FACTORS INFLUENCING CHURN IN TELCO

MATIAS A, CLOE C, VITTORIO F, MARCELA F,
MORITZ G, PILAR G, & ALLAN S.

AI & MACHINE LEARNING

REPORT

Index

AI & MACHINE LEARNING

PREDICTING CHURN

Brief Introduction

The global telecommunications industry has a market size of USD 2.32 trillion but it is over saturated, leading to high customer churning. This analysis aims to find the factors that lead to consumers churning with the goal of helping business create a better strategy.

AI & MACHINE LEARNING

HYPOTHESES

H1

Customers with monthly contracts are more likely to churn compared to ones with yearly and bi-yearly contracts.

H2

In counties with very-low income per capita, a higher churn rate is associated with women than with men, and the opposite behavior is expected in non very-low income counties.

H3

Customers paying via mailed checks have a higher churn rate compared to those using bank withdrawals or credit cards.

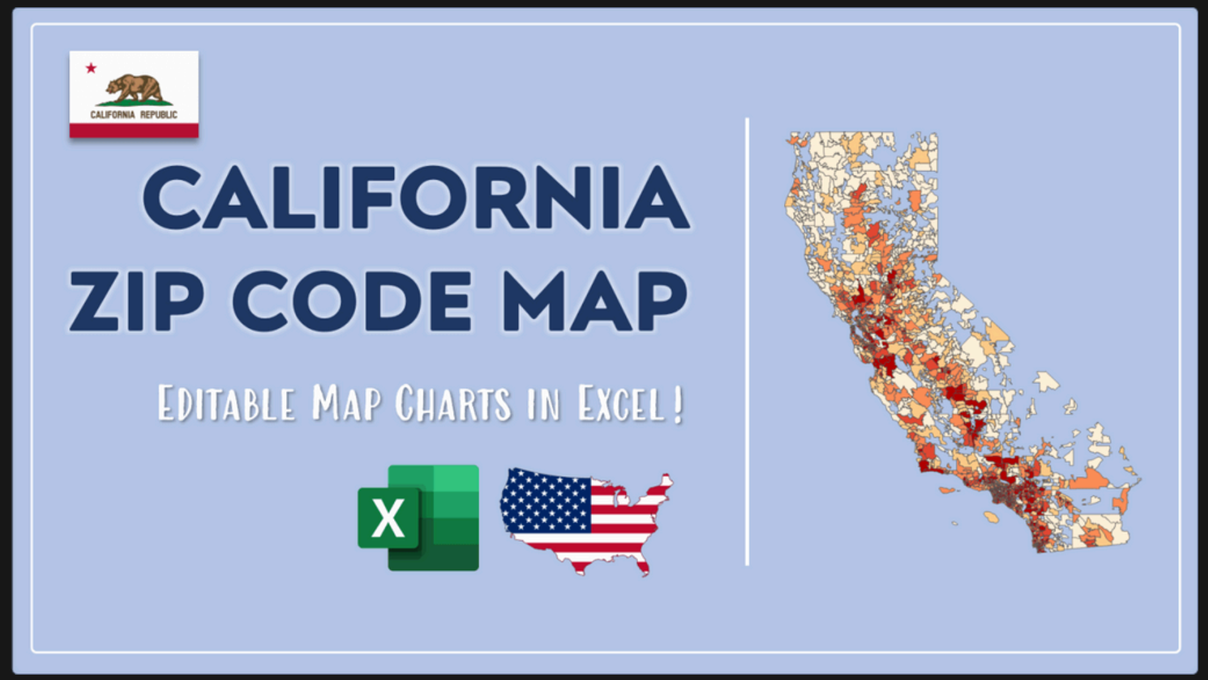
H4

Customers subscribing to multiple company services are paradoxically more likely to churn if their satisfaction score drops below a threshold of 3, as their expectations of service quality are higher.

DATA SOURCES

kaggle

IBM



EDD Employment Development Department

State of California

VARIABLES USED

TARGET VARIABLE

CHURN LABEL: YES = THE CUSTOMER LEFT THE COMPANY THIS QUARTER. NO = THE CUSTOMER REMAINED WITH THE COMPANY.

HIGHLIGHTED VARIABLES

GENDER: THE CUSTOMER'S GENDER: MALE, FEMALE.
CONTRACT: INDICATES THE CUSTOMER'S CURRENT CONTRACT TYPE: MONTH-TO-MONTH, ONE YEAR, TWO YEAR.
PAYMENT METHOD: INDICATES HOW THE CUSTOMER PAYS THEIR BILL: BANK WITHDRAWAL, CREDIT CARD, MAILED CHECK.
SATISFACTION SCORE: A CUSTOMER'S OVERALL SATISFACTION RATING OF THE COMPANY FROM 1 (VERY UNSATISFIED) TO 5 (VERY SATISFIED).
COUNTY INCOME PER CAPITA: INCOME PER CAPITA OF THE COUNTY WHERE THE CUSTOMER LIVES.
ADD ONS: NUMBER OF ADD ONS A CUSTOMER HAS IN THE PACKAGE

OBS:

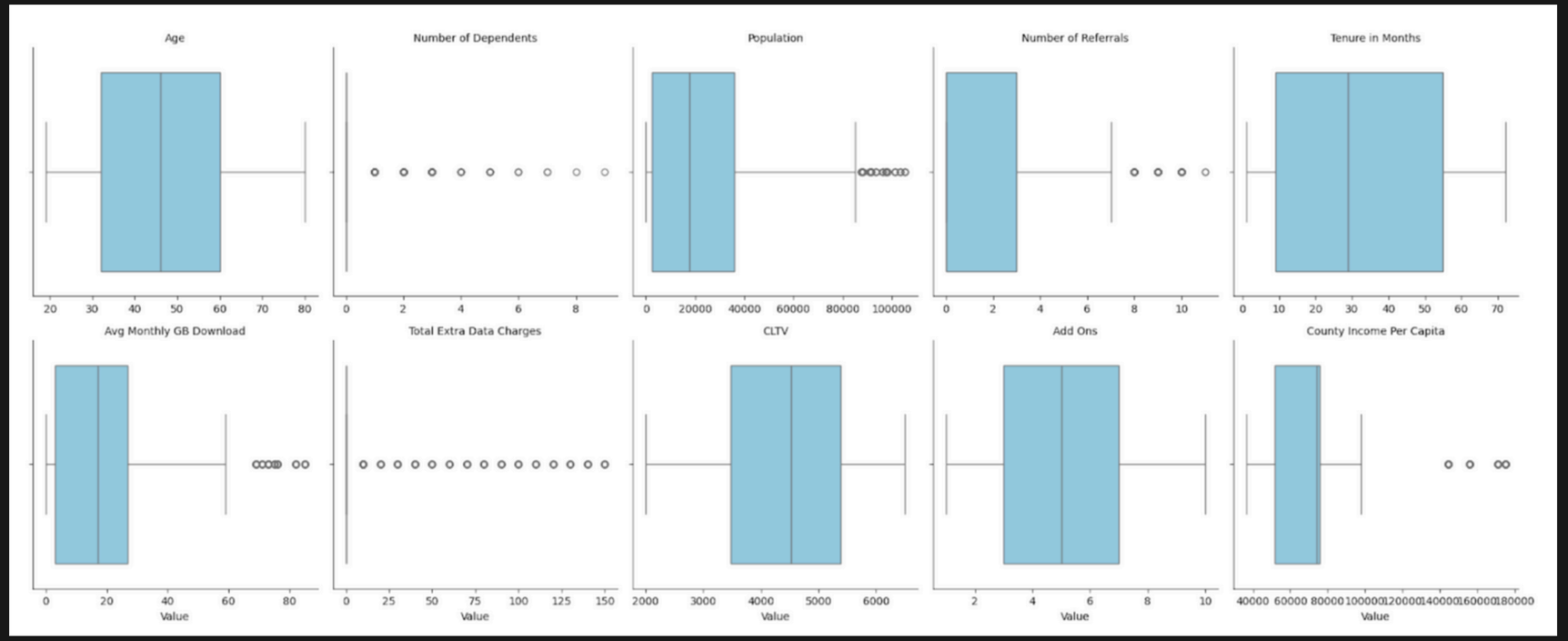
ADD ON VARIABLE WAS CREATED BY AGGREGATING MANY VARIABLES RELATED TO THE SERVICES THE CUSTOMER HAD IN THE PACKAGE

