Internet Downtime Prediction Analysis using Machine Learning

DATA 245 - Machine Learning Technologies

Group 6

Poojan Gagrani

Bhavik Patel

Kashish Thakur

Yuti Khamker

Introduction

- The Internet is a substantial part of our lives and every industry relies on the internet and its services to carry out daily tasks.
- Data Source for the purpose of the project, is owned by Mozilla, which is a telemetry dataset.
- The raw data is cleaned and preprocessed followed by the EDA to explore the correlation outliers in the dataset.
- Four different ML models are combined in an ensemble model to predict the internet outages and shutdowns around the world.

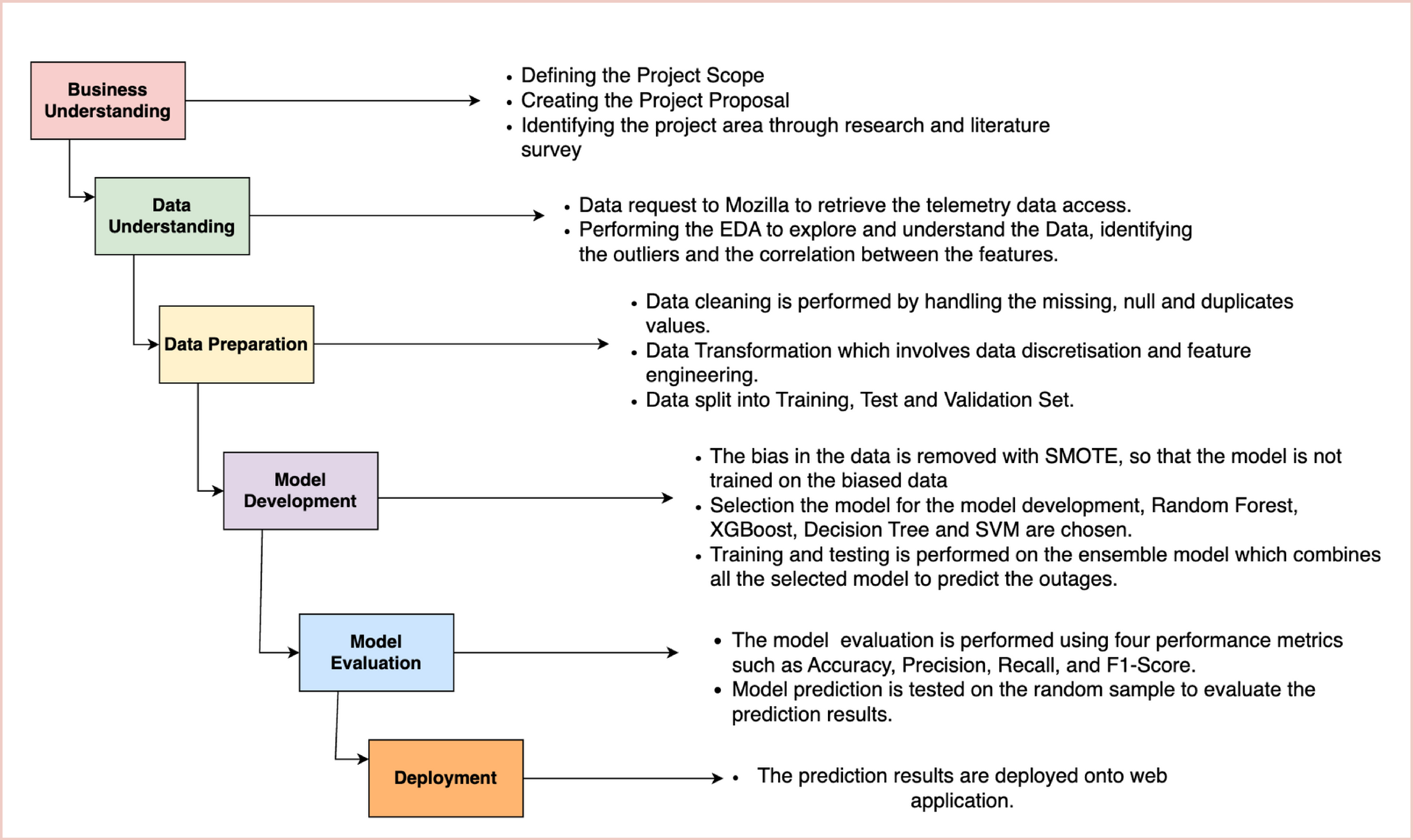
Motivation

The development and deployment of advanced machine learning models, can achieve enhanced accuracy in predicting outage durations. This not only helps to reduce downtime but also lowers costs for businesses and Internet service providers (ISPs).

Data-driven decision-making can benefit stakeholders by enabling preemptive solutions to network problems and enhanced service-level agreements (SLAs).

An improved user experience will result from shorter and more predictable disruptions.

