2418
No internet service    1520
Name: count, dtype: int64
```

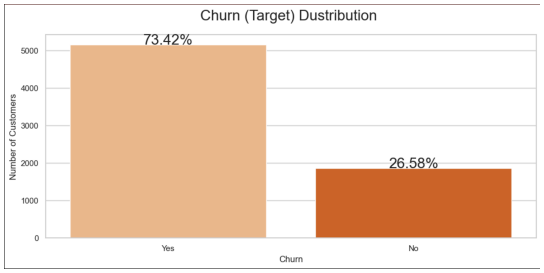
```
deviceprotection
 deviceprotection
No                     3094
Yes                    2418
No internet service    1520
Name: count, dtype: int64


techsupport
 techsupport
No                     3472
Yes                    2040
No internet service    1520
Name: count, dtype: int64


streamingtv
 streamingtv
No                     2809
Yes                    2703
No internet service    1520
Name: count, dtype: int64


streamingmovies
 streamingmovies
No                     2781
Yes                    2731
No internet service    1520
Name: count, dtype: int64
```

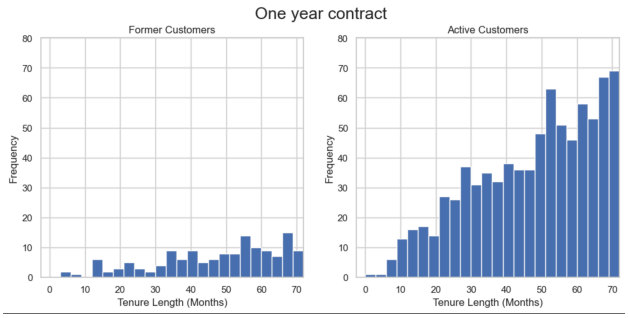
Table 1-1

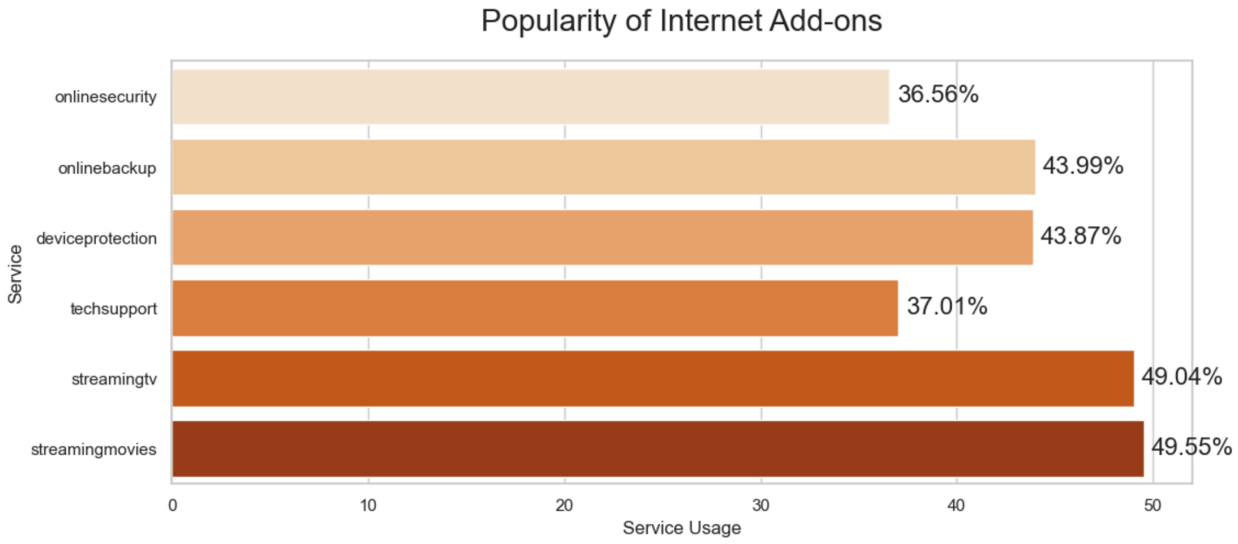
	Name	dtypes	Uniques	Row 1	Row 2	Last Row
0	customerID	object	7043	7590-VHVEG	5575-GNVDE	3186-AJIEK
1	gender	object	2	Female	Male	Male
2	SeniorCitizen	int64	2	0	0	0
3	Partner	object	2	Yes	No	No
4	Dependents	object	2	No	No	No
5	tenure	int64	73	1	34	66
6	PhoneService	object	2	No	Yes	Yes
7	MultipleLines	object	3	No phone service	No	No
8	InternetService	object	3	DSL	DSL	Fiber optic
9	OnlineSecurity	object	3	No	Yes	Yes
10	OnlineBackup	object	3	Yes	No	No
11	DeviceProtection	object	3	No	Yes	Yes
12	TechSupport	object	3	No	No	Yes
13	StreamingTV	object	3	No	No	Yes
14	StreamingMovies	object	3	No	No	Yes
15	Contract	object	3	Month-to-month	One year	Two year
16	PaperlessBilling	object	2	Yes	No	Yes
17	PaymentMethod	object	4	Electronic check	Mailed check	Bank transfer (automatic)
18	MonthlyCharges	float64	1585	29.85	56.95	105.65
19	TotalCharges	object	6531	29.85	1889.5	6844.5
20	Churn	object	2	No	No	No

Here are some of the plots which showcase the variety of the dat





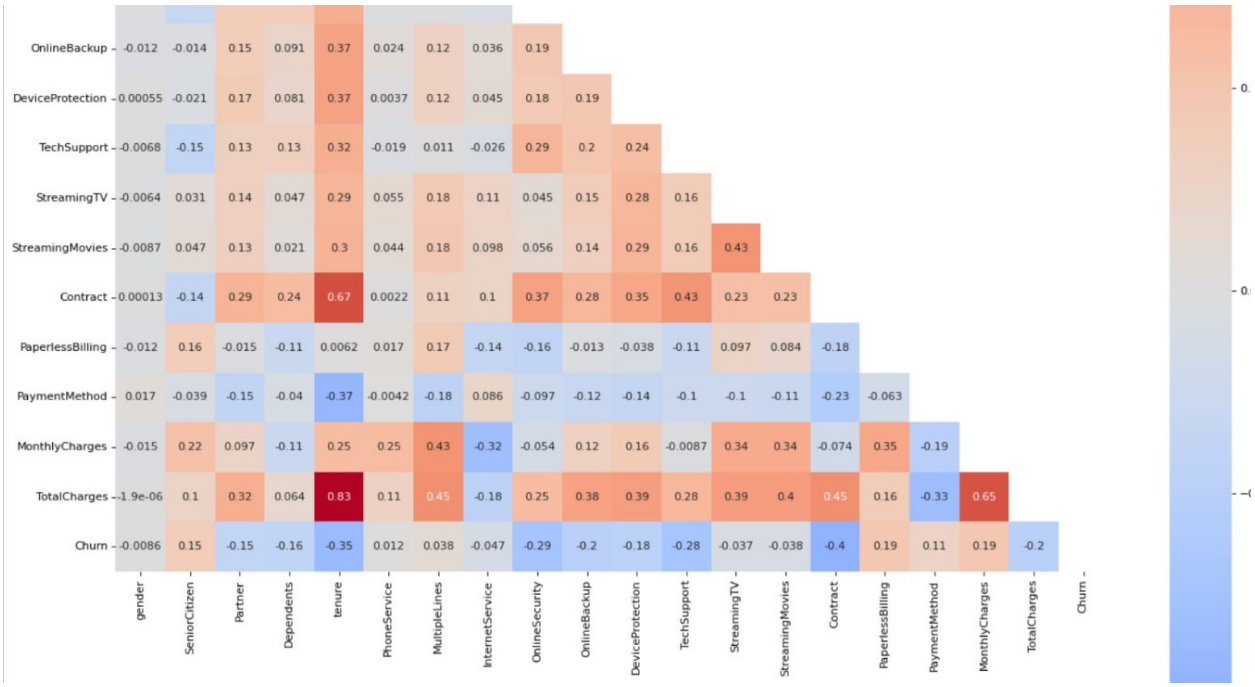




Using the analysis completed, we can say:
Gender: it seems to be an equal distribution of males and females with respect to churn intention (guess that gender is not important feature)