DESCRIPTIVE ANALYSIS

NUMERICAL VARIABLES

1. SUMMARY STATISTICS

	County Income Per Capita	Add Ons
count	7043.000000	7043.000000
mean	72850.579725	4.751384
std	28667.498530	2.691717
min	36314.000000	1.000000
25%	51788.000000	3.000000
50%	74142.000000	5.000000
75%	75720.000000	7.000000
max	175070.000000	10.000000

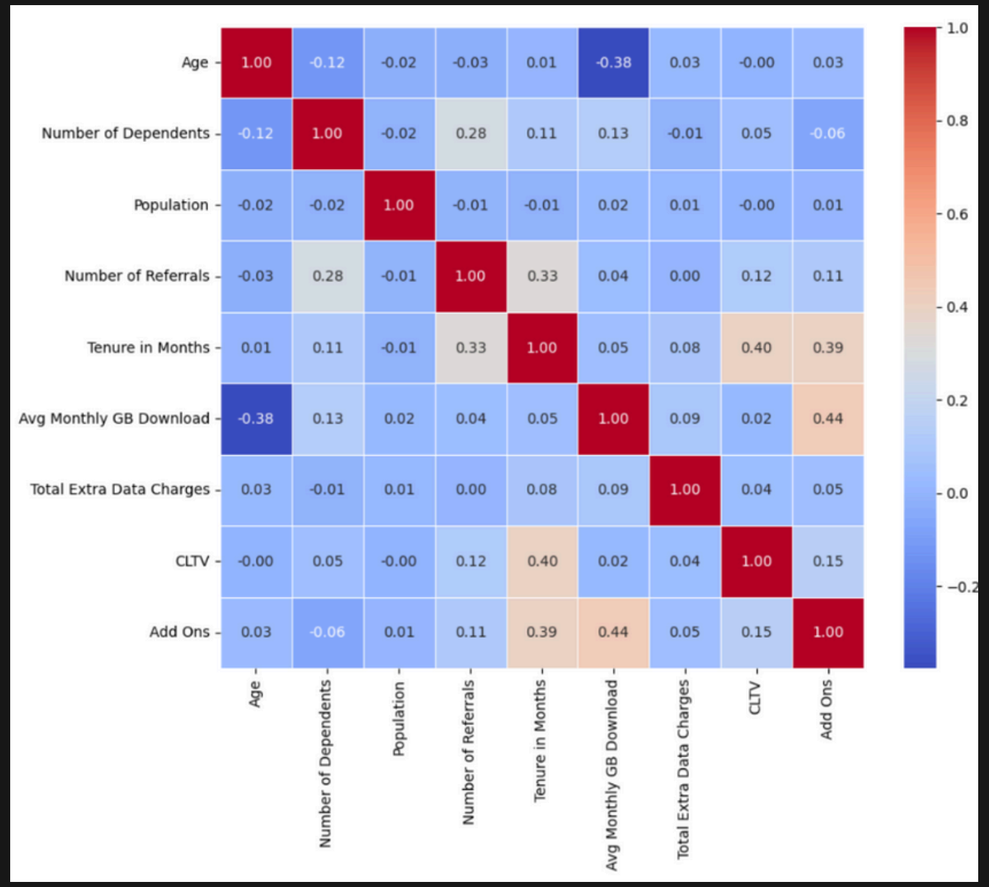


2. OUTLIERS

DESCRIPTIVE ANALYSIS

NUMERICAL VARIABLES

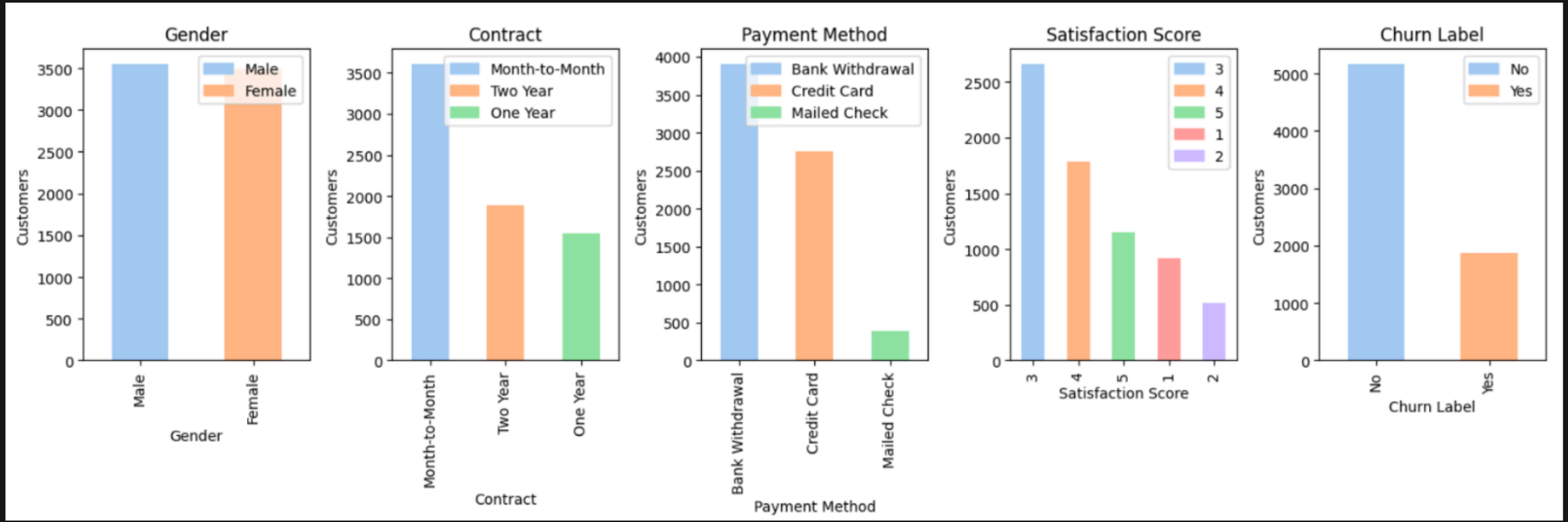
3. CORRELATION MATRIX



DESCRIPTIVE ANALYSIS

CATEGORICAL VARIABLES

BAR PLOT



MODEL APPROACH: CLASSIFICATION MODEL

KEY OBJECTIVE:

PREDICT CUSTOMER CHURN (QUALITATIVE TARGET VARIABLE).
PRIORITIZE IDENTIFYING CHURNERS (POSITIVE = 1) OVER NON-CHURNERS (NEGATIVE = 0).

