BANA 273: MACHINE LEARNING ANALYTICS

Group Project: Telco Customer Churn

Group 16B:

Benjamin Fong

Eui Jin Shin

Himaneesh B E

Siddharth Yadav

Wenting Liu

Prediction on Telco Customer Churn

I. Introduction: explain the business idea, why it's important

Customer churn, or the rate at which subscribers discontinue their services, is a critical metric for businesses, particularly in competitive markets. This dataset includes a diverse set of features, such as customer demographics, usage patterns, and services subscribed, allowing for a nuanced analysis of factors influencing churn. By exploring this dataset, analysts and data scientists can uncover valuable insights into customer behavior, identify patterns that precede churn events, and develop predictive models to help telecommunication companies proactively address customer retention strategies. The dataset presents an opportunity to delve into the intricacies of customer relationships within the telecommunication sector, enabling businesses to make data-driven decisions to enhance customer satisfaction and reduce churn rates.

The problem we would like to address through the topic is as follows:

a. Reducing churn rate, which is a significant challenge for telco companies, impacting revenue, and analyzing the factors contributing to churn can help implement targeted strategies. Reducing churn is not only about retaining customers but is also crucial for financial stability, operational efficiency, brand reputation, and long-term growth. It reflects a commitment to customer satisfaction and ensures that the company remains competitive and resilient in dynamic market environments
b. Improving customer retention by understanding why customers leave allows for the development of customer-centric strategies to enhance customers' satisfaction. By consistently providing value, addressing customer concerns, and fostering positive relationships, businesses create an environment where customers are more likely to stay loyal and engaged over the long term. This not only contributes to a stable customer base but also sets the foundation for sustainable growth and success.

II. Data: data summary, description, visualization.

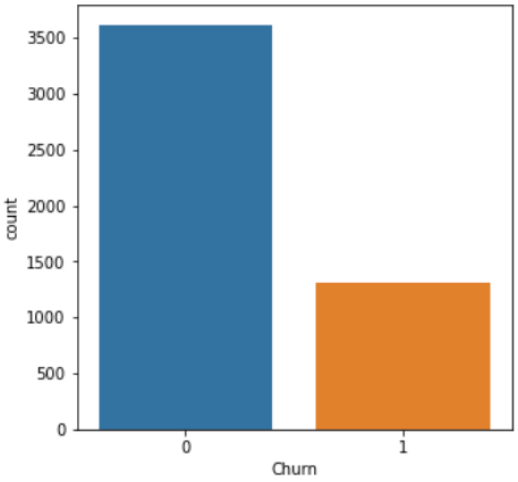
The dataset we are using comes from Kaggle.com. This data set provides information about customers of a telecommunications company and whether they churned (i.e., discontinued their services) or not. Here are some detailed descriptions of our data:

<u>Independent Variables:</u>

Dependents	Does the customer have dependents or not.
Tenure	How long the customer has subscribed to the company's services.
OnlineSecurity	Does the customer use the Online Security service or not.
OnlineBackup	Does the customer use the Online Backup service or not.
InternetService	Does the customer subscribe to Internet Service or not.
DeviceProtection	Does the customer use the Device Protection service or not.
TechSupport	Does the customer use Tech Support services or not.
contracts	The duration of the contract used.
PaperlessBilling	Is the bill sent on a paperless basis or not.
MonthlyCharges	Number of bills charged each month.

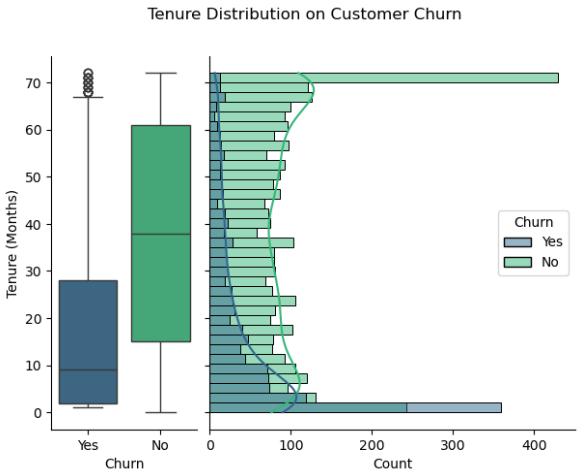
Dependent Variables:

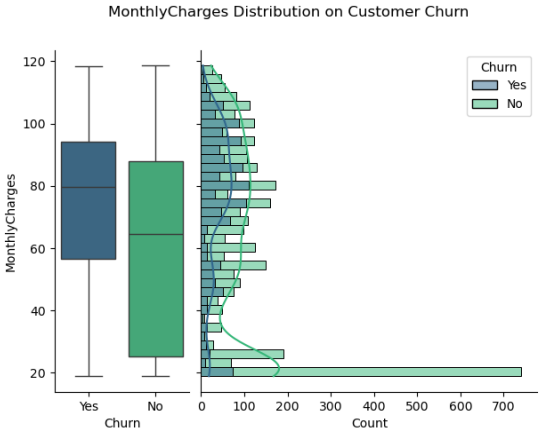
- Churn: Has the customer unsubscribed or not.



Graph1: Customer Churn Count (“1” means “yes”, and “0” means “no”)

By exploring the distribution of the churn data, it has shown that the probability of churn is 26.69%, which is a problem we should pay attention to.



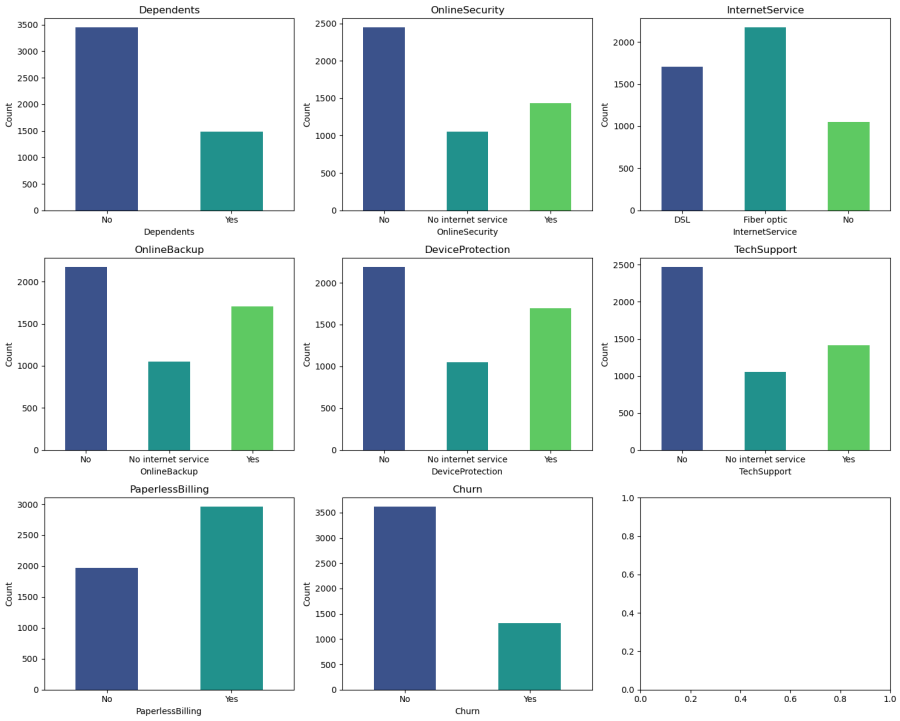


Graph2: Distribution of Tenure on Customer Churn

Graph3: Distribution of Monthly Charges On Customer Churn

The distribution of tenure suggests that shorter subscription durations are associated with higher customer churn.

The distribution of Monthly Charges indicates that churned customers generally pay more than those who don’t.



Graph4: Bar Chart for other attributes

There is a consistent trend of higher values on “No”, except for one attribute - paperless billing.

III. Analysis

First of all, the goal is to classify whether the customer is leaving or not and make predictions regarding this data. Using algorithms to classify data into particular segments is a **classification** problem. Secondly, in this case, we know the specific and detailed patterns, and we know how to classify them. Therefore, we will implement **supervised** learning methods for this problem.