Methodology

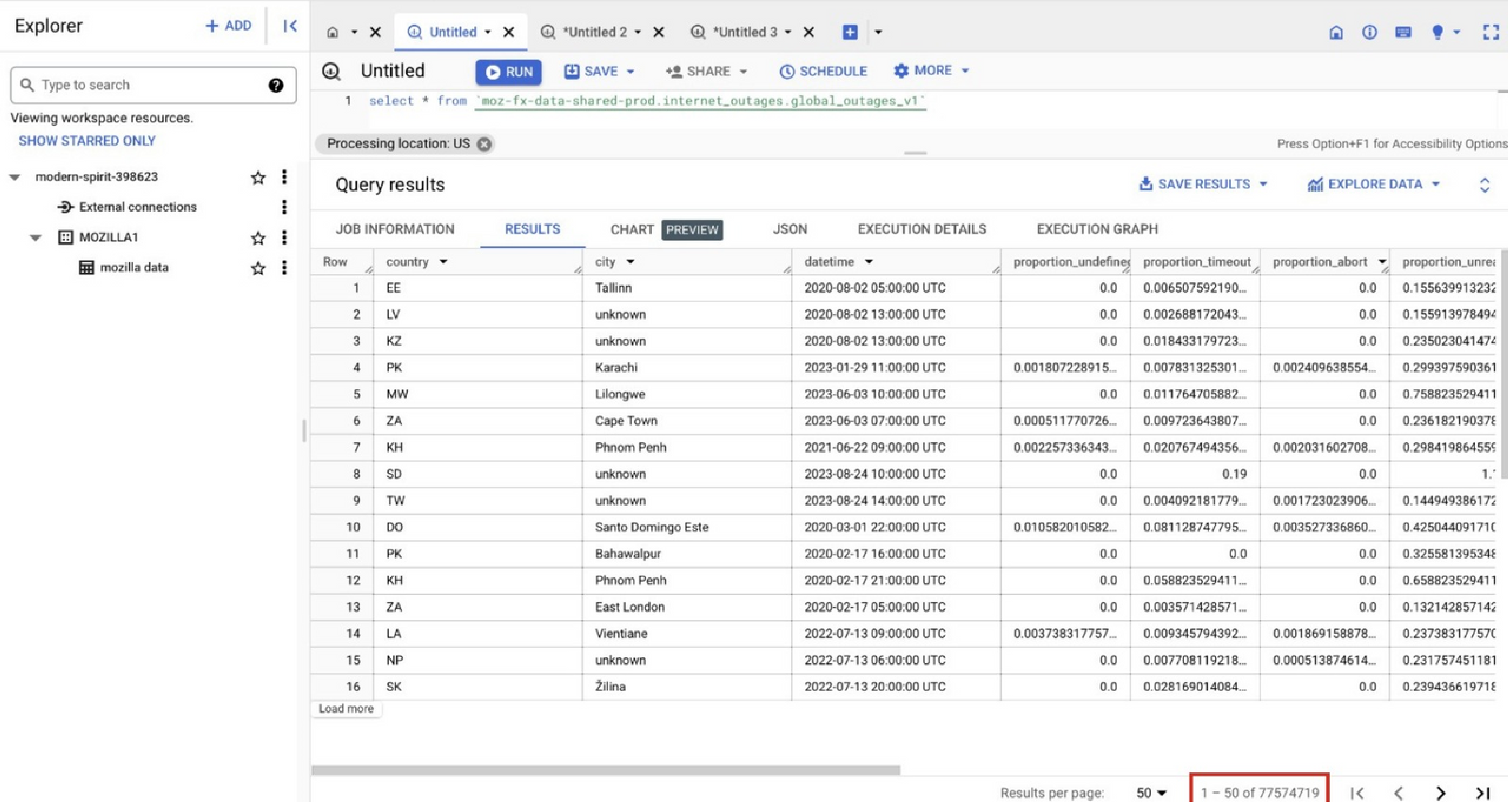


Literature Survey

Name	About the Dataset	Model Used	Results
Yadwad et al. (2022) Fault Prediction for Network Devices Using Service Outage Prediction Model.	IDM Dataset as been used which contains information about Customer trouble tickets.	Bayesian networks Model and Hidden Markov model	The results imply that Hidden Markov Models are particularly strong as using this probabilistic method improves prediction accuracy when compared to other prediction techniques.
Xu, Yu et al. (2021) Intelligent Outage Probability Prediction for Mobile IoT Networks Based on an IGWO-Elman Neural Network.	IOT sensor Data is collected through Monte-Carlo simulation	IGWO-Elman algorithm	The prediction accuracy of IGWO-Elman is better than algorithms.
Basikolo et al. (2023) Towards zero downtime: Using machine learning to predict network failure in 5G and beyond.	The dataset used is from commerical metwork testbed.	Random Forest Regressor, Support Vector Regression (SVR), Bayesian ridge and their proposed SVR.	The SVR model, proposed in this study, can swiftly predict the occurrence of a network failure event within the next ten minutes with an f1-score exceeding 0.9 in just ten seconds.
Chen et al. (2019) Outage Prediction and Diagnosis for Cloud Service Systems	The outage dataset is collected from a Microsoft cloud system	Bayesian network and Gradient Boosting Tree based classification model	Using the AirAlertFull the Precision is 71.11,Recall is 100.00, and F1-Score is 83.17