Senior citizen: There are much fewer senior citizens and there is a larger proportion of senior citizens churning. In the churn plot shows more young people are churning. (can be important)
Partner: People with partners and without partners have almost the same distribution of not churning, single people have more intention to churn. (can be important)
Dependents: There are much fewer people with dependents, there is a larger proportion of people with no dependents churning (looks like an important feature)
Phone Service: There are many more people with a phone service, almost the same intention of churn with people having phone service. (it should be important feature, but we need more explore it)
Multiple Lines: The numbers of people who have and do not have multiple lines are almost the same with respect to churn intention.(not important, but should be explore with partner and dependent)

Internet Service: There are many more people who have an internet service either with DSL or fiber, but there is a large proportion of people with fiber optic internet service who churn. (can expect that it is going to be an important prediction feature especially with Fiber Optic)
Online Security: there are more people with no online security and a larger proportion of the people has online security, has not churned. (customers having online security tend to stay within company compared to customers without online security)
Online Backup: There are more people with no online backup and those who has online backup has less probability of churn (customers having OnlineBackup tend to stay within company compared to customers without OnlineBackup)
Device Protection: There are more people with no device protection and those who have Device Protection has less probability to churn.(the same with previous)
Tech Support: there are more people with no tech support and those who have tech support have less probability to churn.(the same with previous)
Streaming TV: it seems to be almost an equal distribution of people who did and did not have streaming tv with respect to churn intention.