REASONING:

ACQUIRING A CUSTOMER COSTS 5-7X MORE THAN RETAINING ONE

MODEL FOCUS

MINIMIZE FALSE NEGATIVES: MISCLASSIFYING CHURNERS AS NON-CHURNERS IS COSTLY.
EVALUATE MODEL USING RECALL (TRUE POSITIVE RATE):

FEATURE ENGINEERING

ADD ONS (SUM OF ALL SERVICES)

```
# adding variable of services subscribed
columns_to_convert = ['Phone Service', 'Internet Service',
                      'Online Security', 'Online Backup',
                      'Device Protection Plan',
                      'Premium Tech Support', 'Streaming TV',
                      'Streaming Movies', 'Streaming Music', 'Unlimited Data']
mapping = {'Yes': 1, 'No': 0, 'True': 1, 'False': 0}

add_ons = final_dataset[columns_to_convert].replace(mapping)
final_dataset['Add Ons'] = add_ons.sum(axis=1)
```

FROM COUNTIES TO REGIONS

```
county_to_region = {
    "northern california": [
        "trinity county", "lassen county", "kings county", "del norte county", "siskiyou county",
        "shasta county", "humboldt county", "tehama county", "modoc county", "plumas county",
        "sierra county", "mendocino county", "glenn county", "butte county", "colusa county",
        "sutter county", "yuba county", "lake county", "yolo county", "sacramento county",
        "placer county", "nevada county", "el dorado county", "amador county"
    ],
    "central california": [
        "merced county", "tulare county", "madera county", "kern county", "fresno county",
        "stanislaus county", "san joaquin county", "kings county", "calaveras county",
        "tuolumne county", "mariposa county", "san benito county"
    ],
    "bay area": [
        "marin county", "alameda county", "san mateo county", "contra costa county",
        "san francisco county", "santa clara county", "napa county", "sonoma county", "solano county"
    ],
    "southern california": [
        "los angeles county", "orange county", "san diego county", "ventura county",
        "riverside county", "san bernardino county", "imperial county"
    ],
    "coastal region": [
        "santa cruz county", "santa barbara county", "san luis obispo county", "monterey county"
    ],
    "eastern region": [
        "mono county", "inyo county", "alpine county"
    ]
}

county_to_region_flat = {}
for region, counties in county_to_region.items():
    for county in counties:
        county_to_region_flat[county] = region

final_dataset["Region"] = final_dataset["county"].map(county_to_region_flat)
```

```
# adding variable to indicate if lower income or not
import numpy as np
threshold = np.percentile(final_dataset['County Income Per Capita'].unique(), 25)
final_dataset['Low Income'] = (final_dataset['County Income Per Capita'] < threshold).astype(int).astype('object')
```

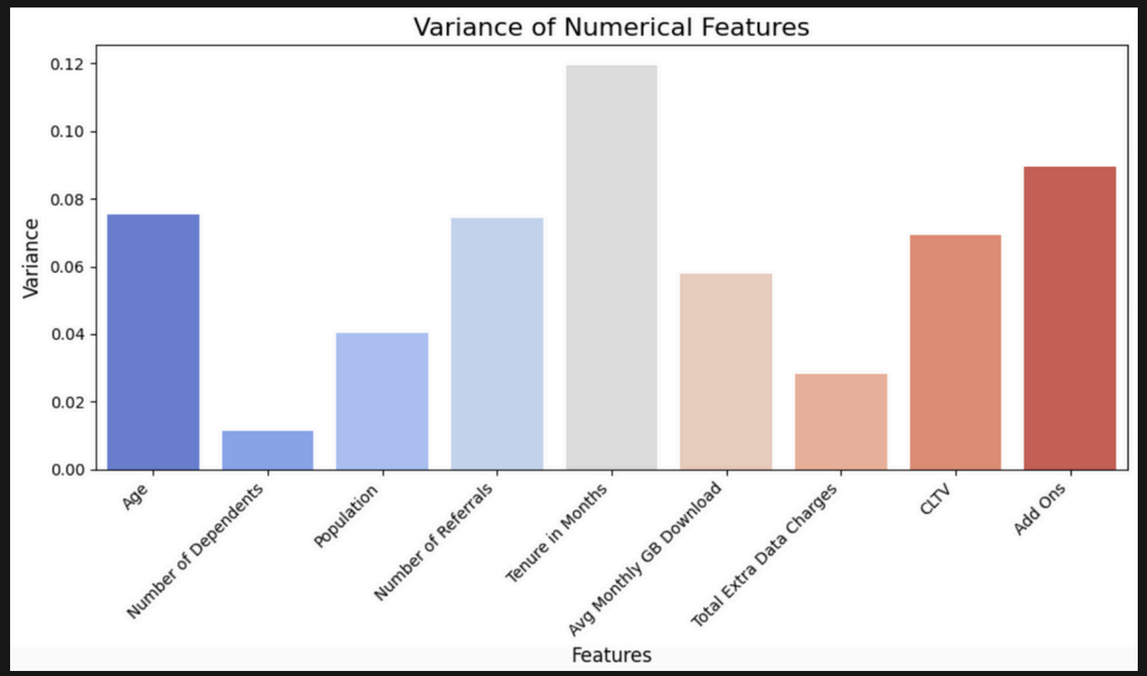
FROM COUNTY INCOME PER CAPITA TO LOW INCOME