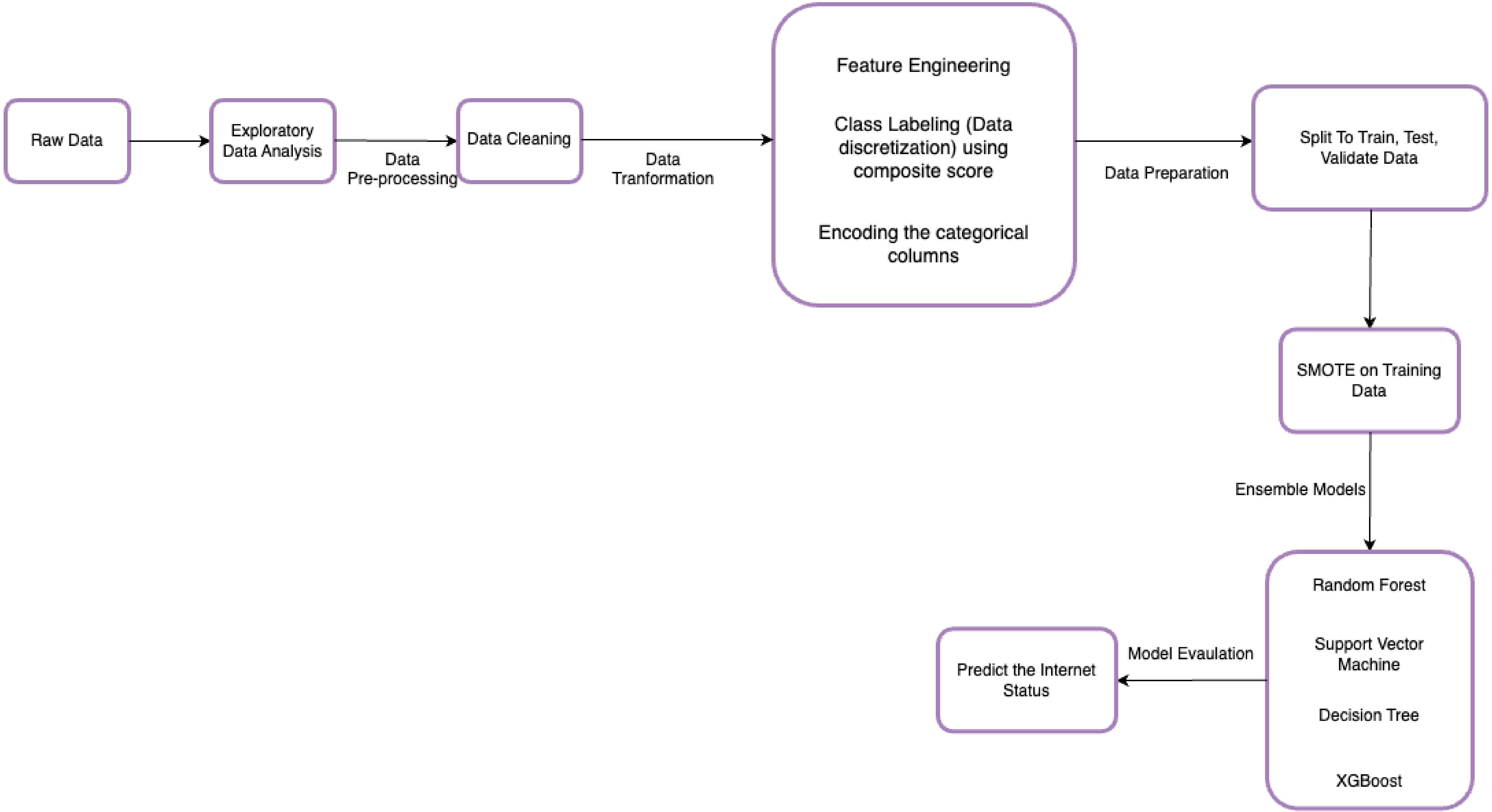
Data Source

- The source of data is the telemetry dataset owned by Mozilla.
- It consists of the aggregated metrics that correlates to internet outages for different countries in the world.
- The data is available in the Big Query, which is used as the repository where data is fetched and accessed using Google Colab.

