SC1015 MINI PROJECT: EARTHQUAKE PREDICTION AND ANALYSIS

By: Rohan Ramakrishna (U2323174B)
Dhruv Agarwal (U2323494L)

OVERVIEW

01 — Problem Statement and Practical Motive	02 — Dataset analysis	03 — Data preparation
04 — Visualisation	05 — ML Models	06 — Results

PROBLEM STATEMENT

To determine the most effective time forecasting or regression model for predicting major earthquakes’ magnitude and the most effective classification model to predict the RANGE of major earthquakes’ magnitudes based on geographical coordinates (latitude/longitude) and depth.

PRACTICAL MOTIVATION

01 Risk Management: To identify of high-risk areas, aiding in urban planning and infrastructure development and risk mitigation.

02 Preparedness: Predicting the magnitude of Earthquakes using data collected from a big timeline can help in the spreading of precautionary measures to the general public.

03 Advancing Science: Contributes to broader scientific understanding of Earth's dynamic processes and seismic activity.

DATASETS

The two datasets we've used are:

1) Earthquakes with magnitude greater than 5.5- This dataset includes a record of the date, time, location, depth, magnitude, and source of every earthquake with a reported magnitude 5.5 or higher from 1965 to 2016

2) Tectonic plates boundary data- This data contains latitude and longitude data that completely encloses 56 tectonic plates.

SAMPLE COLLECTION

	Date	Time	Latitude	Longitude	Type	Depth	Depth Error	Depth Seismic Stations	Magnitude
0	01/02/1965	13:44:18	19.246	145.616	Earthquake	131.6	NaN	NaN	6.0
1	01/04/1965	11:29:49	1.863	127.352	Earthquake	80.0	NaN	NaN	5.8
2	01/05/1965	18:05:58	-20.579	-173.972	Earthquake	20.0	NaN	NaN	6.2
3	01/08/1965	18:49:43	-59.076	-23.557	Earthquake	15.0	NaN	NaN	5.8
4	01/09/1965	13:32:50	11.938	126.427	Earthquake	15.0	NaN	NaN	5.8

Tectonic plate data

DATA CLEANING

1) Removing of columns with null values
2) Correction of incorrect date values
3) Correction of incorrect time values

```python
# Identifying columns with null values
columns_with_null_values = [col for col in data.columns
                            if data[col].isnull().any()]

data = data.drop(columns_with_null_values, axis=1)

data.head()
```

```python
# Correcting incorrect dates
data.at[3378, "Date"] = "02/23/1975"
data.at[7512, "Date"] = "04/28/1985"
data.at[20650, "Date"] = "03/13/2011"
```

