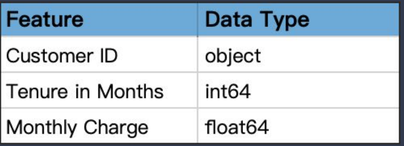
Predictive Analysis of Customer Churn

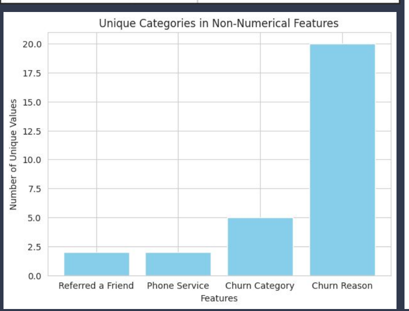
Tuesday Section Group 7: Dhvanil Nanshah, Leah Uzzan, Lu Liu, Ta-Wei Huang, Yaqing Wang

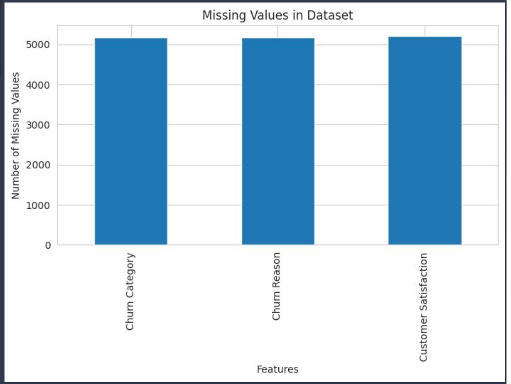
TELECOM

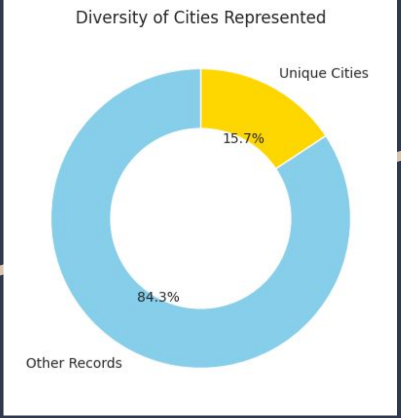
Part 1: Initial Data Exploration

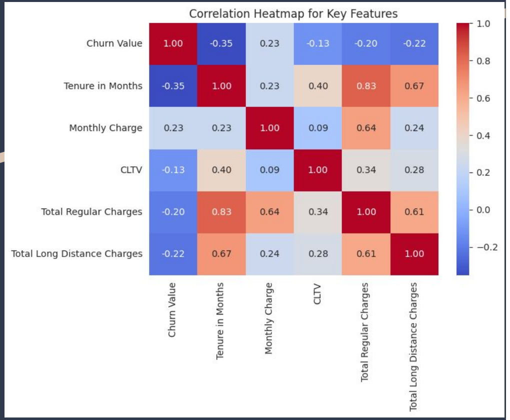


Feature	Data Type
Customer ID	object
Tenure in Months	int64
Monthly Charge	float64