FEATURE SELECTION

FILTER METHODS

1. VARIANCE

VARIANCETHRESHOLD FUNCTION FROM SKLEARN-FEATURE_SELECTION LIBRARY



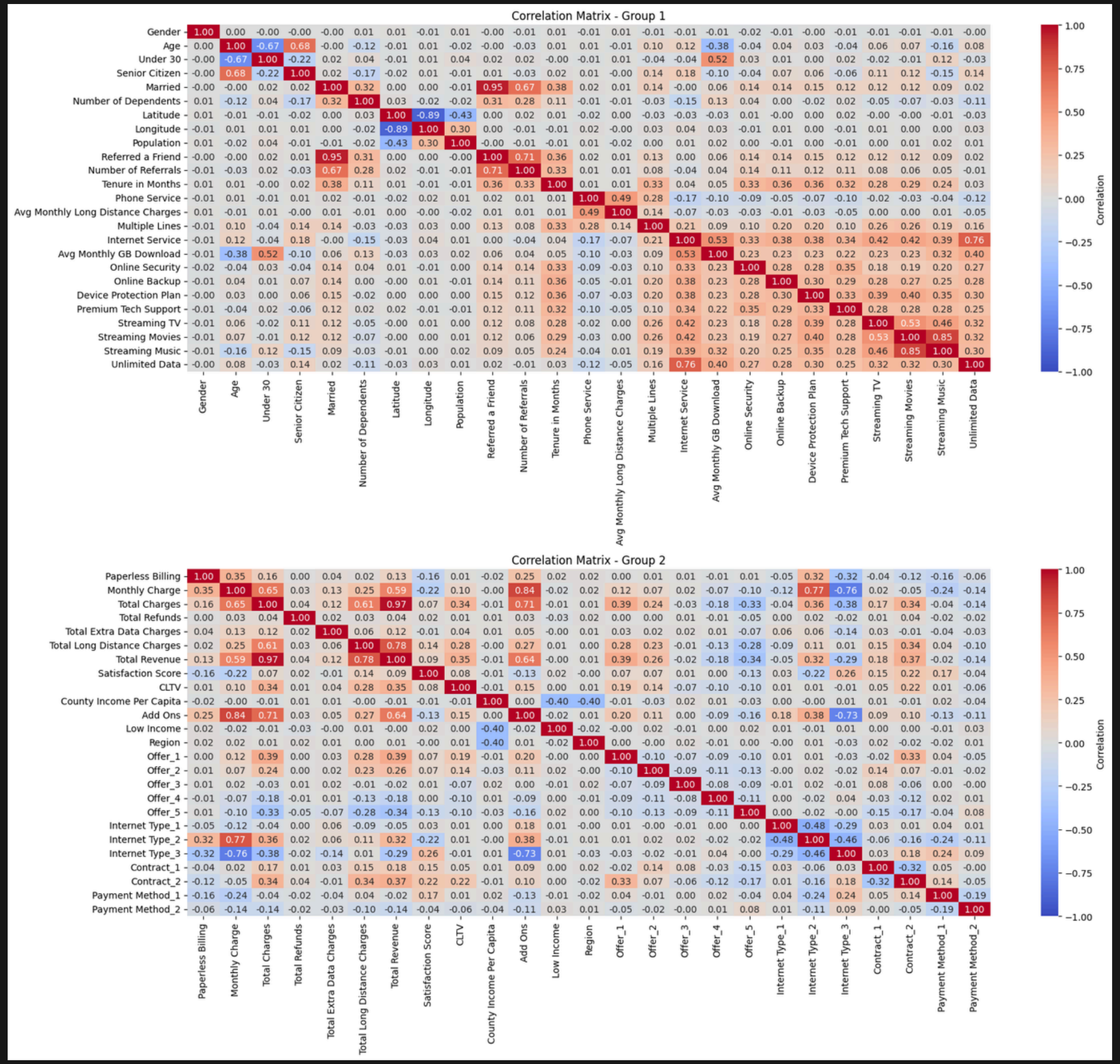

```
Variance of features:
 Age                          0.075392
Number of Dependents          0.011443
Population                    0.040366
Number of Referrals           0.074429
Tenure in Months              0.119466
Avg Monthly GB Download       0.057699
Total Extra Data Charges      0.028008
CLTV                          0.069200
Add Ons                       0.089436
dtype: float64
```