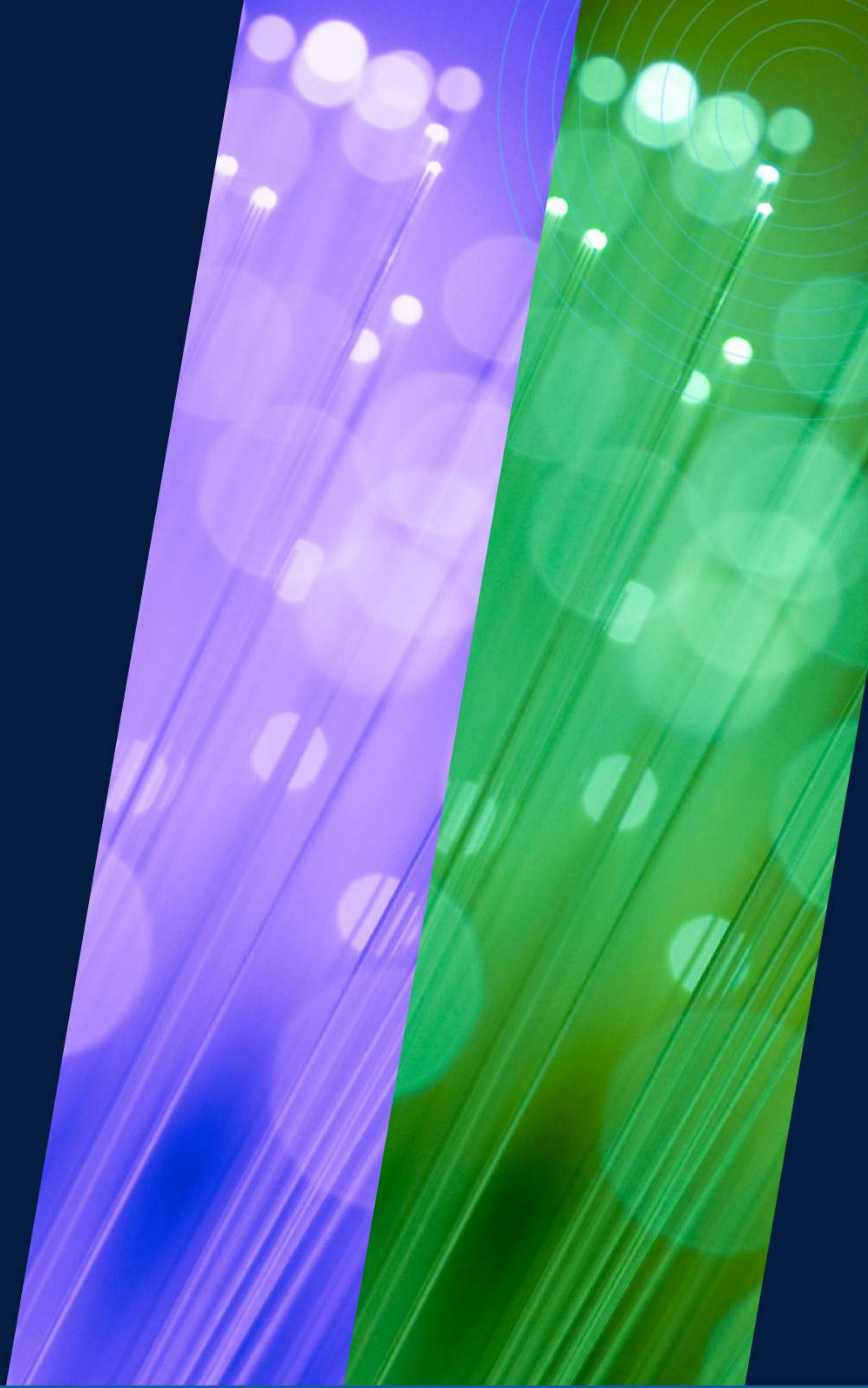
DATA PROCESSING

1) The Data type of 'Date' is converted to dtype datetime64(ns)
2) The Data type of 'Time' is converted to dtype timedelta64
3) Attribute Addition: The "Date_Time" column holds the combined parsed values of datetime.

```
# Converting the datatype of 'Time' from numpy object to timedelta64[ns]
data['Time'] = pd.to_timedelta(data['Time'])

# Merging 'Date' and 'Time' into a new 'Date_Time' column
data["Date_Time"] = data["Date"] + data["Time"]

data.info()
```