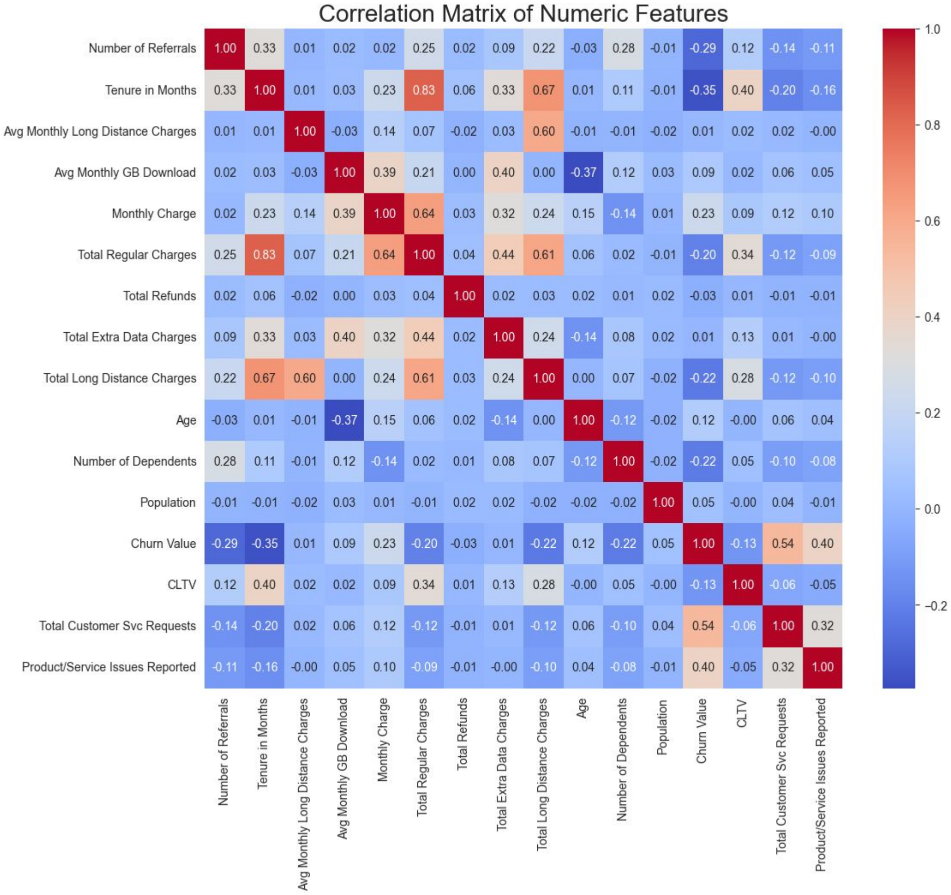


- **Source & Size**:
  - Sourced from telecom customer interactions, the dataset encompasses 7,043 records, reflecting a comprehensive customer base.
- **Feature Richness**:
  - 46 different features for each record.
- **Objective**:
  - Aim to uncover underlying patterns and predictors of customer churn to inform retention strategies.
- **Data Types**:
  - Varied data types including `object`, `int64`, and `float64`.
  - Categorical data present in the form of `object` type for features like `Gender` and `Paying Method`.
- **Missing Values**:
  - Identified missing values in `Customer Satisfaction` and other columns.
  - A strategic approach is required for handling these missing entries.
- **Unique Categories**:
  - High number of unique entries in `City` feature (1,106 unique cities).which suggests a wide geographical distribution of customers.
  - For non-numerical features, there are several binary features (with 2 unique values each), such as `Referred a Friend`, `Phone Service`, etc.
  - `Churn Category` has 5 unique values and `Churn Reason` has 20, indicating multiple reasons customers might leave.
- **Correlation Insights**:
  - Initial correlation analysis reveals significant relationships between features.
  - (Next slide)

Part 1: Initial Data Exploration

Note: Defining strong correlation as between 0.6 and 0.8, and very strong correlation as >0.8

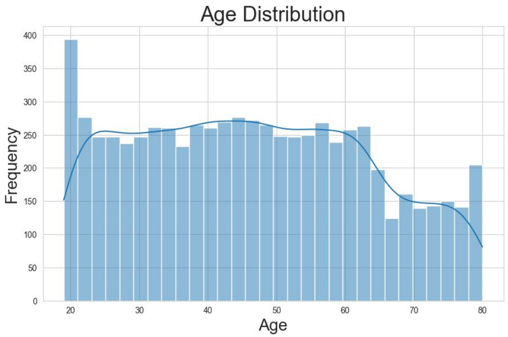
- There are 5 pairs of correlations that stand out as strong, or very strong, with others not listed that fall between weak to moderately strong correlations
  - **Monthly Charge** vs. **Total Regular Charge**
  - **Total Long Distance Charges** vs. **Tenure in months**
  - **Total Long Distance Charges** vs. **Total Regular Charge**
  - **Total Regular Charge** vs. **Tenure in Months** (very very strong)
  - **Avg Monthly Long Distance charges** vs **Total Long Distance Charges**
- The above mentioned pairs are all numerical continuous values, suggesting the need to explore lasso, ridge and elastic net regularization methods if implementing a logistic regression model
  - For decision trees, feature selection should be implemented and tested for a variety of selections among these strongly correlated features for exclusion/inclusion
- Some of these features also show low to no correlation with churn value (around 0), suggesting some of them may potentially need to be dropped