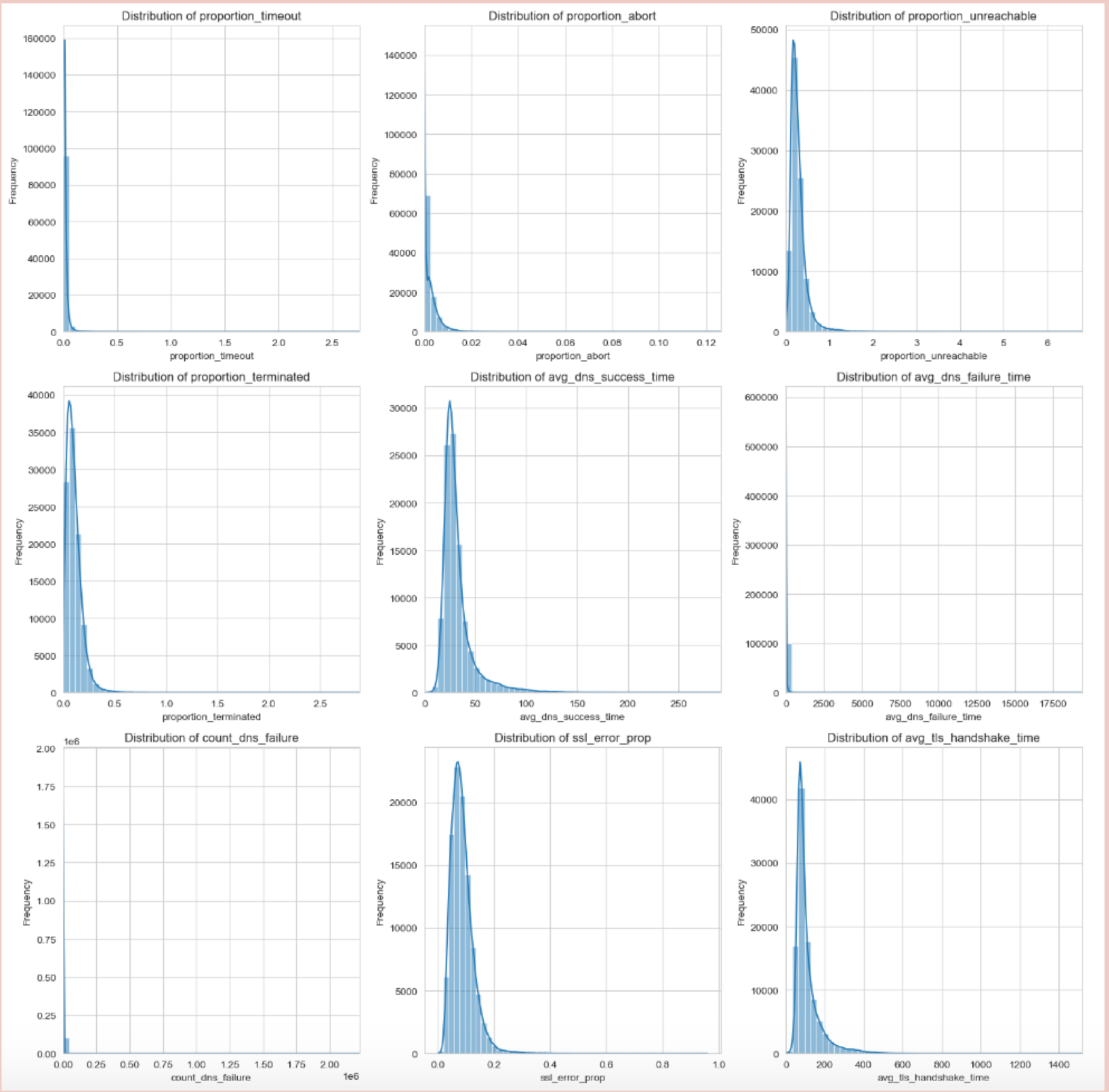


Data Source URL:https://wiki.mozilla.org/Mozilla_Network_Outages_Data_Project

Project Data Flow



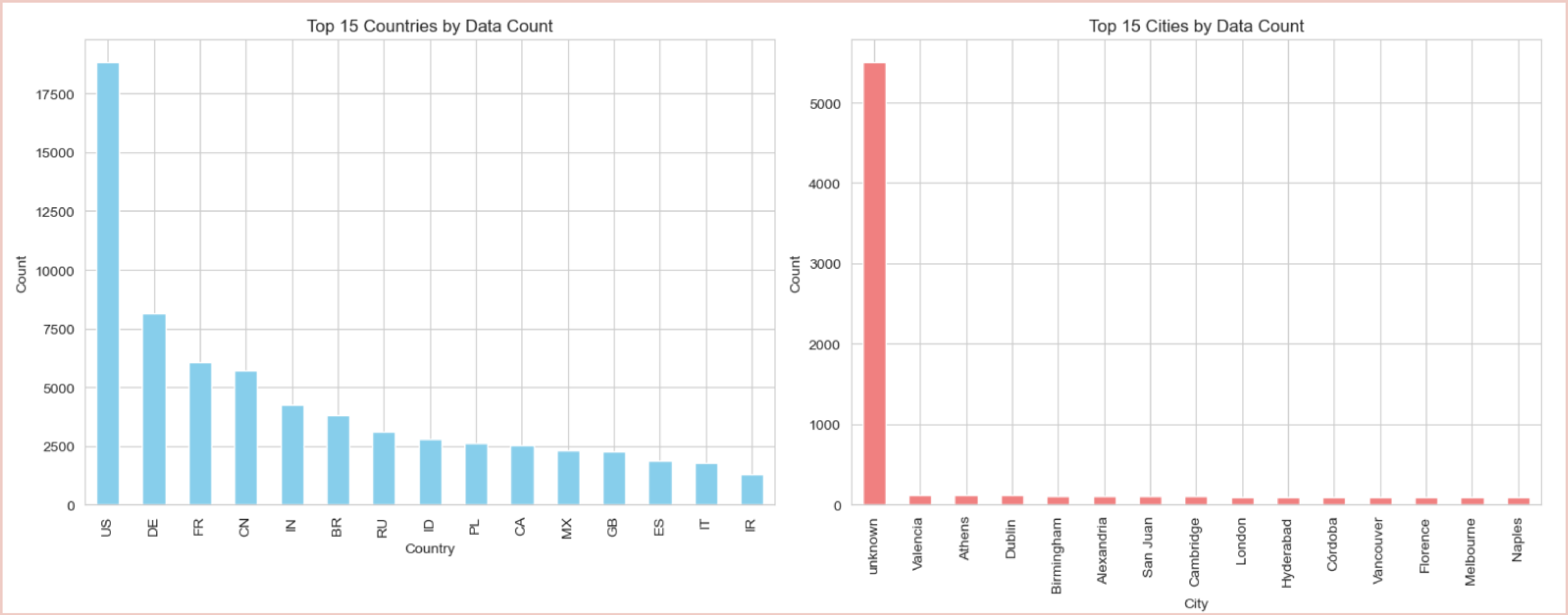
Exploratory Data Analysis



Identifying the Target Feature

- The histogram depicts the distribution of continuous features.
- Most values range between 0 and 1, with possible outliers exceeding 1.

Exploratory Data Analysis



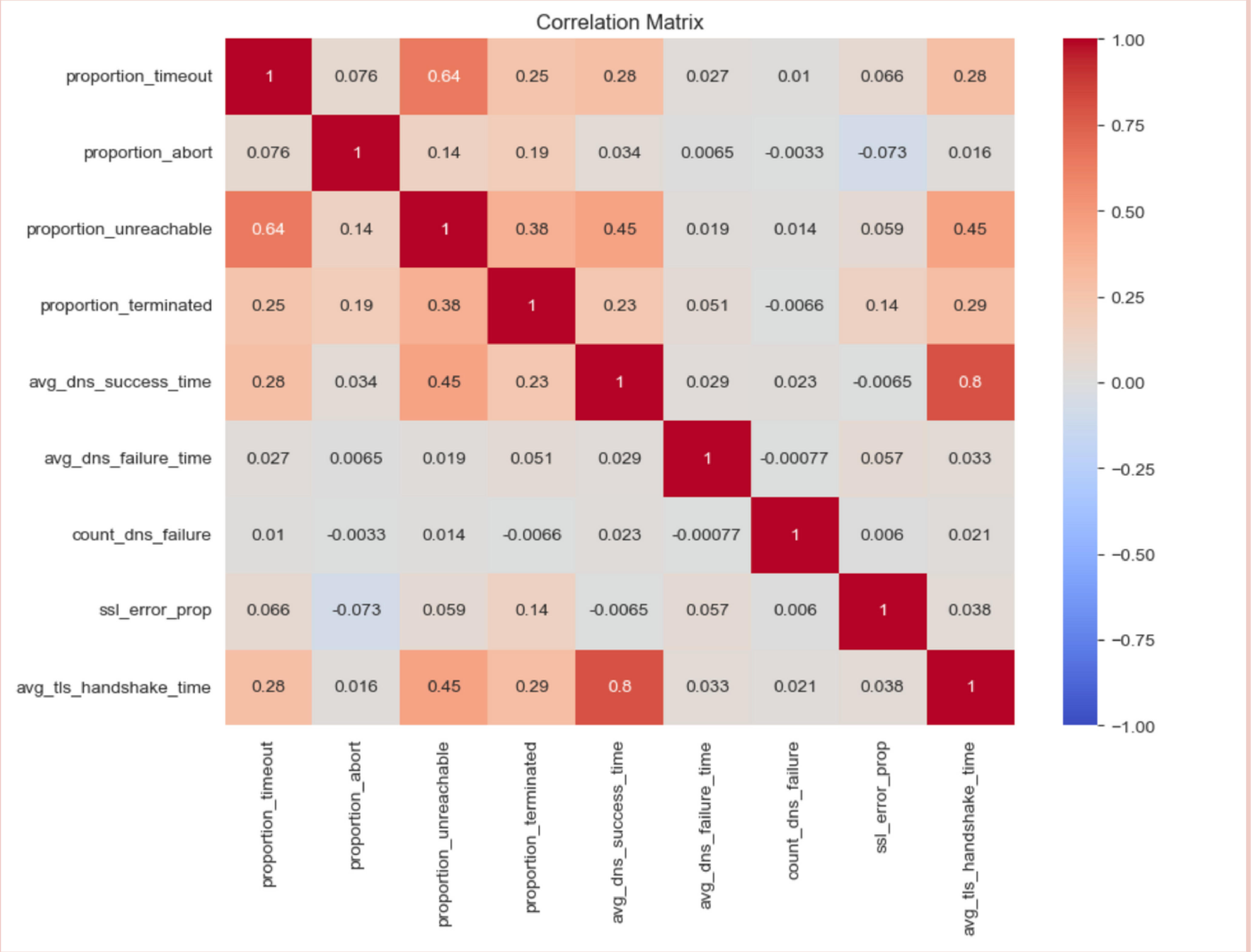
Identifying the most common country and city in the Dataset