DATA PREPARATION

Updated column properties:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 23412 entries, 0 to 23411
Data columns (total 13 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Date              23412 non-null  datetime64[ns]
 1   Time              23412 non-null  timedelta64[ns]
 2   Latitude          23412 non-null  float64
 3   Longitude         23412 non-null  float64
 4   Type              23412 non-null  object
 5   Depth             23412 non-null  float64
 6   Magnitude         23412 non-null  float64
 7   ID                23412 non-null  object
 8   Source            23412 non-null  object
 9   Location Source   23412 non-null  object
 10  Magnitude Source  23412 non-null  object
 11  Status            23412 non-null  object
 12  Date_Time         23412 non-null  datetime64[ns]
dtypes: datetime64[ns](2), float64(4), object(6), timedelta64[ns](1)
memory usage: 2.3+ MB
```

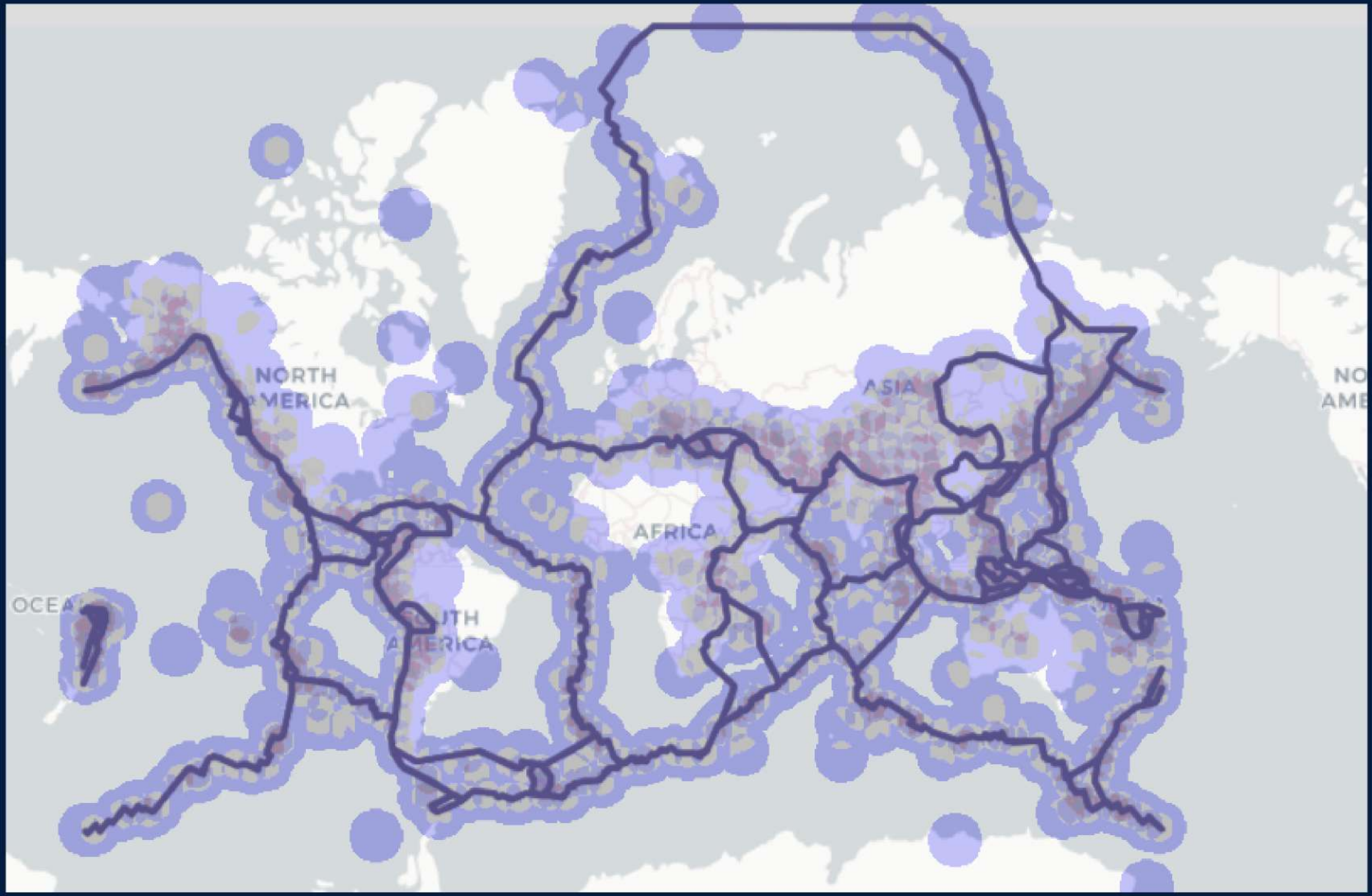
DATA PREPARATION

Addition of new Date_Time column:

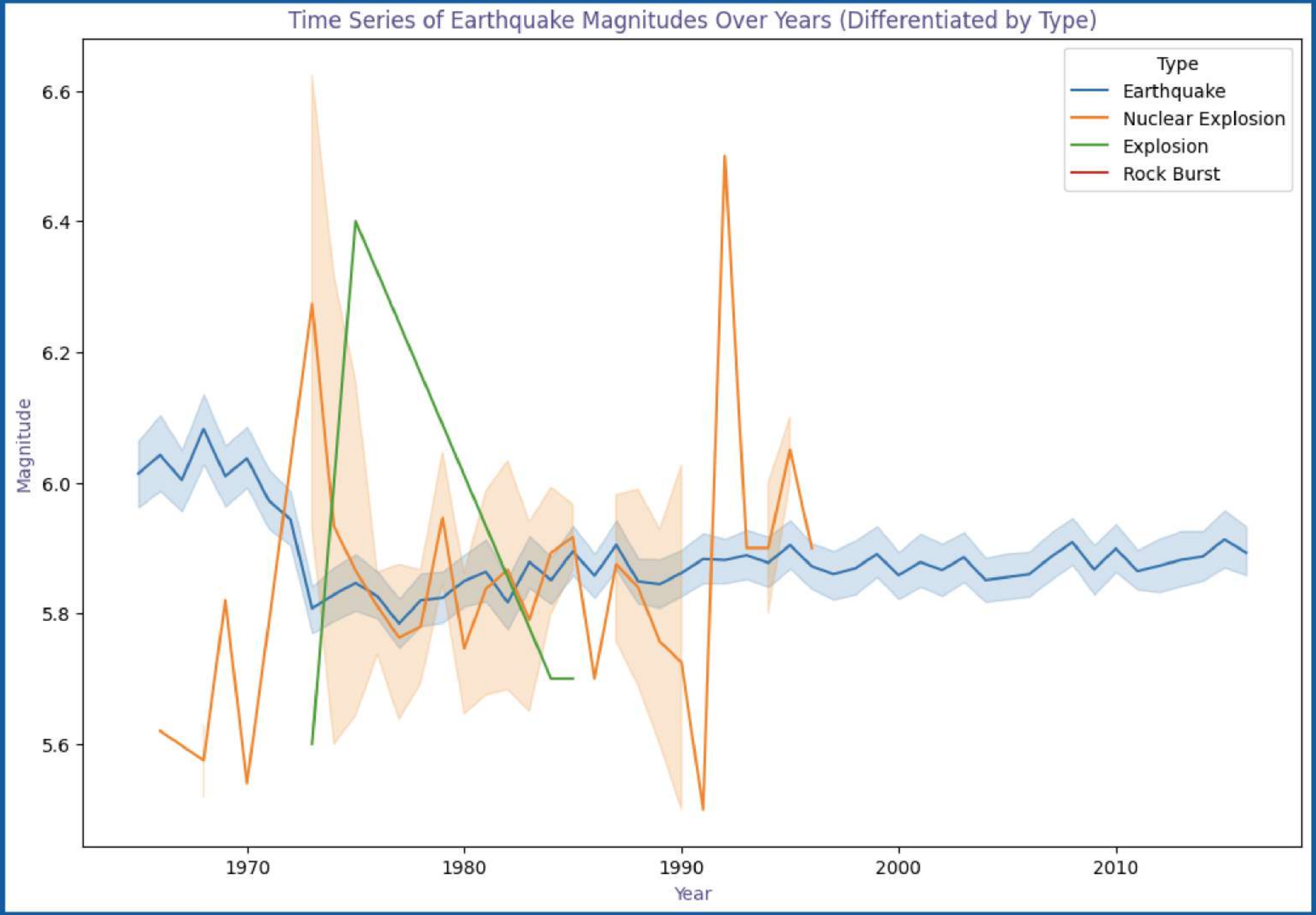