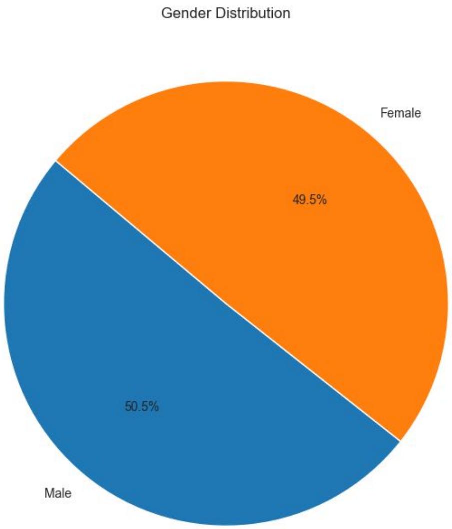


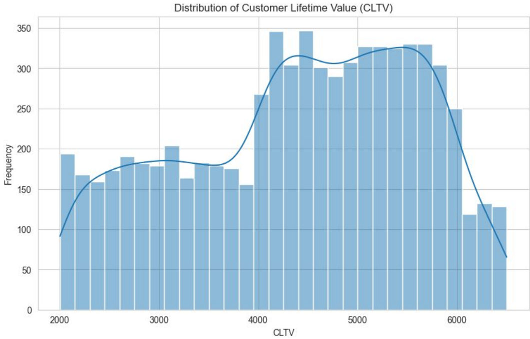
Correlation Matrix of Numeric Features

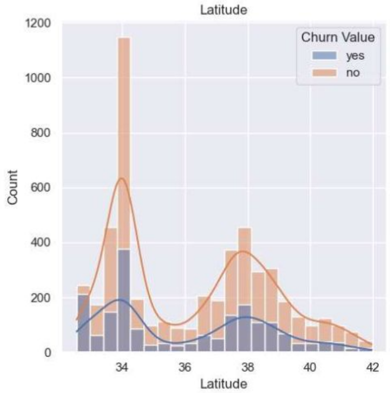
Part 1: Initial Data Exploration

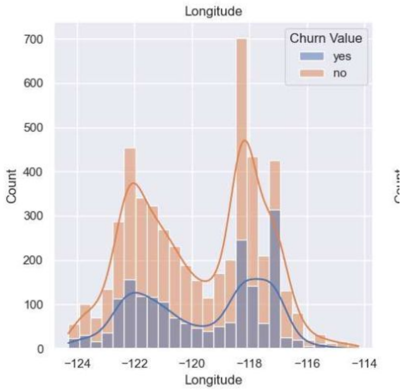
- The dataset is moderately imbalanced where we are interested in predicting the minority class, represented by the customers who churn
  - **Churn Count**: 1869
  - **Non-Churn Count**: 5174
- The frequency distribution of ages of customers is roughly uniform between 22 and 64 with a decline after 64, possibly due to relocation for better care/retirement, and between 18 and 22, where the frequency is higher
- The gender distribution is roughly proportional
- The distribution of CLTVs is a right skewed bimodal
  - This suggests there might be two dominant subgroups of types of customers responsible for a majority of the non-churn as longer brand loyalty results in a larger CLTV in most cases, but especially for the telecom industry as their realistic upper bound on profit per individual is significantly limited compared to something like construction
- The latitude and longitude histograms illustrate that the distribution shapes of churn and non churn are roughly identical across latitude and longitude measurements. This reinforces the intuition that those features, unless further processed, hold little to no value in training the intended model







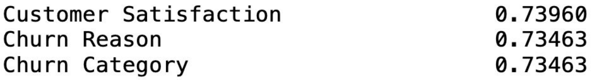




Part 2: Cleaning and Sampling

1. Missing data

After investigating data, we found only three attributes has the highest missing proportion (>70%), so we decided to drop the features and focus on the remaining. Additionally it was deemed that Churn Reason and Churn Categories could not provide meaningful data not already captured by the trends of the other features.



```
Customer Satisfaction          0.73960
Churn Reason                   0.73463
Churn Category                 0.73463
```

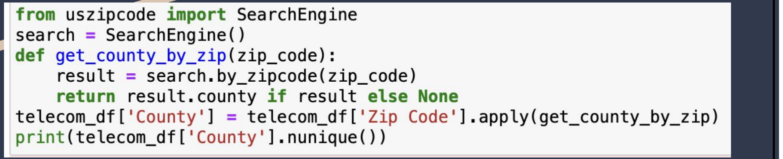
2. Data Duplication Verification

Verified the lack of duplicate entries by checking the number of unique Customer IDs to the number of overall entries in the database.