The classification models we used to analyze the data were as followings:

1. Logistic Regression
2. Naïve Bayes
3. Decision Tree

IV. Show results without and with pre-processing data:

1. Logistic Regression

Method	Accuracy	Mean Accuracy	F1 score	AUC-ROC
Logistic Regression (NO pre-processing)	0.7839	0.7949	0.5782	0.83
Logistic Regression (Variable selection)	0.7839	0.7947	0.5782	0.83
Logistic Regression (SMOTE resampling)	0.7522	0.7949	0.6936	0.84
Logistic Regression (SMOTE + Boosting)	0.7511	0.7909	0.6927	0.83

2. Naive Bayes

Method	Accuracy	Mean Accuracy	F1 score	AUC-ROC
Naïve Bayes (NO pre-processing)	0.7586	0.7639	0.6136	0.80
Naïve Bayes (Variable selection)	0.7505	0.7602	0.5967	0.80
Naïve Bayes (SMOTE resampling)	0.7311	0.7639	0.7119	0.80
Naïve Bayes (SMOTE + Boosting)	0.7655	0.7897	0.7023	0.83

3. Decision Tree

Method	Accuracy	Mean Accuracy	F1 score	AUC-ROC
Decision Tree (NO pre-processing)	0.7281	0.7302	0.6136	0.67
Decision Tree (Variable selection)	0.7809	0.7953	0.5645	0.81

Decision Tree (SMOTE resampling)	0.7594	0.7953	0.7915	0.83
Decision Tree (SMOTE + Boosting)	0.7777	0.7566	0.7159	0.85

a. Without data pre-processing

Without any data pre-processing, the accuracies of Logistic Regression, Naïve Bayes, and Decision Tree are respectively 0.7839, 0.7586, and 0.7281. This is the first benchmark.

b. Variable Selection

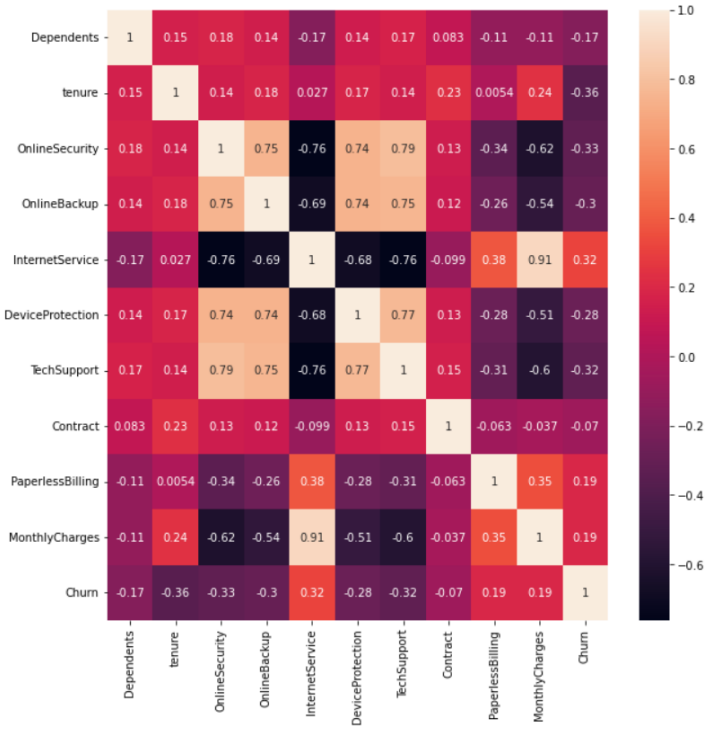
In the data pre-processing stage, we checked through any missing values, duplicated values, considering whether any attribute needs to be categorized, and whether our data is imbalanced. We created dummy variables and normalized numerical values as following:

```python
# create dummy variables

df = pd.DataFrame()

df["Dependents"] = [ 1 if x=="Yes" else 0 for x in data["Dependents"] ]
df["tenure"] = data["tenure"]
df["OnlineSecurity"] = [ 1 if x=="Yes" else 2 if x=="No internet service" else 0 for x
in data["OnlineSecurity"] ]
df["OnlineBackup"] = [ 1 if x=="Yes" else 2 if x=="No internet service" else 0 for x in
data["OnlineBackup"] ]
df["InternetService"] = [ 1 if x=="DSL" else 2 if x=="Fiber optic" else 0 for x in
data["InternetService"] ]
df["DeviceProtection"] = [ 1 if x=="Yes" else 2 if x=="No internet service" else 0 for
x in data["DeviceProtection"] ]
df["TechSupport"] = [ 1 if x=="Yes" else 2 if x=="No internet service" else 0 for x in
data["TechSupport"] ]
df["Contract"] = [ 1 if x=="Month-to-month" else 2 if x=="Two year" else 0 for x in
data["Contract"] ]
# In this case, 0 stands for One year

df["PaperlessBilling"] = [ 1 if x=="Yes" else 0 for x in data["PaperlessBilling"] ]
df["MonthlyCharges"] = data["MonthlyCharges"]
df["Churn"] = [ 1 if x=="Yes" else 0 for x in data["Churn"] ]
```

```
df.head(5)
```