	Date	Time	Latitude	Longitude	Type	Depth	Magnitude	ID	Source	Location Source	Magnitude Source	Status	Date_Time
0	1965-01-02	0 days 13:44:18	19.246	145.616	Earthquake	131.6	6.0	ISCGEM860706	ISCGEM	ISCGEM	ISCGEM	Automatic	1965-01-02 13:44:18
1	1965-01-04	0 days 11:29:49	1.863	127.352	Earthquake	80.0	5.8	ISCGEM860737	ISCGEM	ISCGEM	ISCGEM	Automatic	1965-01-04 11:29:49
2	1965-01-05	0 days 18:05:58	-20.579	-173.972	Earthquake	20.0	6.2	ISCGEM860762	ISCGEM	ISCGEM	ISCGEM	Automatic	1965-01-05 18:05:58
3	1965-01-08	0 days 18:49:43	-59.076	-23.557	Earthquake	15.0	5.8	ISCGEM860856	ISCGEM	ISCGEM	ISCGEM	Automatic	1965-01-08 18:49:43
4	1965-01-09	0 days 13:32:50	11.938	126.427	Earthquake	15.0	5.8	ISCGEM860890	ISCGEM	ISCGEM	ISCGEM	Automatic	1965-01-09 13:32:50

Visualisation

Relation between natural earthquakes and the locations of tectonic plates



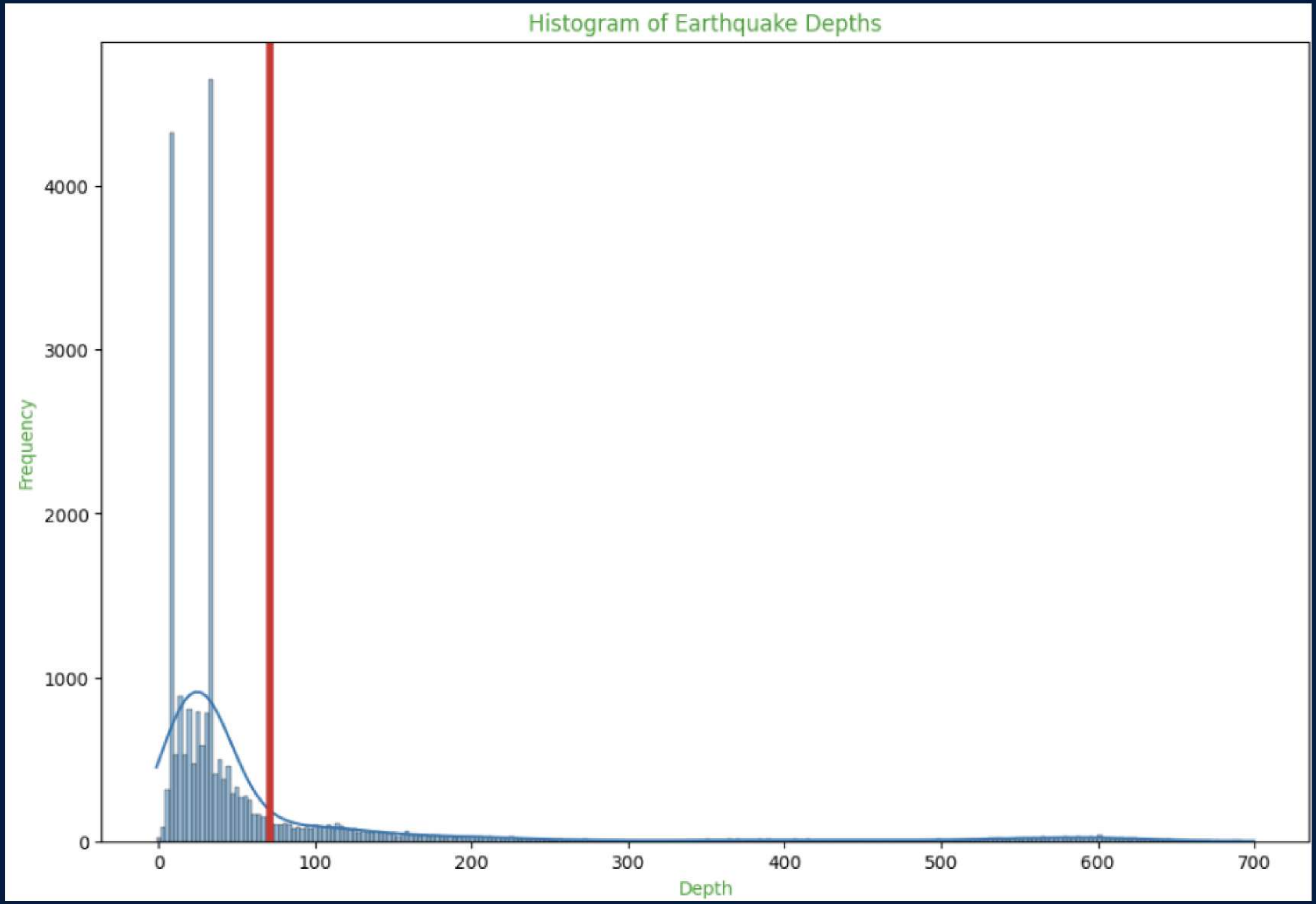
VISUALISATION



Distribution of magnitudes from 1965 to 2016 with interpretation of the cause

VISUALISATION

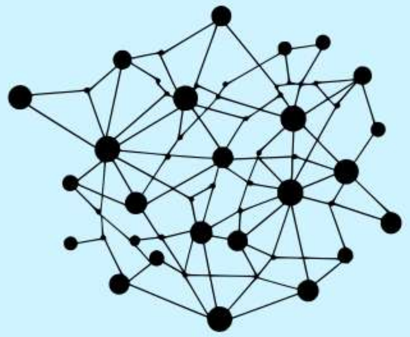