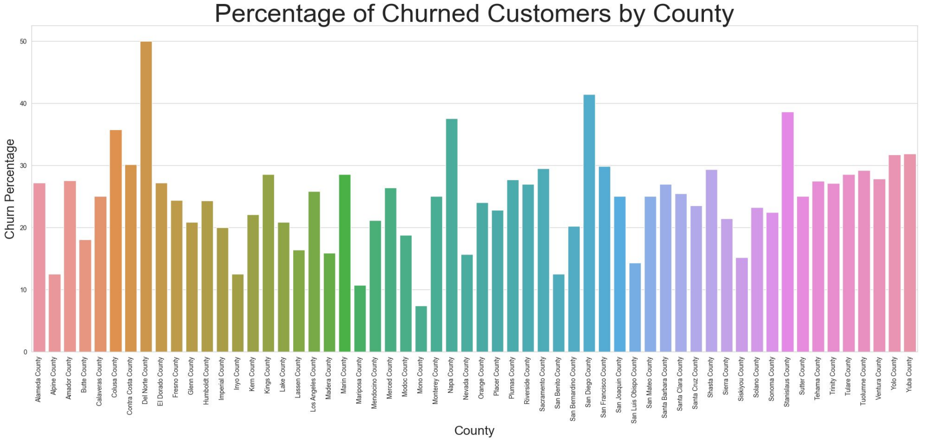
3. Feature Engineering

a) Zip Code to County Mapping

To preserve some geospatial information from the data set, given the lack of practical utility of longitudinal, latitudinal, city and zip code (the last two of which had a quantity of unique values paralleling the data set in magnitude), we generated a new column for the counties represented by the zipcodes. This decreased the unique values from thousands to just 57 representative Counties. See graphical figure and caption for more details.



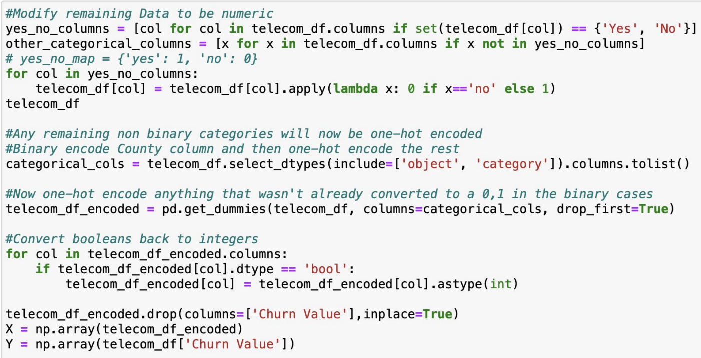
```
from uszipcode import SearchEngine
search = SearchEngine()
def get_county_by_zip(zip_code):
    result = search.by_zipcode(zip_code)
    return result.county if result else None
telecom_df['County'] = telecom_df['Zip Code'].apply(get_county_by_zip)
print(telecom_df['County'].nunique())
```



This figure shows that there is notable variation in churn rates depending on the county of California being observed. This geospatial data could be further distilled by potentially partitioning the state and capturing counties into regions. This sort of geospatial data captures information on things such as local consumer values and local market competition.

b) Categorical Feature Encoding

Binary Categorical Features - A simple label coding method, with Yes replaced with 1 and No with 0 ***Non-binary categorical features*** - For the remaining categorical features, we dropped the "city" and "customerID" features . Since the former information is contained in the zipcode, the latter is often of no practical meaning. Then the rest we performed One Hot Encoding for data manipulation. No features were found requiring or benefiting from ordinal encoding.

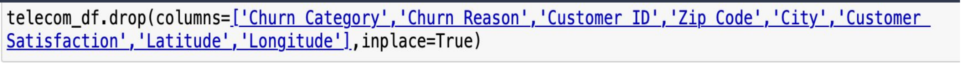


```
#Modify remaining Data to be numeric
yes_no_columns = [col for col in telecom_df.columns if set(telecom_df[col]) == {'Yes', 'No'}]
other_categorical_columns = [x for x in telecom_df.columns if x not in yes_no_columns]
# yes_no_map = {'yes': 1, 'no': 0}
for col in yes_no_columns:
    telecom_df[col] = telecom_df[col].apply(lambda x: 0 if x=='no' else 1)
telecom_df

#Any remaining non binary categories will now be one-hot encoded
#Binary encode County column and then one-hot encode the rest
categorical_cols = telecom_df.select_dtypes(include=['object', 'category']).columns.tolist()

#Now one-hot encode anything that wasn't already converted to a 0,1 in the binary cases
telecom_df_encoded = pd.get_dummies(telecom_df, columns=categorical_cols, drop_first=True)

#Convert booleans back to integers
for col in telecom_df_encoded.columns:
    if telecom_df_encoded[col].dtype == 'bool':
        telecom_df_encoded[col] = telecom_df_encoded[col].astype(int)

telecom_df_encoded.drop(columns=['Churn Value'],inplace=True)
X = np.array(telecom_df_encoded)
Y = np.array(telecom_df['Churn Value'])
```