THRESHOLD = 0.01

FEATURE SELECTION

FILTER METHODS

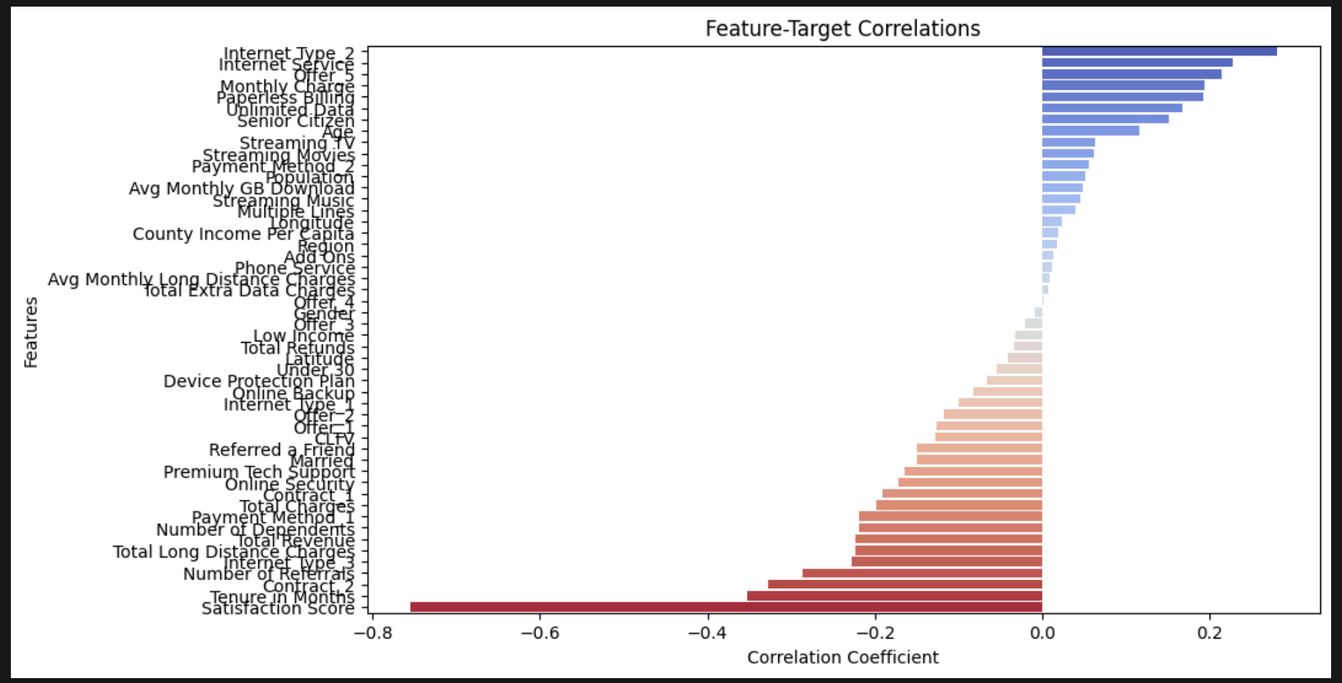
2. FEATURES-FEATURES CORRELATION



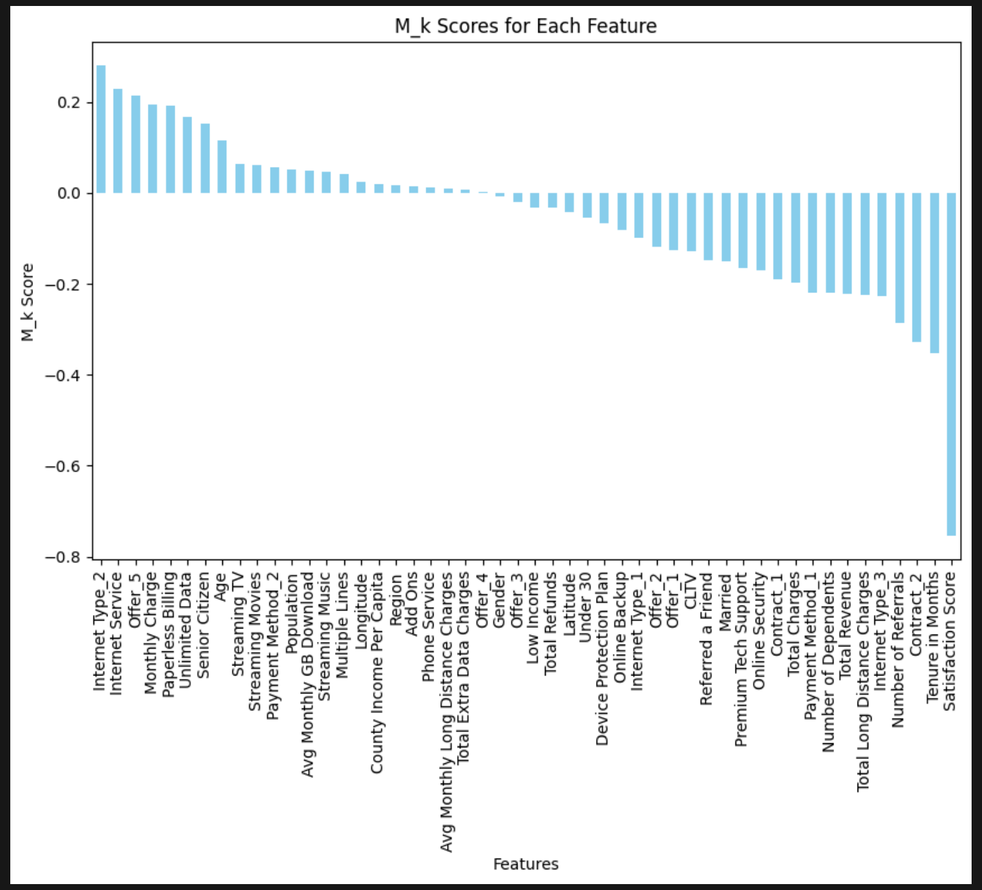
FEATURE SELECTION

FILTER METHODS

3. FEATURES-TARGET



4. $$M = \frac{k\overline{corr}_{cf}}{\sqrt{k+k(k-1)\overline{corr}_{ff}}}$$



FEATURE SELECTION

FILTER METHODS

5. MUTUAL INFORMATION

```
from sklearn.feature_selection import mutual_info_classif

X = final_dataset.loc[:, final_dataset.columns != 'Churn Label']
y = final_dataset['Churn Label']

mutual_info = mutual_info_classif(X, y)

mi_scores = pd.DataFrame({
    'Feature': X.columns,
    'Mutual Information': mutual_info
}).sort_values(by='Mutual Information', ascending=False)

print(mi_scores)
```

	Feature	Mutual Information
32	Satisfaction Score	0.412971
11	Tenure in Months	0.076486
10	Number of Referrals	0.073307
47	Contract_2	0.067436
26	Monthly Charge	0.051433
44	Internet Type_2	0.051415
35	Add Ons	0.050404
30	Total Long Distance Charges	0.048022
27	Total Charges	0.042595
31	Total Revenue	0.040274
15	Internet Service	0.035467
7	Longitude	0.033514
6	Latitude	0.031631
16	Avg Monthly GB Download	0.031536
5	Number of Dependents	0.031145
45	Internet Type_3	0.028774
46	Contract_1	0.027483
8	Population	0.025732
25	Paperless Billing	0.022842
48	Payment Method_1	0.022834
42	Offer_5	0.021800
17	Online Security	0.018217
20	Premium Tech Support	0.016921
38	Offer_1	0.014989
21	Streaming TV	0.013980
1	Age	0.012993
24	Unlimited Data	0.012676

RESAMPLING TECHNIQUE

1.

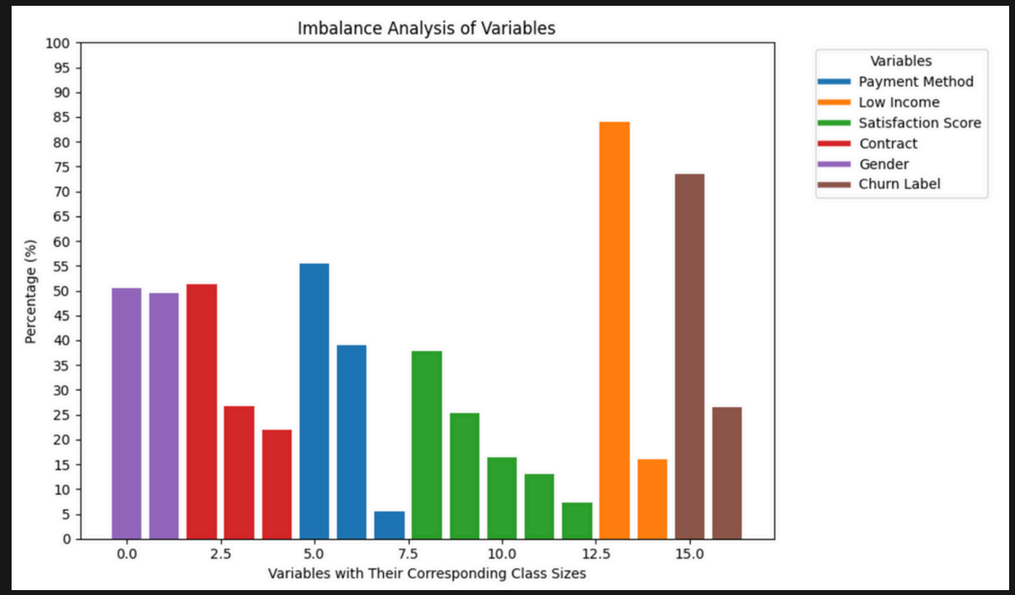
```python
# For this we are using the old dataframe (final_dataset before encoding) again - created a copy earlier
# Compute value counts and percentages for each column
columns_to_check = ['Gender', 'Contract', 'Payment Method', 'Satisfaction Score', 'Low Income','Churn Label']
summary = {}

for column in columns_to_check:
    # Convert column to string type to handle mixed types
    df_copy[column] = df_copy[column].astype(str)
    counts = df_copy[column].value_counts(normalize=True) * 100
    summary[column] = counts.to_dict()  # Convert to dictionary for easier handling in plain Python

# Create a summary table for percentages
percentages_table = []
for column, value_counts in summary.items():
    for value, percentage in value_counts.items():
        percentages_table.append({
            'Column': column,
            'Value': value,
            'Percentage': round(percentage, 2)
        })

# Display the percentages
print("Proportion Analysis of Variables:")
for entry in percentages_table:
    print(f"Column: {entry['Column']}, Value: {entry['Value']}, Percentage: {entry['Percentage']}%")
```

2.

```
Proportion Analysis of Variables:
Column: Gender, Value: Male, Percentage: 50.48%
Column: Gender, Value: Female, Percentage: 49.52%
Column: Contract, Value: Month-to-Month, Percentage: 51.26%
Column: Contract, Value: Two Year, Percentage: 26.74%
Column: Contract, Value: One Year, Percentage: 22.01%
Column: Payment Method, Value: Bank Withdrawal, Percentage: 55.5%
Column: Payment Method, Value: Credit Card, Percentage: 39.03%
Column: Payment Method, Value: Mailed Check, Percentage: 5.47%
Column: Satisfaction Score, Value: 3, Percentage: 37.84%
Column: Satisfaction Score, Value: 4, Percentage: 25.4%
Column: Satisfaction Score, Value: 5, Percentage: 16.31%
Column: Satisfaction Score, Value: 1, Percentage: 13.09%
Column: Satisfaction Score, Value: 2, Percentage: 7.35%
Column: Low Income, Value: 0, Percentage: 84.03%
Column: Low Income, Value: 1, Percentage: 15.97%
Column: Churn Label, Value: No, Percentage: 73.46%
Column: Churn Label, Value: Yes, Percentage: 26.54%
```

90% TO 10% RULE OF THUMB FOR THE IMBALANCE

3.



No significant imbalance detected.
Resampling not required as the class distribution reflects real industry data.