Frequency of NATURAL EARTHQUAKES with certain depths



ML MODELS

Time Series Forecasting and Regression Based Models:

1. ARIMA (AutoRegressive Integrated Moving Average)
2. Gradient Boosting Regression (GBR)
3. Support Vector Regression (SVR)
4. Random Forest Regressor (RFR)
5. Custom Neural Network (NN)



Classification Models:

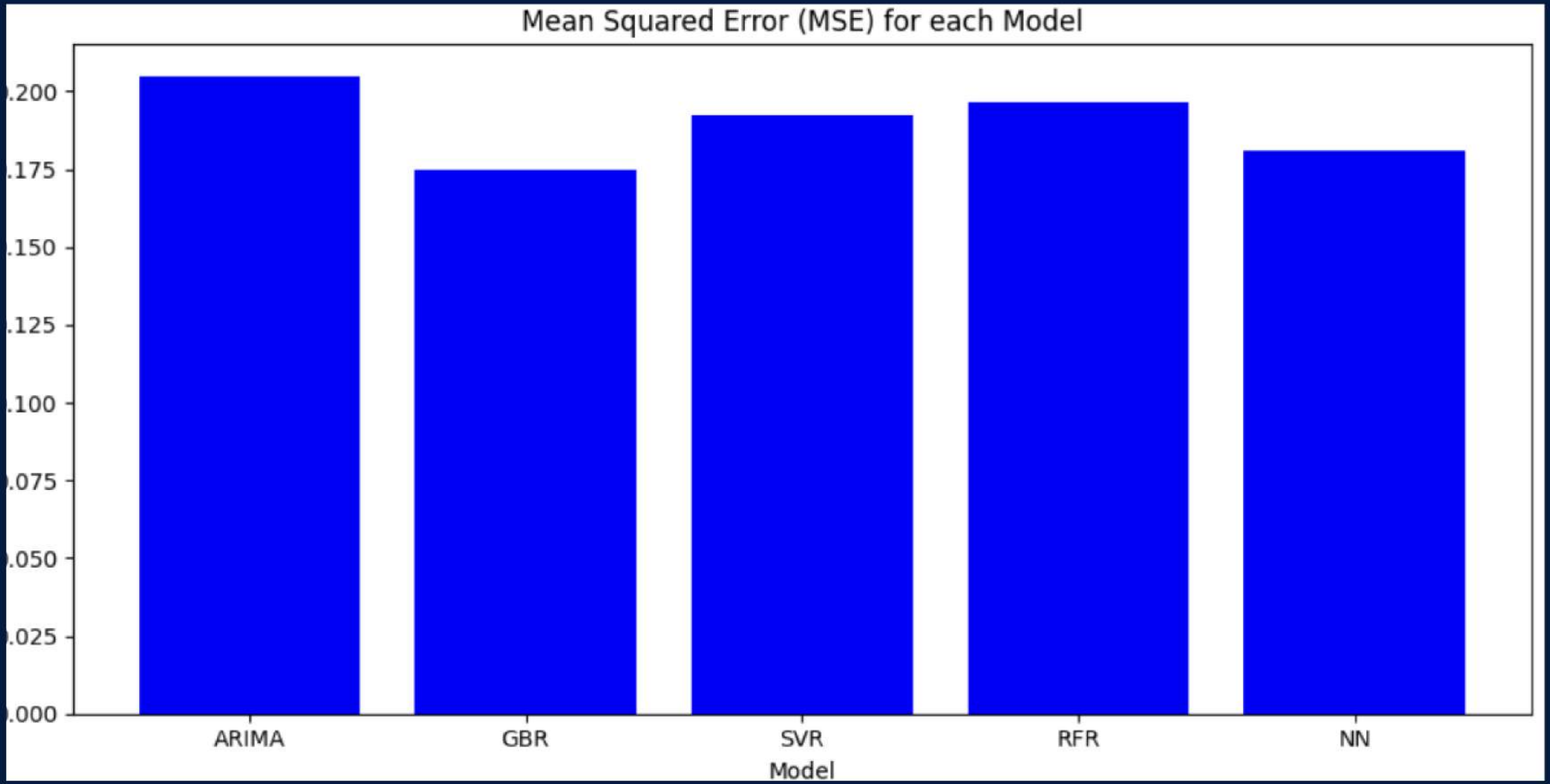
1. Custom Neural Network (NN)
2. Gradient Boosting Classifier (GBC)

RESULT ANALYSIS

Final Results for Regression Models

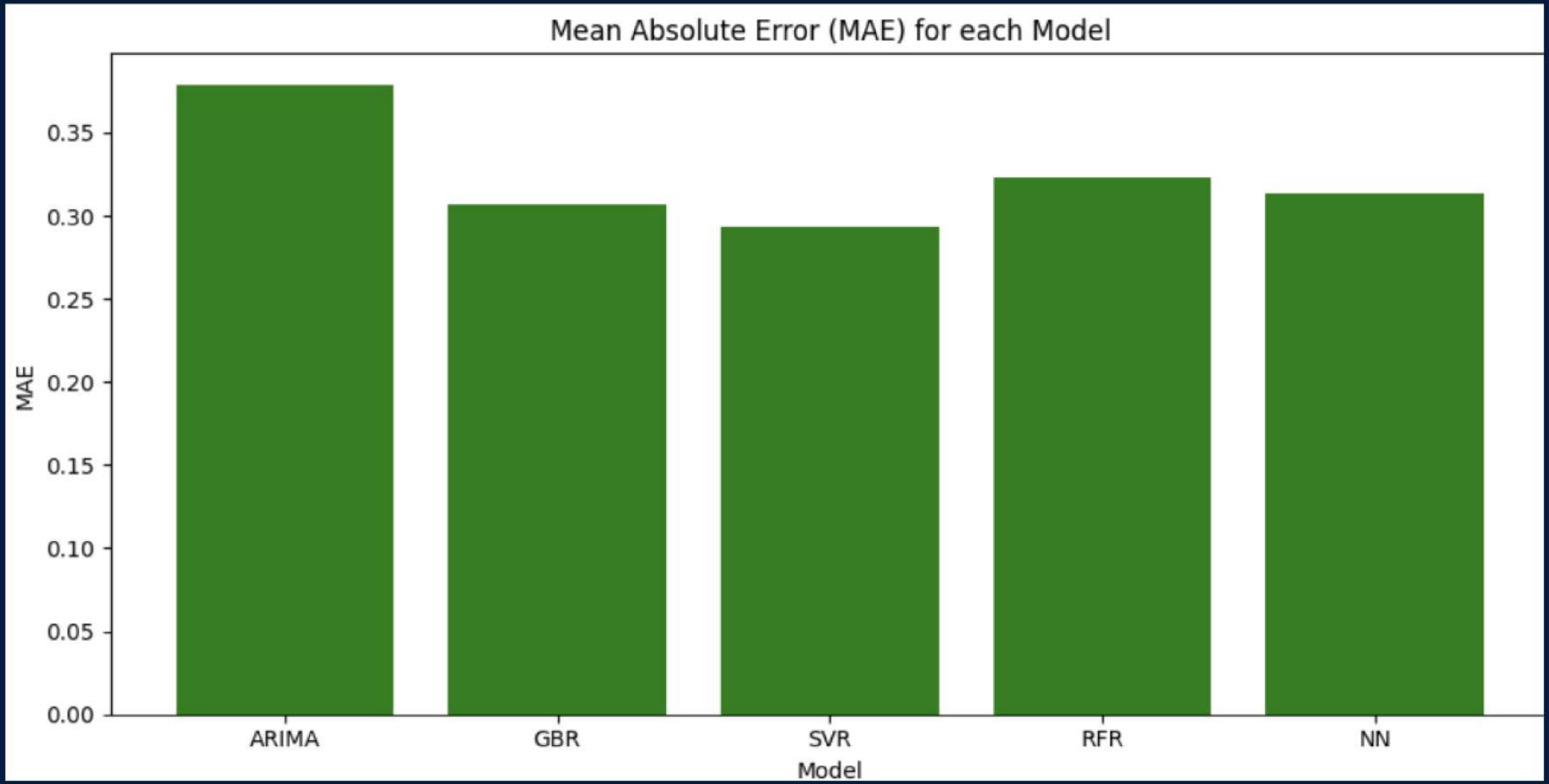