Part 2: Cleaning and Sampling

4. Preliminary Feature Selection

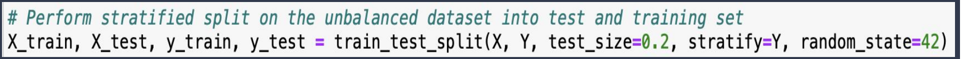
- Removed Longitude and Latitude as those continuous variables provide minimal information by themselves given the complex partitioning of the state of California by things such as socio-economics, predominant ethnicities and distribution of competition.
- Removed Customer ID, and Zip Code as they provided no information relevant to the model's training (Zip Code has already been used to engineer the feature 'County'
- Remainder removed due to missing data outlined in section 1 of Cleaning and Sampling



```
telecom_df.drop(columns=['Churn Category','Churn Reason','Customer ID','Zip Code','City','Customer Satisfaction','Latitude','Longitude'],inplace=True)
```

5. Preliminary Data Sampling

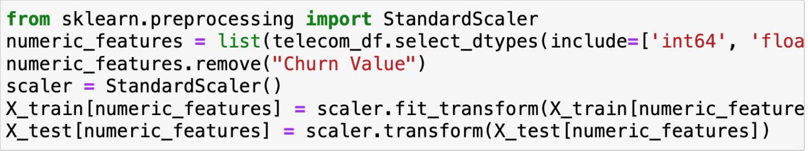
- Preliminary approach is to utilize all data but split it using a stratified split as to maintain the original ratios of the Churn and Not Churn classes in the training and test data. This is an appropriate approach as the initial objective is to create baseline balanced weight decision tree and logistic regression models to weight future models against and guide strategic planning on how to handle the imbalanced data set.



```
# Perform stratified split on the unbalanced dataset into test and training set
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, stratify=Y, random_state=42)
```

6. Data Standardization

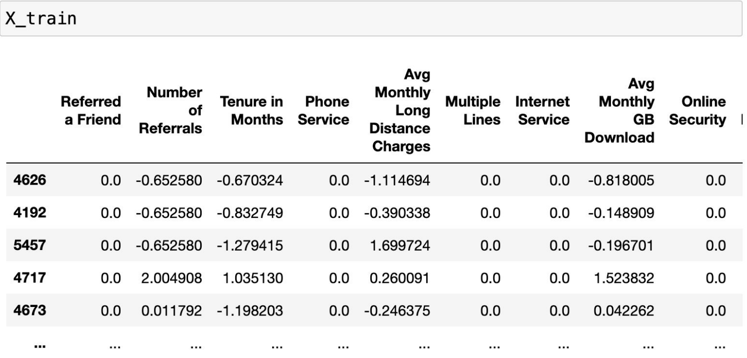
- Numerical features with high variance can skew the performance of a baseline logistic regression model, though tree-based models are less affected. Standardizing these features to a mean of zero and standard deviation of one is essential to balance their contribution to the model's predictive power and prevent bias towards larger-scale features.



```
from sklearn.preprocessing import StandardScaler
numeric_features = list(telecom_df.select_dtypes(include=['int64', 'floa
numeric_features.remove("Churn Value")
scaler = StandardScaler()
X_train[numeric_features] = scaler.fit_transform(X_train[numeric_feature
X_test[numeric_features] = scaler.transform(X_test[numeric_features])
```

7. Training Data

- After splitting the training and test data, the total size of x_train data is 5634*103 and the total size of x_test data is 1409*103.