To analyse the categorical features in the dataset, the plot displays the count of datapoints of top 15 countries and cities

Exploratory Data Analysis

Finding the correlation between the numerical features

- The heat-map above is used to understand the correlation between continuous features in the dataset.
- We can see that most of the features have positive correlation.



Data Cleaning

```
df.isnull().sum()

country                    246
city                         0
datetime                     0
proportion_undefined         0
proportion_timeout           0
proportion_abort             0
proportion_unreachable       0
proportion_terminated        0
proportion_channel_open      0
avg_dns_success_time         0
missing_dns_success          1
avg_dns_failure_time        10
missing_dns_failure          1
count_dns_failure          295
ssl_error_prop               1
avg_tls_handshake_time     696
dtype: int64
```

Identified Null Values

```
[ ] df.isnull().sum()

country                    0
city                       0
datetime                   0
proportion_undefined       0
proportion_timeout         0
proportion_abort           0
proportion_unreachable     0
proportion_terminated      0
proportion_channel_open    0
avg_dns_success_time       0
missing_dns_success        0
avg_dns_failure_time       0
missing_dns_failure        0
count_dns_failure          0
ssl_error_prop             0
avg_tls_handshake_time     0
dtype: int64
```

Removed Null Values

	country	city	datetime	proportion_undefined	proportion_timeout	proportion_abort
42	MM	unknown	2023-03-15 05:00:00+00:00	0.000000	0.009988	0.002497
43	MU	unknown	2023-03-15 10:00:00+00:00	0.000000	0.022642	0.007547
51	NL	unknown	2023-03-15 01:00:00+00:00	0.000000	0.008701	0.000322
54	NZ	unknown	2023-03-15 06:00:00+00:00	0.000000	0.011407	0.007605
98	IR	unknown	2023-02-12 03:00:00+00:00	0.000220	0.049934	0.002782
...	...	...	...	...	...	...
11905423	BE	unknown	2023-07-12 21:00:00+00:00	0.000000	0.008821	0.002786
11905447	CY	unknown	2023-07-12 06:00:00+00:00	0.000000	0.000000	0.000000
11905474	CD	unknown	2023-09-14 21:00:00+00:00	0.000000	0.052632	0.000000
11905497	BG	unknown	2023-06-06 06:00:00+00:00	0.000562	0.005807	0.000375
11905511	CH	unknown	2023-06-06 23:00:00+00:00	0.000000	0.010029	0.001910

649844 rows × 16 columns

Handling unkown values

The identified unknown values are not dropped as it consist of legitimate data values

Data Transformation and Model building

Feature Engineering
Date Time column converted to time_slot column

```
def get_detailed_time_slot(hour):
    if 0 <= hour < 6:
        return 'Late Night'
    elif 6 <= hour < 9:
        return 'Early Morning'
    elif 9 <= hour < 12:
        return 'Late Morning'
    elif 12 <= hour < 15:
        return 'Early Afternoon'
    elif 15 <= hour < 18:
        return 'Late Afternoon'
    elif 18 <= hour < 21:
        return 'Early Evening'
    else:
        return 'Night'
```

Data Discretisation,
Creating target feature quality_label

```
features = ['proportion_timeout', 'proportion_unreachable',
            'proportion_terminated',
            'avg_dns_failure_time', 'count_dns_failure']

# Create a composite score as a simple sum of standardized fe
df['composite_score'] = df[features].apply(lambda x:
                          (x - x.mean()) / x.std()).sum(axis=1)

# Calculate the quantiles on this composite score
quantiles = df['composite_score'].quantile([0.25, 0.5, 0.75])

# Define the labeling function with the correct quartile valu
def label_quality(score, quantiles):
    if score <= quantiles[0.25]:
        return 'good'
    elif score <= quantiles[0.50]:
        return 'moderate'
    elif score <= quantiles[0.75]:
        return 'bad'
    else:
        return 'worse'
```

Label Encoding
Value of categorical features converted to numerical values

```
our DataFrame
= label_encoder_country.fit_transform(df['country'])
abel_encoder_city.fit_transform(df['city'])

is a categorical variable that you want to encode ordin
| = ordinal_encoder_time_slot.fit_transform(df[['time_sl
```

Recursive Feature Elimination
Using the RFE to find the important descriptive features

```
[('proportion_timeout', 1), ('propor
1), ('avg_dns_failure_time', 1), ('c
hour', 1), ('composite_score', 1), (
d', 4), ('missing dns failure', 5),
```