Python

```
# Normalize "tenure" and "MonthlyCharges" columns -- (max - min) as denominator

df["tenure"] = (df["tenure"] - df["tenure"].min()) / (df["tenure"].max() -
df["tenure"].min())
df["MonthlyCharges"] = (df["MonthlyCharges"] - df["MonthlyCharges"].min()) /
(df["MonthlyCharges"].max() - df["MonthlyCharges"].min())

df.head(5)
```



Graph 5: Correlation Coefficient Heatmap

When we went into a deep observation on the correlations between different attributes with the target variable(“Churn”), almost all the correlation coefficients vary from 0.1 to 0.3. There's only one feature, which is the "Contract", which has a correlation coefficient of 0.07, that is less than 10%. Hence, we would like to see if there will be any change in accuracy by dropping this feature.

By dropping the column of “Contract” from building and fitting the models, the accuracies of the three models become 0.7839, 0.7505, and 0.7079. Even though the accuracies do not change too much, the F1 scores and the AUC-ROC all decreased. Therefore, we still include this feature for the latter modeling stage.

c. SMOTE Resampling

By implementing the SMOTE method to resample the data set and building the models again, the overall accuracies do not change too much, but the mean accuracies and the AUC-ROC increased. Therefore, it’s a good processing method to keep using.

```python
from imblearn.over_sampling import SMOTE

# make a copy of the df
df_copy = df.copy()

# define the indep. and dep.variables y and X
y = df_copy["Churn"]
X = df_copy.drop( ["Churn"], axis=1 )


# Create an instance of the SMOTE class
smote = SMOTE(sampling_strategy=0.66, random_state=42)


# Create an instance of the SMOTE class
X_resampled, y_resampled = smote.fit_resample(X, y)
```

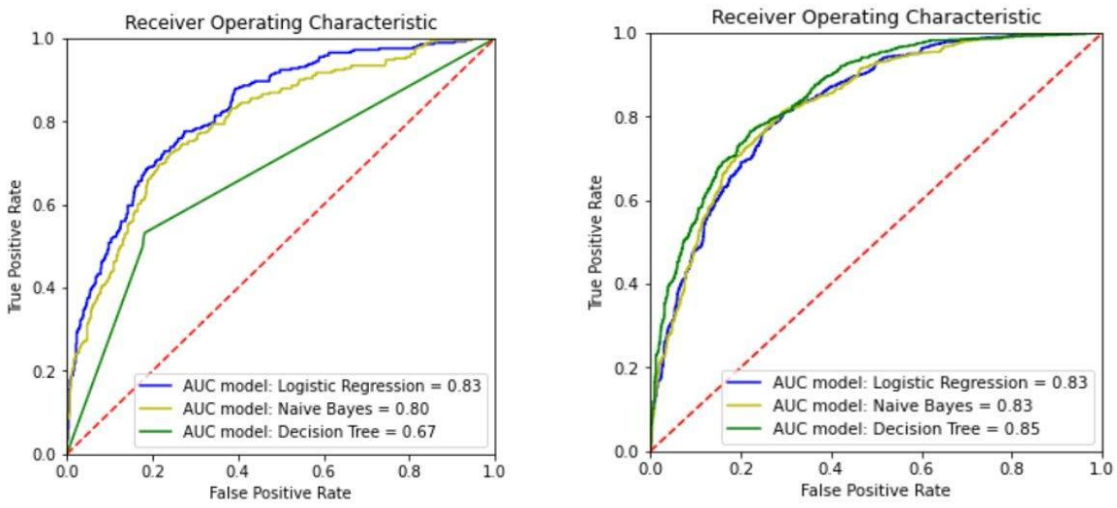
d. SMOTE + Boosting

Boosting is one of the ensemble learning methods that predict a target variable by training a number of models and combining their predictions together. An ensemble of models is generally more accurate than any of them works individually. Boosting uses a single learning algorithm but reweighs the training data, then combines their predictions to make it learn multiple models sequentially. Assigning a larger weight to the wrong predicted data, and a smaller weight to the correct ones, so that our next model could pay more attention to the wrong predictions.