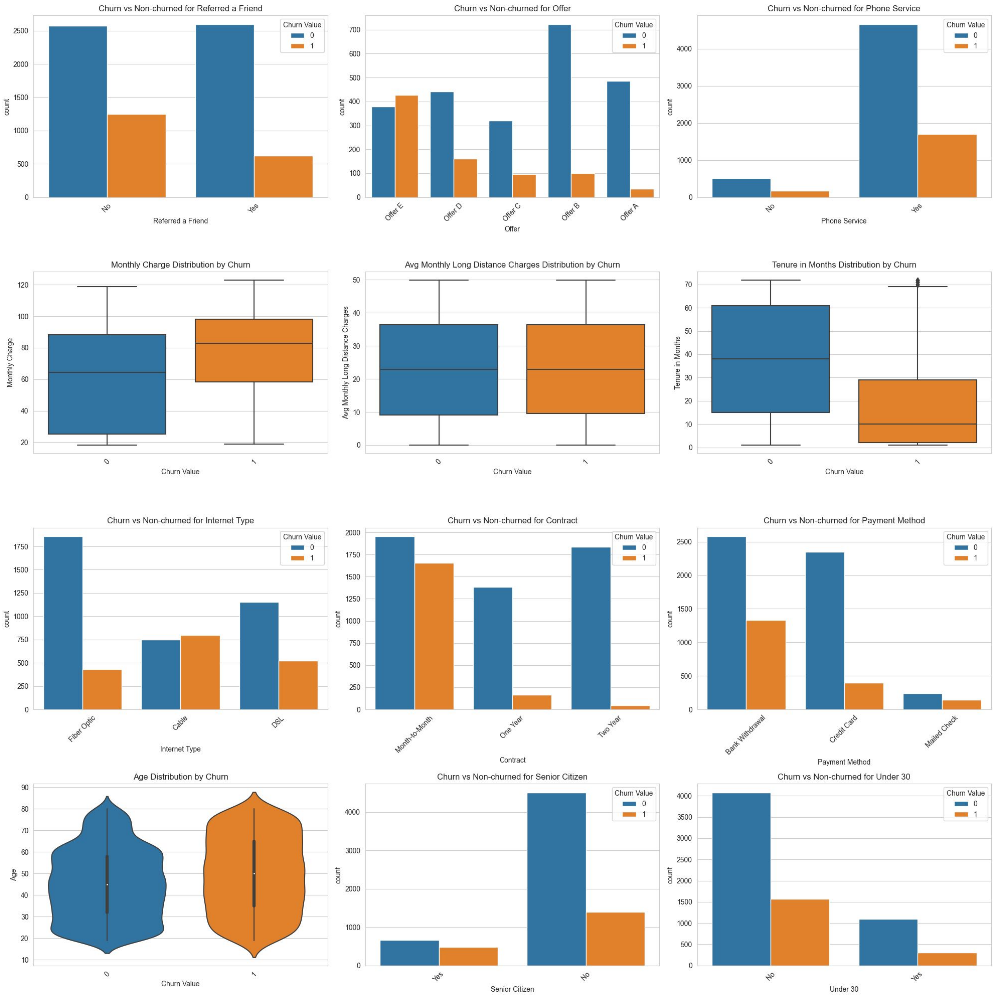


```
X_train
```

	Referred a Friend	Number of Referrals	Tenure in Months	Phone Service	Avg Monthly Long Distance Charges	Multiple Lines	Internet Service	Avg Monthly GB Download	Online Security
4626	0.0	-0.652580	-0.670324	0.0	-1.114694	0.0	0.0	-0.818005	0.0
4192	0.0	-0.652580	-0.832749	0.0	-0.390338	0.0	0.0	-0.148909	0.0
5457	0.0	-0.652580	-1.279415	0.0	1.699724	0.0	0.0	-0.196701	0.0
4717	0.0	2.004908	1.035130	0.0	0.260091	0.0	0.0	1.523832	0.0
4673	0.0	0.011792	-1.198203	0.0	-0.246375	0.0	0.0	0.042262	0.0
...	...	...	...	...	...	...	...	...	...

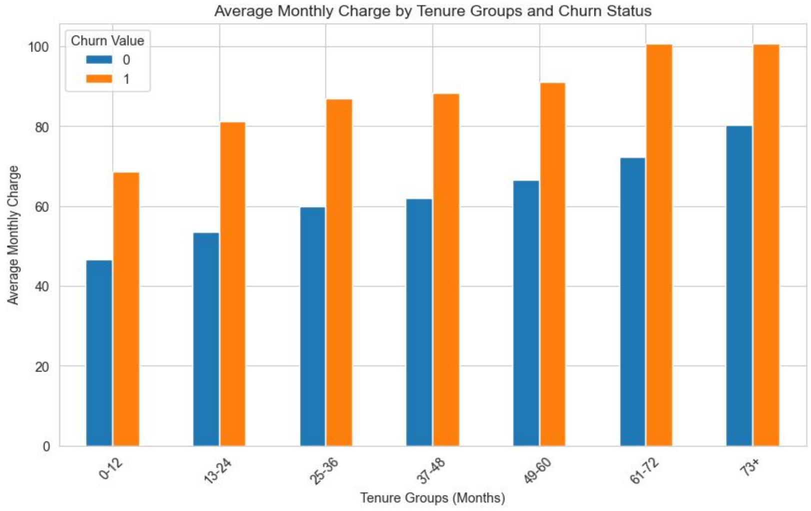
Part 3: Insights from Exploratory Data Analysis