Standard Scaler
Using the scalar technique to scale the numerical features

Data Preparation
Transformed data is split into Train, Validation and Test Data

```
# Features are separated: numerical features that need scaling and categorical encoded features that dor
numerical_features = ['proportion_timeout', 'proportion_unreachable', 'proportion_terminated',
                      'avg_dns_success_time', 'avg_dns_failure_time', 'count_dns_failure', 'ssl_error_pr
categorical_features = ['country_encoded', 'city_encoded', 'time_slot_encoded']
X_numerical = df[numerical_features]
X_categorical = df[categorical_features]

# Target variable
y = df['quality_label_encoded']

# Split the data into train+validate and test sets (90-10 split)
X_temp_num, X_test_num, y_temp, y_test = train_test_split(X_numerical, y, test_size=0.1, stratify=y, rar
X_temp_cat, X_test_cat = train_test_split(X_categorical, test_size=0.1, random_state=42, shuffle=True)

# Further split the train+validate into train and validate sets (89-11 split, approximates to 80-10 of c
X_train_num, X_validate_num, y_train, y_validate = train_test_split(X_temp_num, y_temp, test_size=1/9, s
X_train_cat, X_validate_cat = train_test_split(X_temp_cat, test_size=1/9, random_state=42, shuffle=True)
scaler = StandardScaler()

# Fit the scaler on the numerical part of the training data and transform
X_train_num_scaled = scaler.fit_transform(X_train_num)
X_validate_num_scaled = scaler.transform(X_validate_num)
X_test_num_scaled = scaler.transform(X_test_num)
```

SMOTE
Applied on the training data to remove the bias in the data

```
0      199981
1      199981
3      199981
2      199981
```

Ensemble Modeling
Decision tree + Random Forest + SVM + XGBoost

Application

Model Comparision

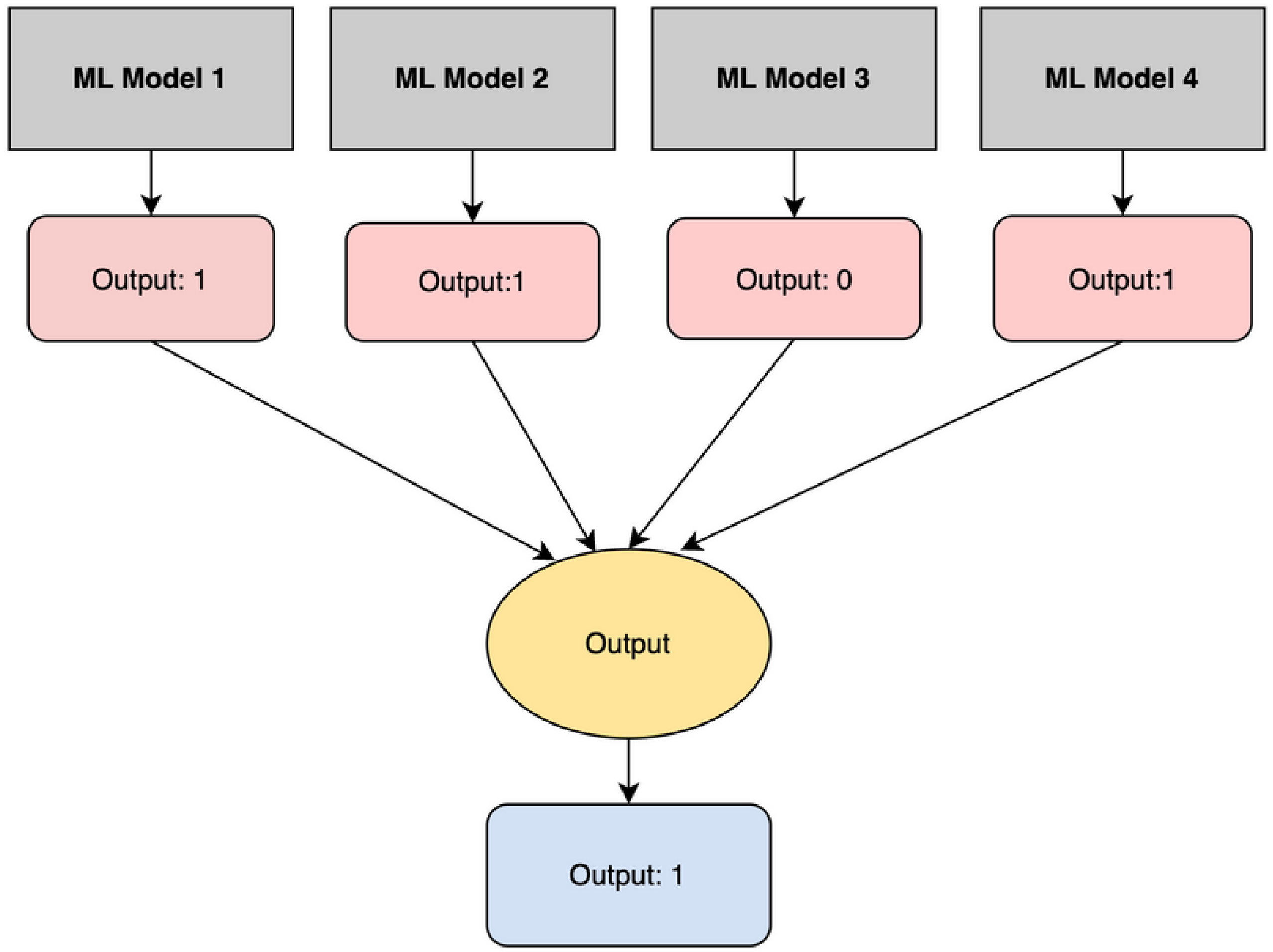
Models	Accuracy	Precision	F1 Score	Training TIme
Random Forest	96%	98%	98%	30 minutes
Support Vector Machine	71.8%	72.3%	72.3%	45 minutes
Decision Tree	93.42%	96%	96%	15 minutes
XGBoost	97%	99%	99%	25 minutes
Ensemble Model	96.28%	96%	96%	20 minutes

Ensemble Model

- The ensemble model is created by combining the four machine learning algorithm which are Support Vector Machine, Random Forest, Decision Tree, and XGBoost Algorithm.