Streaming Movies: there are more people with no streaming movies and those who have streaming movies have more probability to churn.
Contract: There are many more people who are on a month-to-month contract and a large proportion of this group of people has churned. People with one year contracts are less churn. People with two years contracts are the least people may churn.(is one of the most important feature)
Paperless Billing: The number of people with paperless billing has quite larger proportion in people who has churned
PaymentMethod: There are more people adopting electronic check as a payment method and a large proportion of them have churned.

Here is the **Correlation Heatmap** for the data:



Conclusion, based on our analysis, we can see the more services added to the customer, the less people churn. Gender is not a feature since it does not give much information why customers are leaving, phone service and multiple lines aren't service. We decided to drop columns Gender, StreamingTV, StreamingMovies, PhoneService.

2.5 Choosing the models

We experimented with a variety of models to explore their strengths and limitations:

- **AdaBoost**: A boosting algorithm effective for imbalanced datasets and interpretable due to its feature importance scores.
- **Logistic Regression**: A baseline model for binary classification that provides clear probabilistic outputs.
- **Random Forest**: Robust to overfitting, handles categorical and numerical features well, and provides feature importance.
- **K-Nearest Neighbors (KNN)**: Non-parametric model that performs well for smaller datasets or where relationships are less linear.
- **Naive Bayes**: Fast and computationally efficient for high-dimensional data.
- **Support Vector Machine (SVM)**: Effective for high-dimensional spaces and provides robust classification boundaries.
- **Decision Tree**: Easy to interpret, suitable for feature importance analysis, and forms the basis for ensemble methods like Random Forest.

All above algorithms were tried out and gave following results:

Model	Accuracy
Adabost	77
Logistic Regression	75
Random Forest	74
K-Nearest Neighbors	75
Naive Bayes	74
Support Vector Machine	75
Decision tree	73

While analysis the confusion matrix of above training, it was observed that the training was giving a lot of false negatives i.e. customer who are leaving the service, but are predicted as retained. After analysis, we figured out that this is happening due to having significantly lesser amount of data for churn customers compared to non-churn customers.

2.6 Data Sampling and Its Need

The data imbalance can cause models to prioritize overall accuracy and can cause overfitting, often ignoring minority class. This results in poor recall, high false negatives and biased decision boundaries.

There are various techniques available to resolve this, called Sampling techniques.

There are two types of samplings:

Undersampling: Reduces the majority class size.
- Advantage: Faster training, balanced dataset.
- Limitation: Potential loss of important majority-class information.

Oversampling: Increases minority class size by duplicating samples.
- Advantage: No information loss.
- Limitation: Risk of overfitting due to repeated samples.

We decided to go ahead with oversampling technique. We tried with 2 available options, SMOTE and ADASYN. Finally, we focused on using SMOTE for our project.
SMOTE (Synthetic Minority Oversampling Technique) is a technique that generates synthetic samples to balance data instead of duplicating minority samples.
SMOTE (Synthetic Minority Oversampling Technique) is a resampling technique used to address imbalanced datasets in machine learning by generating synthetic examples for the minority class. This helps improve the performance of models by providing a more balanced representation of the classes.