TRAIN/VALIDATION/TEST SPLIT

50 / 25 /25 SPLIT

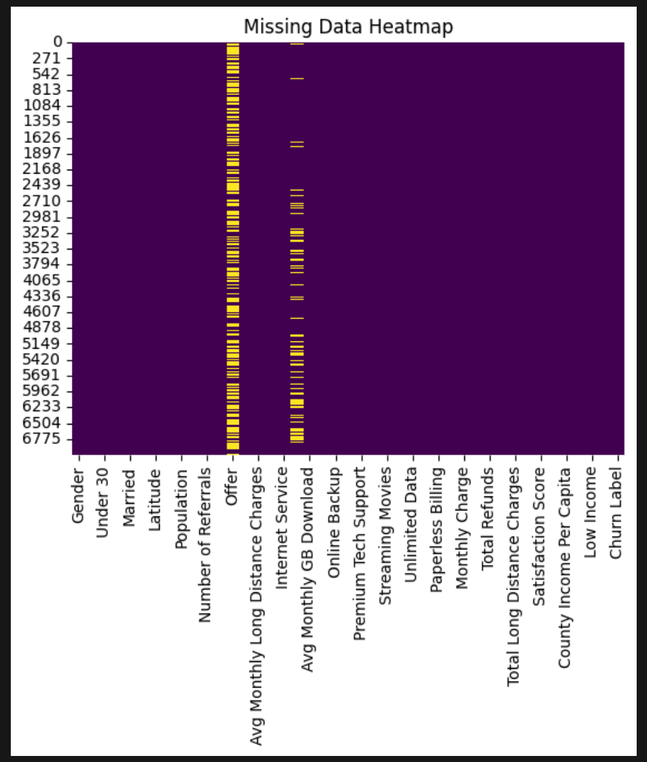
```python
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler

X = final_dataset.loc[:, final_dataset.columns != 'Churn Label']
y = final_dataset['Churn Label']

## SCALE DATASET
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# First, split into 50% training and 50% temporary (validation + test)
X_train, X_temp, y_train, y_temp = train_test_split(
    X, y, test_size=0.50, random_state=1234, shuffle=True, stratify=y
)

# Then, split the temporary set (50%) into 25% validation and 25% test
X_val, X_test, y_val, y_test = train_test_split(
    X_temp, y_temp, test_size=0.50, random_state=1234, shuffle=True, stratify=y_temp
)

print(f"Training set: {len(X_train)} rows")
print(f"Validation set: {len(X_val)} rows")
print(f"Test set: {len(X_test)} rows")
```

```
Training set: 3521 rows
Validation set: 1761 rows
Test set: 1761 rows
```

50%
25%
25%

PREPROCESSING STEPS

1. CHECKING FOR MISSING VALUES



2. MISSING VALUES SUM PER COLUMN

```
final_dataset.isnull().sum(axis=0)
```

Offer	3877
Internet Type	1526

3. IMPUTING NA

```
final_dataset["Internet Type"].fillna("No Internet", inplace=True) # Those with no internet service
final_dataset["Offer"].fillna("No marketing offer received", inplace=True) # Those with no marketing offer
```

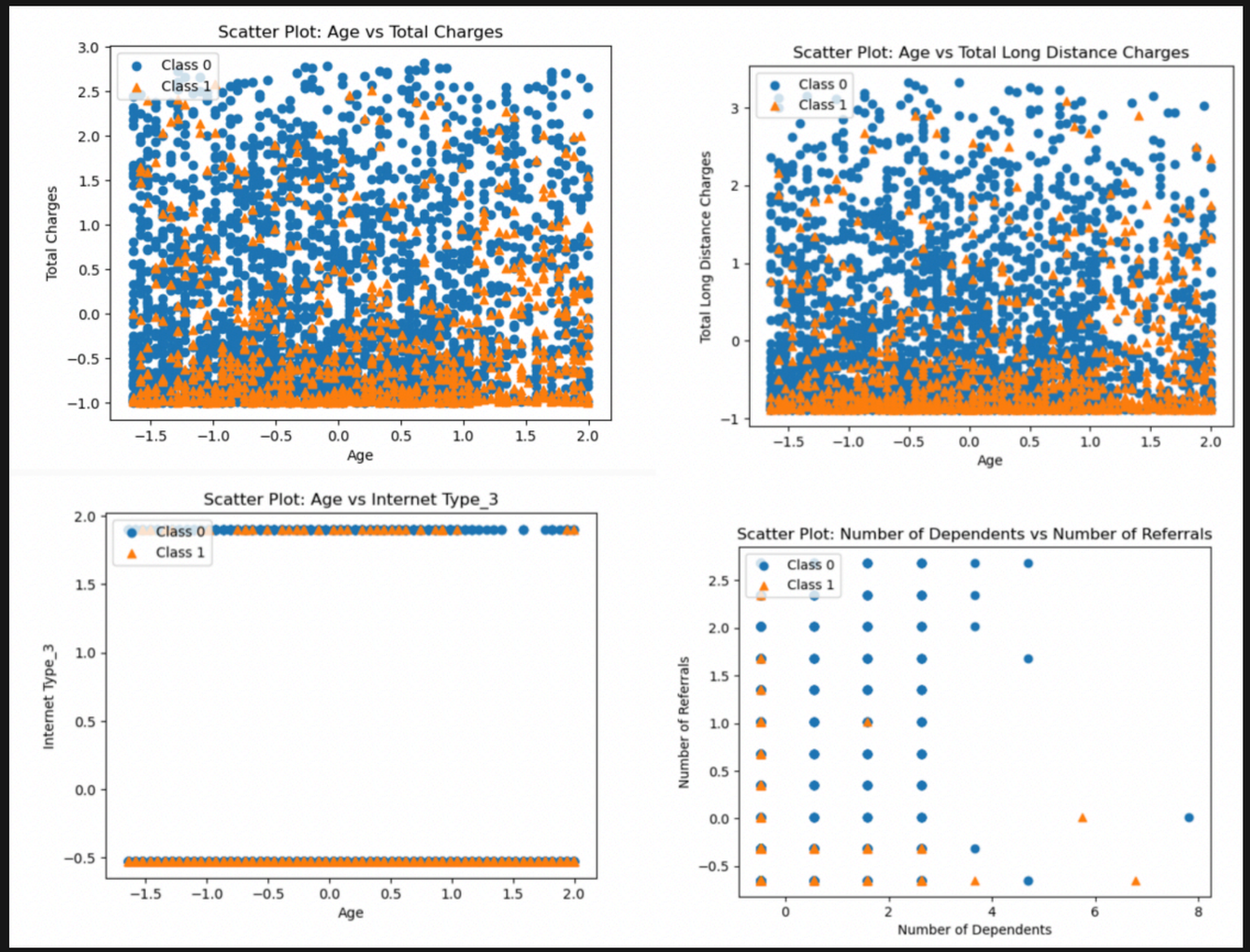
4. DROPPING VARIABLES

```
columns_to_drop = [
    "Customer ID", "Dependents", "Country", "State", "City", "Quarter",
    "Customer Status", "Churn Category", "Churn Reason", "county","Churn Score", "zip"
]
final_dataset = final_dataset.drop(columns=columns_to_drop, axis=1)
```

5. ENCODING

- *Transform nominal categorical columns into numerical values.*
- *Binary attributes encoded as 0 or 1.*
- *One-hot encoding used for non-ordinal features.*