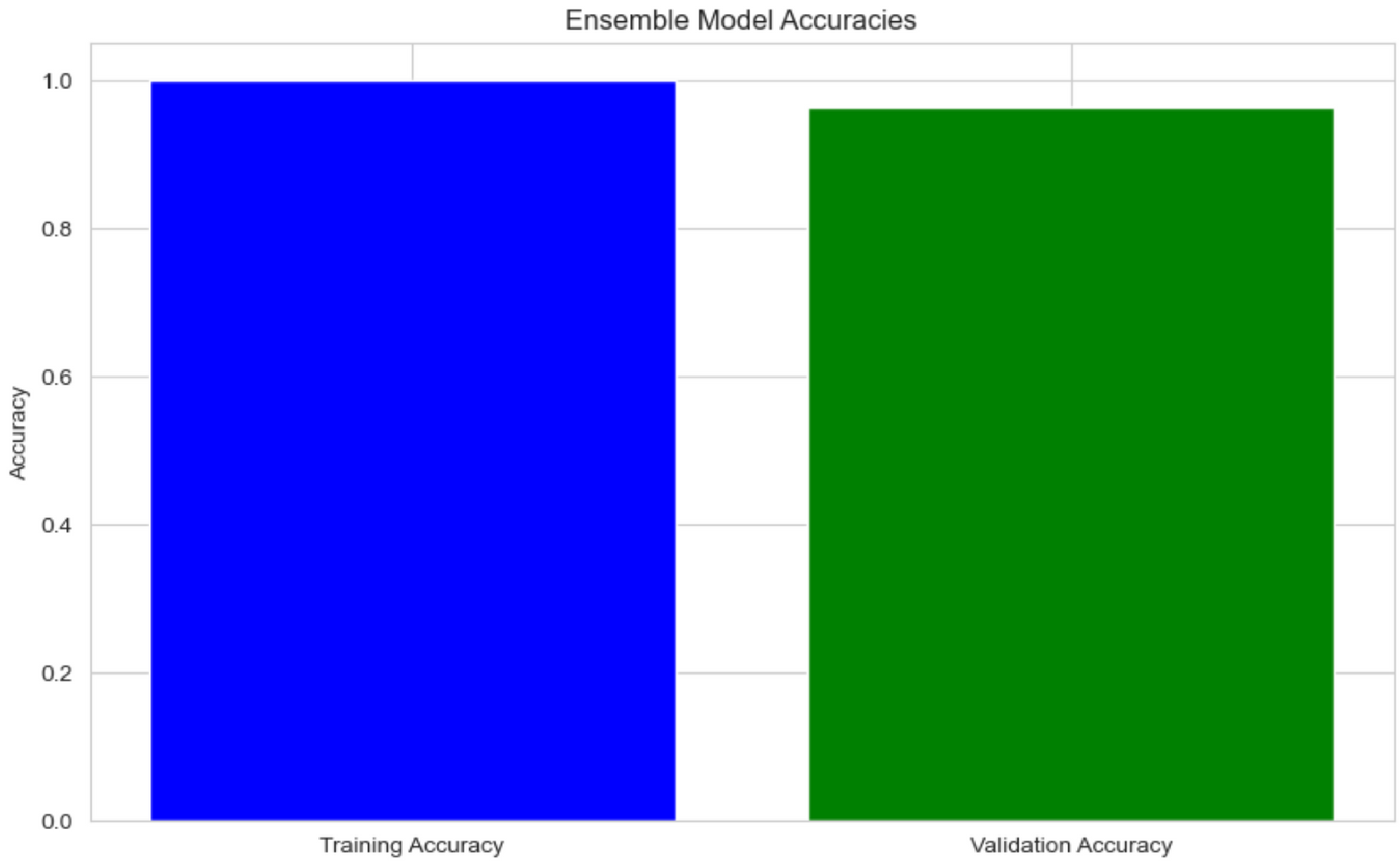
- The prediction from the all the combine algorithm is integrated to generate the final outcome.

- The ensemble model uses a Hard voting technique that performs the prediction based on the majority voting.



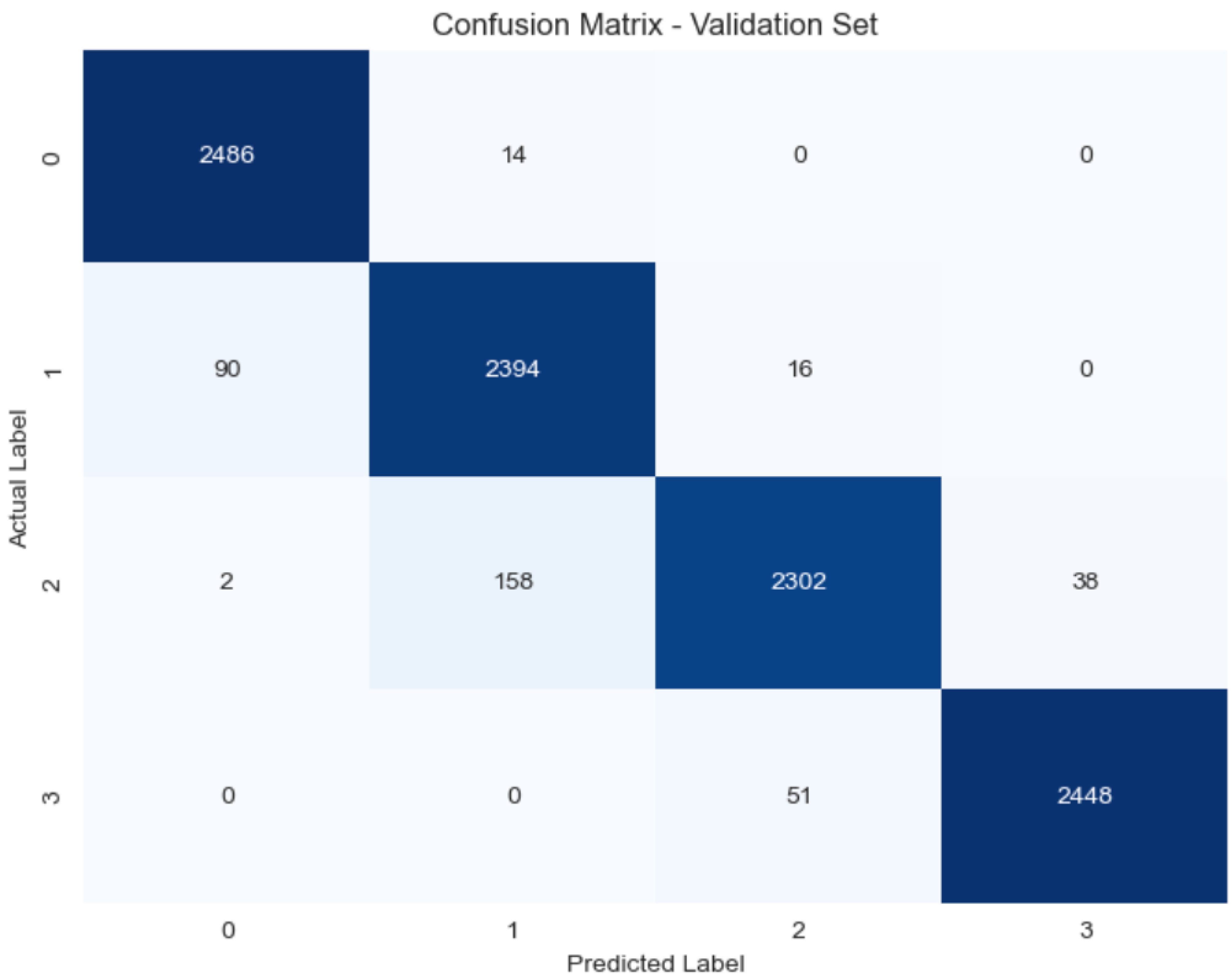
Depiction of Hard Voting in Ensemble Model