We encountered an overfitting problem for logistic regression while using boosting method. Then we realized that we should always consider using weak or simple learners, so that overfitting problem can be prevented. After using the a simpler learning model, the overfitting problem has beem solved.

```python
# Pre pruning technique

from sklearn.model_selection import GridSearchCV

params = {'max_depth': [2,4,6,8,10,12],
          'min_samples_split': [2,3,4],
          'min_samples_leaf': [1,2]}

clf = tree.DecisionTreeClassifier()
gcv = GridSearchCV(estimator=clf,param_grid=params)
gcv.fit(X_train,y_train)


model = gcv.best_estimator_
print("Best Fit Model")
print(model)


# 【Notice:】  By using the max_depth=4,
# it improves the accuracy of our model by 7%  (0.71 --> 0.78)
```

One thing worth mentioning is that we used different techniques to improve our models. For example, we changed hyper-parameters for a better result of performance, and used the pre-pruning technique for the decision tree model.



Graph 6: The AUC-ROC curves of the models without and with pre-processing

The graph on the left is the ROC curve without any data pre-processing, and the one on the right is the ROC curve with implementation of SMOTE resampling and boosting method. By looking at the final ROC curve, all of the three models have a good performance on prediction after they are adjusted and improved using different methods. The Decision Tree model performs a slightly better function compared to the other two models, because its true positive rate approaches 1 the fastest.

V. Takeaways

1. Even though there weren't any missing values or duplicated values in this data set, it's still important to check for those every single time while doing data preparation, because they might influence the data and our classification result.
2. Creating dummy variables, normalizing numerical values, and handling with categorical variable are also the things we need to pay attention to.
3. Checking for correlations between all the factors. If two factors have a very high correlation coefficient, which means that they will affect one and other in a significant level, especially those with the dependent variable which we are interested in to make classifications or predictions. If this situation happens, consider dropping the column which is highly correlated to the factor that we want to classify or predict.

4. Identifying the correct model to choose is also very important. Firstly, we need to understand the business problem and the data set itself. Are we going to use supervised learning or non-supervised learning methods? And what are the options of models we could choose from.
5. Making copies of the original data and implementing different models on the copies will keep the original data safe.
6. After implementing the models, comparing them according to their accuracy in classification and prediction, so that we could pick the one with the best performance. Do not only use the overall accuracy score, but also use the stratified accuracy, which we can do via cross-validation, and f1 score.
7. Analyzing the data and findings, converting them into a well story-telling conclusion will help stakeholders or business managers to better understand the situation to make a better decision.

VI. Interpretation

Data pre-processing is crucial for accurate machine learning models

The study highlights the importance of data pre-processing in enhancing the performance of machine learning models. By removing the redundant "Contract" feature and employing SMOTE resampling, the accuracy of the models improved significantly. This underscores the need for careful data preparation to ensure reliable predictions.

Ensemble learning methods like boosting offer enhanced accuracy

The study demonstrates the effectiveness of ensemble learning techniques, particularly boosting, in boosting the accuracy of machine learning models. Boosting involves training multiple models and combining their predictions, leading to improved overall performance. However, to avoid overfitting, it is essential to employ weak or simple learners.

Decision tree model emerges as the top performer

Among the three models evaluated, the decision tree model emerged as the frontrunner in terms of prediction accuracy. Its ability to rapidly approach a true positive rate of 1 indicates its proficiency in identifying churned customers.

Key takeaways for business managers

Business managers can derive valuable insights from the study to optimize their machine learning strategies:

- Data pre-processing: Implement data pre-processing techniques to enhance model accuracy.
- Ensemble learning: Consider employing ensemble learning methods like boosting for improved performance.
- Model selection: Carefully select the machine learning model that best suits the specific problem at hand.
- Data analysis: Analyze and interpret data to provide stakeholders with meaningful insights for informed decision-making.

By incorporating these recommendations, business managers can leverage machine learning to make informed decisions, optimize customer retention strategies, and ultimately enhance business outcomes.