LINEAR SEPARABILITY OF CLASSES



LOGISTIC REGRESSION

SEQUENTIAL FEATURE SELECTION (SCORING RECALL & 5 FOLDS)

```
# Initialize the Sequential Feature Selector with backward selection
sfs_forwards = SequentialFeatureSelector(log_reg_forwards, n_features_to_select='auto', direction='forward', scoring='recall', cv = 5)
# By setting cv = 5 the feature selector will use 5-fold cross-validation to evaluate the performance of different subsets of features at each
```

```
Selected features: ['Gender', 'Senior Citizen', 'Married', 'Number of Dependents', 'Longitude', 'Referred a Friend', 'Number of Referrals',
'Tenure in Months', 'Avg Monthly Long Distance Charges', 'Online Security', 'Online Backup', 'Device Protection Plan', 'Unlimited Data', 'Mon
thly Charge', 'Total Extra Data Charges', 'Total Long Distance Charges', 'Satisfaction Score', 'Region', 'Offer_1', 'Offer_2', 'Offer_5', 'In
ternet Type_2', 'Contract_1', 'Contract_2', 'Payment Method_1']
```

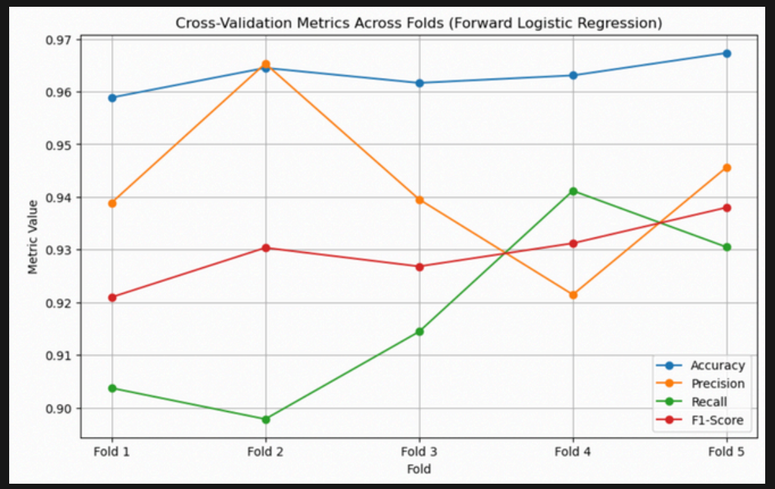
EVALUATING PERFORMANCE OF LOGISTIC REGRESSION MODEL

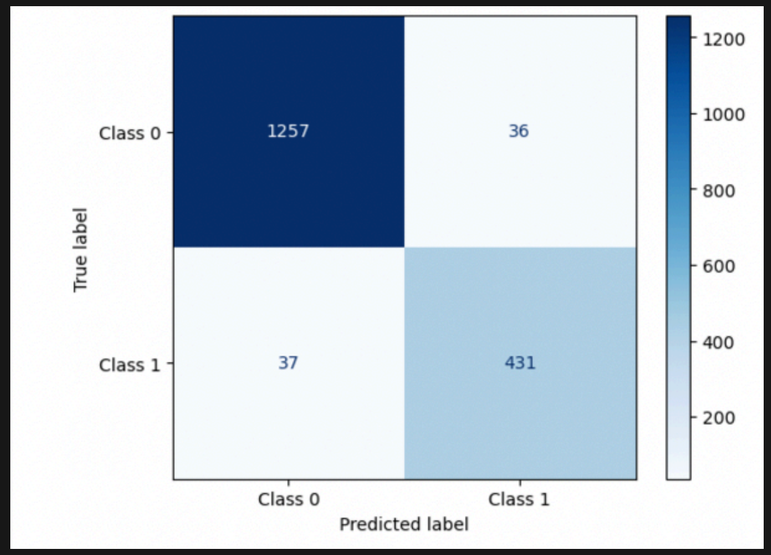
```
Metrics(y_train, y_train_pred_f, data='train')
```

```
The metrics for the train dataset are:
Precision: 0.949
Recall: 0.915
Accuracy: 0.964
F1 Score: 0.932
```

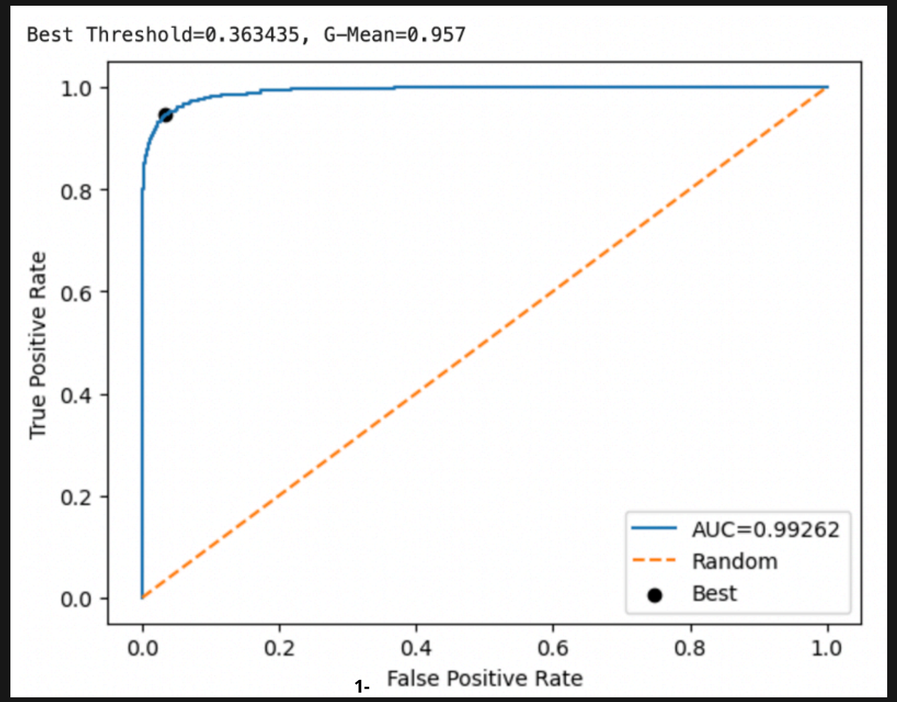
```
Metrics(y_test, y_test_pred_f, data='test')
```

```
The metrics for the test dataset are:
Precision: 0.923
Recall: 0.921
Accuracy: 0.959
F1 Score: 0.922
```





LOG REGR OPTIMISING THRESHOLD



$$G\text{-}Mean = \sqrt{TPR \times (1 - FPR)}$$

MAXIMISES SEPARABILITY SO IT IS GOOD AT IDENTIFYING POSITIVE (CHURN) AND NEGATIVE CLASS

LOWERING THRESHOLD- BETTER RECALL

$$\text{Recall} = \frac{\text{True Positive}}{\text{Predicted Results}}$$

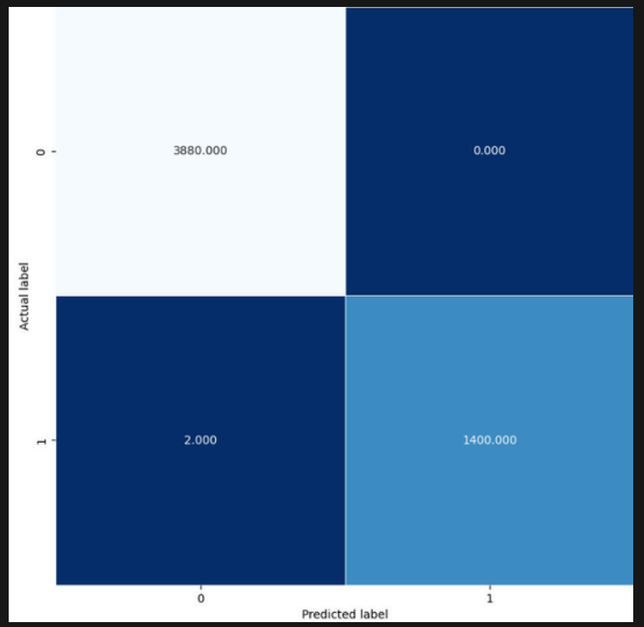
0.5 THRESHOLD → THRESHOLD 0.363435

```
Metrics(y_test, y_test_pred_f, data='test')

The metrics for the test dataset are:
Precision: 0.923
Recall: 0.921
Accuracy: 0.959
F1 Score: 0.922
```