Model Performance (cont.d)



Plotting Training and Validation accuracies

Model Performance



The confusion matrix shows the majority of the labels that are predicted were actually from that labeled class

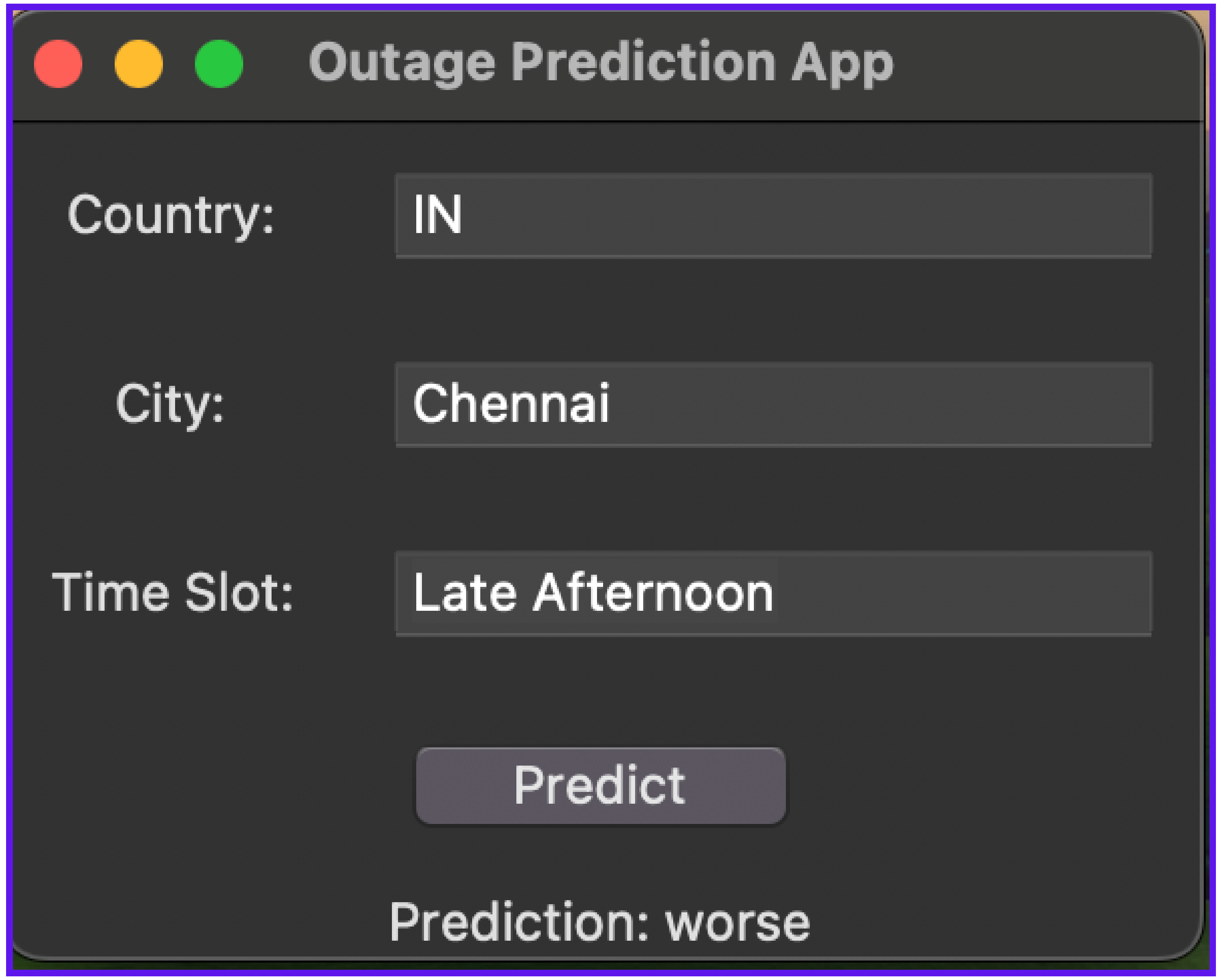
Conclusion

By precisely forecasting internet outages using spatial and temporal data, this project greatly improves network management and holds out the prospect of better service reliability and customer satisfaction in the telecom industry. By addressing present network management issues, the project lays foundations for upcoming improvements in service quality and predictive maintenance.

Future Scope

- Incorporating real-time data feeds to update the predictions dynamically.
- Conducting a cost-benefit analysis to understand the financial impact of outages.
- Develop models to detect anomalies in network traffic that could indicate potential issues leading to outages.

Deployment



Application which accepts Country, City and Time slot as parameter and predicts the outage

Thank you!

Q & A