How SMOTE Works:

1. Identify Nearest Neighbors:
   - For each instance in the minority class, SMOTE identifies its k-nearest neighbors within the same class based on a distance metric, typically Euclidean distance.
2. Random Selection:
   - A random instance is selected from the k-nearest neighbors of the original instance.
3. Synthetic Sample Generation:
   - A synthetic data point is created by interpolating between the original instance and the randomly selected neighbor. The interpolation formula is: Synthetic sample=$x+\delta \cdot (x_{neighbor} - x)$ where:x is the original instance, $x_{neighbor}$ is the randomly chosen neighbor, δ is a random number in the range [0, 1].
4. Repeat for Desired Oversampling:
   - This process is repeated until the desired number of synthetic samples is generated for the minority class.

Key Features of SMOTE:

- Balances Class Distribution:
  - By oversampling the minority class, SMOTE reduces the class imbalance.
- Does Not Duplicate Data:
  - Unlike random oversampling, which duplicates existing instances, SMOTE generates new synthetic data, helping reduce overfitting.
- Works in Feature Space:
  - Synthetic samples are created based on the feature values of the minority class, not just blindly duplicating rows.

This improved the models' ability to generalize for both Churn and non churn.

Applying SMOTE led to a noticeable improvement in most models' performance, especially in recall for the minority class. This highlighted the effectiveness of oversampling in tackling imbalanced data.

2.8 Model Stacking and Its Use Case

After the analysis of all the models after using SMOTE on the data, it was observed that the results were improved. We decided to try the stacking of models onto one another.

Model stacking is a technique in machine learning where multiple models are combined to make better predictions than any single model can achieve on its own. Think of it as a team of experts working together, where each expert specializes in something different, and a leader combines their opinions to make the best decision.
How Model Stacking Works:

1. Train Multiple Models:
    - Different models (like decision trees, logistic regression, or neural networks) are trained on the same data. These models might use different algorithms or settings (hyperparameters).
    - Each model tries to make predictions based on what it learns.
2. Collect Predictions:
    - Instead of directly using the predictions from one model, the predictions from all the models are collected. For example, if three models predict whether it will rain, you now have three different opinions.
3. Use a "Meta-Model" to Combine Predictions:
    - A new model (called a meta-model or stacker) is trained to learn how to combine the predictions from the individual models. This meta-model decides how much to trust each model's opinion.
    - For example, if one model is better at predicting rain and another is better at predicting sunshine, the meta-model will give more weight to the appropriate model for each scenario.

The meta-model makes the final decision based on the combined input from all the individual models.
In machine learning, stacking works similarly, combining diverse "opinions" (predictions) to arrive at the most accurate answer.
We chose **AdaBoost** and **Naive Bayes** as base models for stacking due to their complementary strengths:

- **AdaBoost**: Captures feature importance and handles imbalanced datasets effectively.

- **Naive Bayes**: Computationally efficient and excels in high-dimensional spaces.

The meta-model was **Logistic Regression**, chosen for its ability to interpret stacked predictions probabilistically.