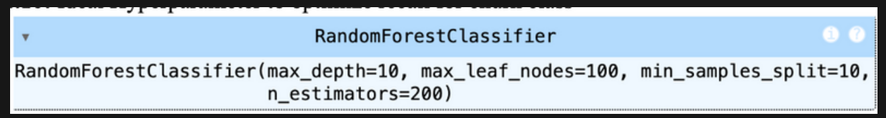
```
Metrics(y_test, y_test_pred_f_t, data='test')

The metrics for the test dataset are:
Precision: 0.887
Recall: 0.955
Accuracy: 0.956
F1 Score: 0.920
```

RANDOM FOREST

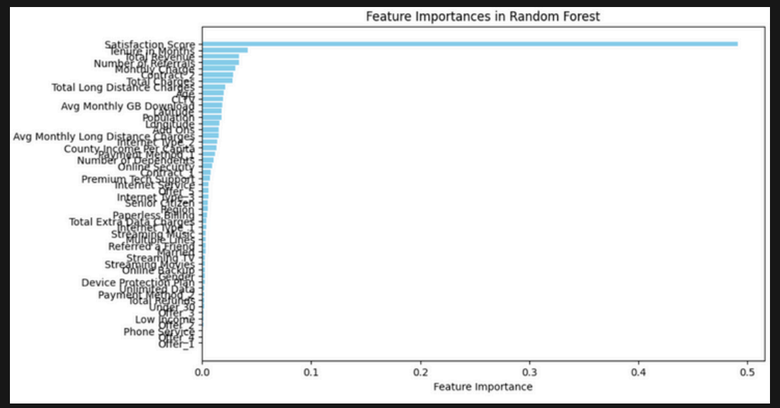
BASE MODEL WAS OVER FITTING



HYPERPARAMETER TUNING, TARGETING RECALL



FEATURE SELECTION



THE BEST MODEL

Random Forest Model	Logistic Regression Model
Model Evaluation Metrics (Test Set): Accuracy: 0.9625 Precision: 0.9653 Recall: 0.8910 F1-Score: 0.9267	The metrics for the test dataset are: Precision: 0.887 Recall: 0.955 Accuracy: 0.956 F1 Score: 0.920

Interpretability:

- Logistic Regression provides clear, quantifiable insights (odds ratios) for both categorical and numerical variables.
- Easier to understand decision paths compared to Random Forest, which is more complex (ensemble of decision trees) - no clear decision paths.

Handling Complexity:

- Logistic Regression handles the data well without overcomplicating the model (reasonable assumtion despite in graphs no clear separation of classes), avoiding unnecessary complexity seen in Random Forest when already clear results.

Fine-Tuning for Recall:

- Logistic Regression can be fine-tuned with lower thresholds to prioritize recall effectively and directly. Random forest would need more splits, finer partitions introducing more complexity to interpretability of the model.

HYPOTHESIS ANALYSIS

HYPOTHESIS 1:
STRONGLY SUPPORTED

HYPOTHESIS 2:
NOT SUPPORTED

HYPOTHESIS 3:
NOT SUPPORTED

HYPOTHESIS 4:
STRONGLY SUPPORTED

HYPOTHESIS 1

Monthly contracts lead to higher churn compared to yearly and bi-yearly contracts.

Churn likelihood reduces by 85.6% with one-year contracts and 96.7% with two-year contracts

Interaction used: NO

HYPOTHESIS 2

In very-low-income counties, women are more likely to churn; in higher-income counties, men are more likely to churn.

Gender has negligible impact, and females in very-low-income counties are only 5.1% less likely to churn than males.

Interaction used: YES

HYPOTHESIS 3

Mailed check payments are linked to higher churn rates compared to bank withdrawals or credit cards.

Customers paying via mailed checks are 21.9% less likely to churn compared to bank withdrawal.

Interaction used: NO

HYPOTHESIS 4

Customers with multiple subscriptions are more likely to churn if satisfaction drops below 3.

Low satisfaction increases churn likelihood 42x, especially for customers with multiple services (6.5x higher).

Interaction used: YES

H2

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1(\text{Gender}) + \beta_2(\text{Low Income}) + \beta_3(\text{Gender} \times \text{Low Income})$$

H4

$$\log\left(\frac{p}{1-p}\right) = \beta_0 + \beta_1(\text{Add Ons}) + \beta_2(\text{Low Satisfaction}) + \beta_3(\text{Add Ons} \times \text{Low Satisfaction})$$

CONCLUSION

As seen previously, the team decided to use the Logistic Regression model as it was more interpretable and yielded great results without having to increase complexity. This allows for an easier interpretation of hypothesis and application to the real-world scenario.

The team found out that shorter contracts are associated to higher churn rates and that customers subscribed to multiple services are more likely to churn when their satisfaction decreases. These results are useful for Telco companies to tailor their strategy and prevent loses.

Appendix Hypothesis

H1

Feature	Coefficient	Odds Ratio	Interpretation
Contract_1 (One Year)	-1.937	0.144	Customers with one-year contracts are 85.6% less likely to churn compared to monthly contracts.
Contract_2 (Two Year)	-3.401	0.033	Customers with two-year contracts are 96.7% less likely to churn compared to monthly contracts.

Appendix Hypothesis

H2

Feature	Coefficient	Odds Ratio	Interpretation
Gender (Female)	0.0018	1.001836	Gender alone has a negligible effect on churn; females have nearly the same odds of churn as males.
Low Income (Very-Low)	-0.097327	0.907259	Customers in very-low-income counties are 9.3% less likely to churn than those in higher-income counties.
Gender * Low Income Interaction	-0.052196	0.949143	Females in very-low-income counties are 5.1% less likely to churn compared to males in non-very-low-income counties.

Appendix Hypothesis

H3

Feature	Coefficient	Odds Ratio	Interpretation
Payment Method_2 (Mailed Checks)	-0.247	0.781	Customers paying via mailed checks are 21.9% less likely to churn compared to those using bank withdrawal.

Appendix Hypothesis

H4

Feature	Coefficient	Odds Ratio	Interpretation
Add Ons	-1.937	0.144	For each additional service subscribed, the likelihood of churn decreases significantly (about 85.6% less likely to churn), when considered independently.
Low Satisfaction	3.733	41.801	Indicates that customers with a satisfaction score below 3 are much more likely to churn, with the odds of churn being approximately 42 times greater for these customers compared to those with higher satisfaction scores.
Add Ons * Low Satisfaction	1.878	6.537	Suggests that when a customer subscribes to multiple services and has a low satisfaction score, the likelihood of churn increases substantially (by about 6.5 times) compared to those with low satisfaction but fewer services