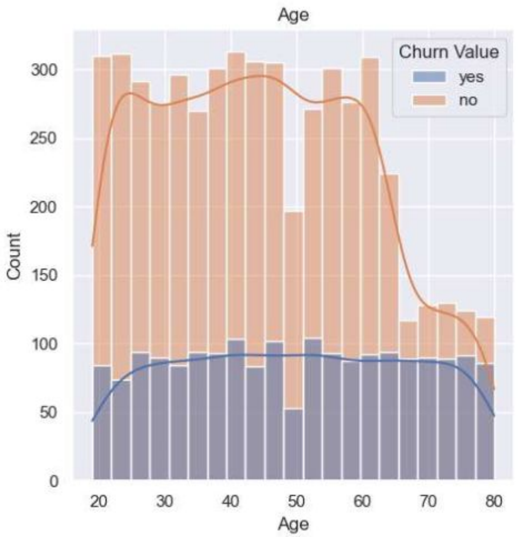
- Those who churn are possibly slightly more likely to not refer friends (intuitive)
- Certain offers from the telecom company are much more likely to result in customer churn over time (ex: provider contracts vs. non-contract)
- A majority (by volume) of churning happens with customers who subscribe to the phone service
- Churning appears to happen more frequently with customers paying higher monthly charges (intuitive)
- Long distances charge statistical measures are roughly evenly distributed for those that churn and those that do not churn
- Those that churn tend to have shorter tenures with the service
- The cable service has a significantly larger churn ratio (almost 50%) than the DSL and Fiber Optic service(lowest ratio). Perhaps suggesting the ordinal ranking of the values for price provided by each service, or the availability of alternatives (including competitors)
- Churning is significantly more likely to occur amongst customers paying month-to-month vs being on contracts. Coincides with the churn rate for the different offers
- Different payment methods seem to produce different churn ratios, suggesting that certain payment methods might be more popular with particular demographics (intuitive). Ex: Mailed checks might be more popular among the elderly, who happen to also churn more
- The age distribution of those who churn are fairly uniformly distribute except at the tail ends where the very old and very young (teens) are located. Those that do not churn have a roughly uniform distribution of ages up until the age 65 after which those that do not churn are in lower frequencies. This may be due to death or relocation.

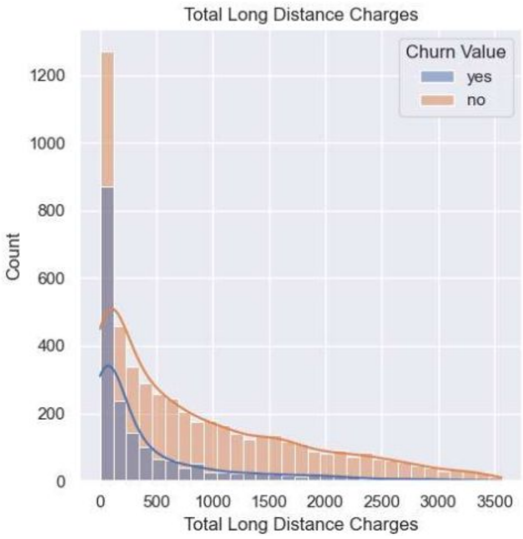


Part 3: Insights from Exploratory Data Analysis

- The figure at the top suggests that as the tenure of customers increases, their tolerance for higher average monthly charges increases, as indicated by the trend of increasing average monthly charges of those that churn and do not churn going from smaller to larger bins of tenure durations
- The figure on the bottom left reinforces the prior analysis of age not influencing churn rate in any noticeable way as suggested by the uniform distribution prior to around the age of 64, after which the churn ratio drastically increases
  - This further implies that the “senior” status of customers can be an important discriminating factor in determining churn
- The figure on the bottom right that the telecom company either provides good value or has little competition in the market for long distance calls. Long distance calls, as per the date of origin of this data set, would refer to intercontinental calls. This proposes the notion that those with higher total long distance charges, thus a greater need to speak to people located far away, are less likely churn for whatever reason, whether that be a lack of options or good value.







Part 4: Proposed ML techniques

Baseline model:

- Balanced Decision Tree
- Logistic Regression

Improved models:

- Random Forest
- Gradient Boosting
- AdaBoost

Deal with Imbalanced Data:

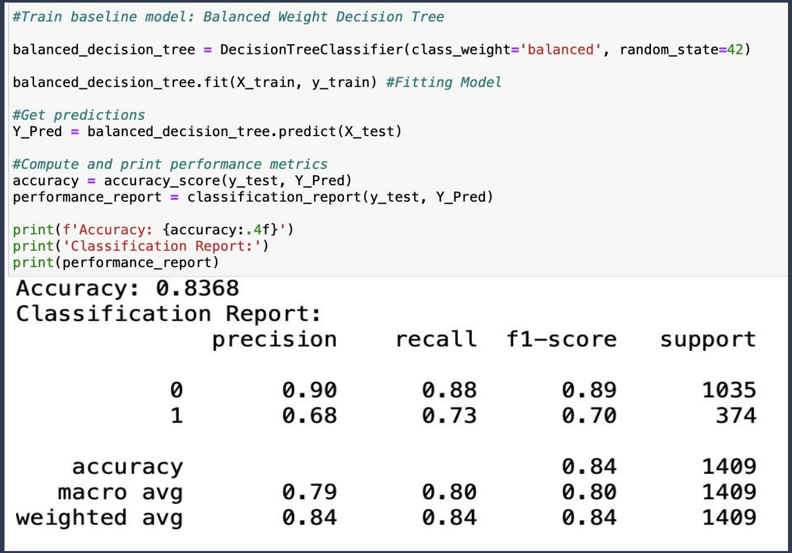
- Oversampling
- Downsampling
- SMOTE

Metrics to consider:

- Recall & Precision for minority class
- F1
- AUC - ROC Score

Part 4: Defining Baseline Models

Balanced Weight Decision Tree

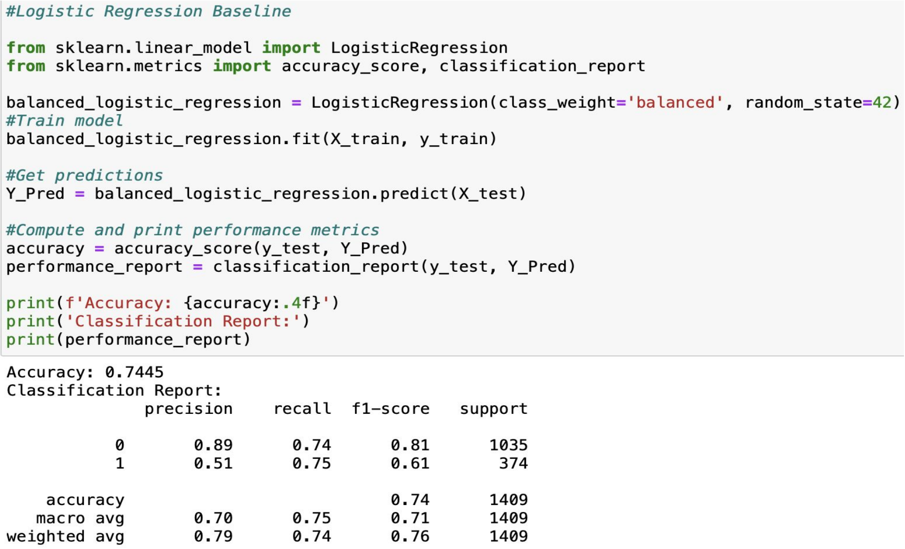


```
#Train baseline model: Balanced Weight Decision Tree

balanced_decision_tree = DecisionTreeClassifier(class_weight='balanced', random_state=42)

balanced_decision_tree.fit(X_train, y_train) #Fitting Model

#Get predictions
Y_Pred = balanced_decision_tree.predict(X_test)

#Compute and print performance metrics
accuracy = accuracy_score(y_test, Y_Pred)
performance_report = classification_report(y_test, Y_Pred)

print(f'Accuracy: {accuracy:.4f}')
print('Classification Report:')
print(performance_report)
```

```
Accuracy: 0.8368
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.88      0.89      1035
           1       0.68      0.73      0.70       374

    accuracy                           0.84      1409
   macro avg       0.79      0.80      0.80      1409
weighted avg       0.84      0.84      0.84      1409
```

- Model shows does well predicting the majority class and has a good balance between precision and recall as indicated by the F1-Score
- Model predicts only about 68% of Churn cases hence misses 32% of them - Overall poor with minority class predictions
- Low F1-Score for the churn case indicates it could do a better balancing the two
- Overall a better job can be done handling the imbalanced data sets and more robust models should be explored
- Weighted Average > Macro Average suggests the majority class performance is significantly more influential on the overall performance

Balanced Logistic Regression Model



```
#Logistic Regression Baseline

from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, classification_report

balanced_logistic_regression = LogisticRegression(class_weight='balanced', random_state=42)
#Train model
balanced_logistic_regression.fit(X_train, y_train)

#Get predictions
Y_Pred = balanced_logistic_regression.predict(X_test)

#Compute and print performance metrics
accuracy = accuracy_score(y_test, Y_Pred)
performance_report = classification_report(y_test, Y_Pred)

print(f'Accuracy: {accuracy:.4f}')
print('Classification Report:')
print(performance_report)
```

```
Accuracy: 0.7445
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.74      0.81      1035
           1       0.51      0.75      0.61       374

    accuracy                           0.74      1409
   macro avg       0.70      0.75      0.71      1409
weighted avg       0.79      0.74      0.76      1409
```

- The balanced logistic regression also suffers from doing more poor on the minority class but the performance deficit is more severe
- Overall suffers the same faults as the balanced weight decision tree model but performs worse across all categories.
- Suggests optimizing the tree classification approach may be better for this modeling problem