2.10 Stacking Performance

The stacked model outperformed all standalone models, achieving the highest accuracy and recall. **The combination of AdaBoost's robust predictions with Naive Bayes' efficiency created a well-rounded solution for churn prediction.** This model gave **80% accuracy.**

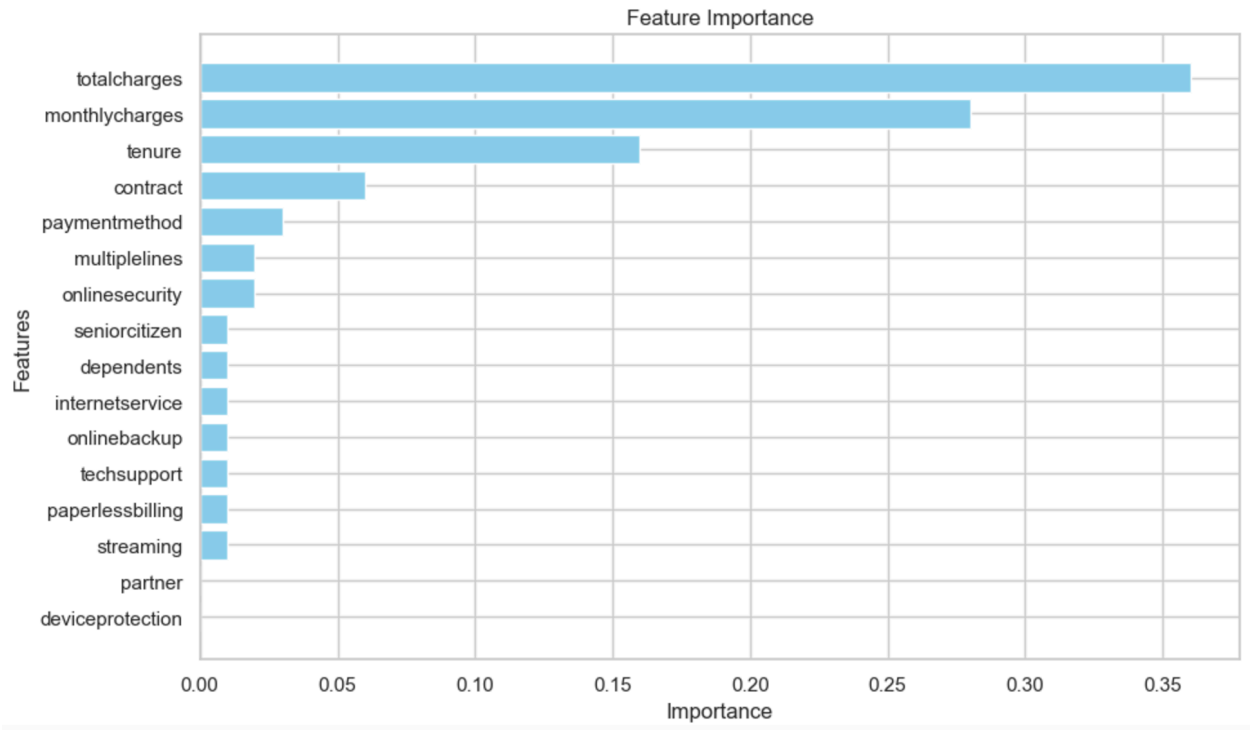
Evaluation Metrics Used for the models are
• Accuracy: Percentage of correctly classified instances.
• Precision: Proportion of true positives among all predicted positives.
• Recall: Proportion of true positives among all actual positives.
• F1 Score: Harmonic mean of precision and recall.
• ROC-AUC: Area under the receiver operating characteristic curve.

Test Results (Stacking Model):
• Accuracy: 80%
• Precision: 78%
• Recall: 81%
• F1 Score: 79%
• ROC-AUC: 0.85

Key Insights:

- The stacked model consistently outperformed individual models in recall, making it ideal for identifying churn-prone customers.
- Key features contributing to churn prediction were **Total Charges**, **Monthly Charges**, and **Tenure**.

Following was the feature importance found after training of the model



3. Challenges faced

• **Feature Dependencies**: Example: Total Charges depends on Tenure and Monthly Charges, which Naive Bayes struggles to capture.
• **Noise in Data**: AdaBoost overfocused on noisy samples, leading to overfitting.
• **Hyperparameter Tuning:** Finding the optimal kernel for SVM and the number of estimators for AdaBoost required significant effort.
• **Overfitting of the data:** Due to imbalance in data of classes(churn and non churn), the models tended to overfit.`

4. Conclusion

- **Best Model**: The stacked model (AdaBoost + Naive Bayes) proved to be the most effective, combining interpretability and predictive power.
- **Key Features**: Total Charges, Monthly Charges, and Tenure were the primary drivers of churn.
- The approach highlighted the value of sampling techniques and ensemble learning for churn prediction.

5. Future Work

- **Incorporating Temporal Analysis**: Analyze customer behavior trends over time to refine churn prediction.

- **Expand Feature Space**: Include additional external factors such as market trends or customer feedback.
- **Automated Model Selection**: Implement AutoML frameworks to optimize the choice of algorithms.
- **Real-Time Deployment**: Deploy the stacked model for live churn prediction and test its performance in real-world scenarios.
- **Explore advanced models** like Gradient Boosting, XGBoost, or CatBoost.
- Collect **more data** to improve model generalization.
- Integrate **sentiment analysis** from customer reviews as a feature.

6. Code and Data Location

• Code: In attached Zip, in .ipynb file
• Dataset Location: Link