	Model	MSE	MAE	Time
0	ARIMA	0.204821	0.378641	0.331743
1	GBR	0.174783	0.307118	0.009643
2	SVR	0.192384	0.293057	4.016614
3	RFR	0.196806	0.323015	0.198519
4	NN	0.181095	0.313725	10.863598

RESULTS



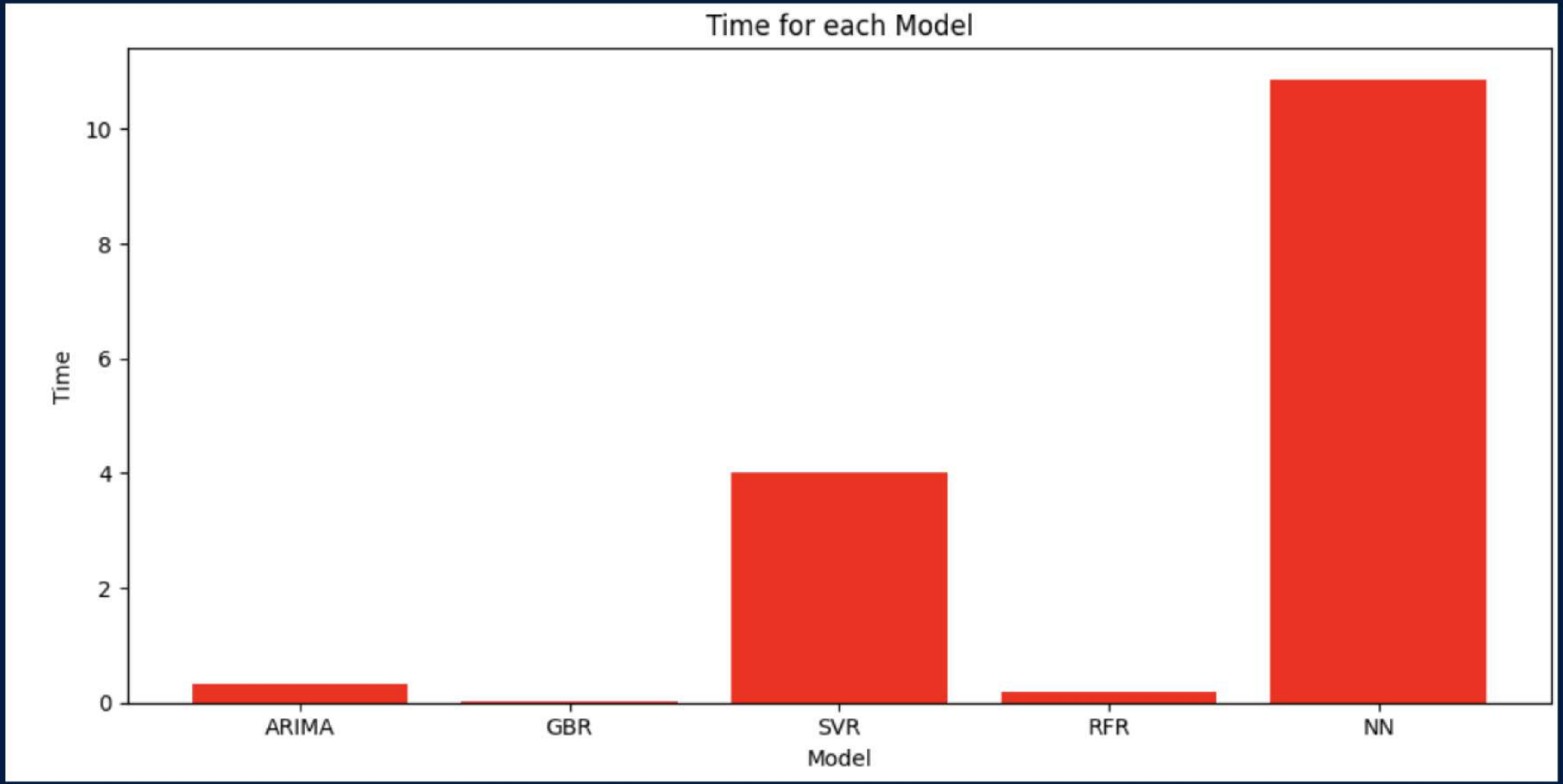
Mean Squared Error (MSE) for each Model
.200
.175
.150
.125
.100
.075
.050
.025
.000
ARIMA
GBR
SVR
RFR
NN
Model

RESULTS



Mean Absolute Error (MAE) for each Model
MAE
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0.00
ARIMA
GBR
SVR
RFR
NN
Model

RESULTS

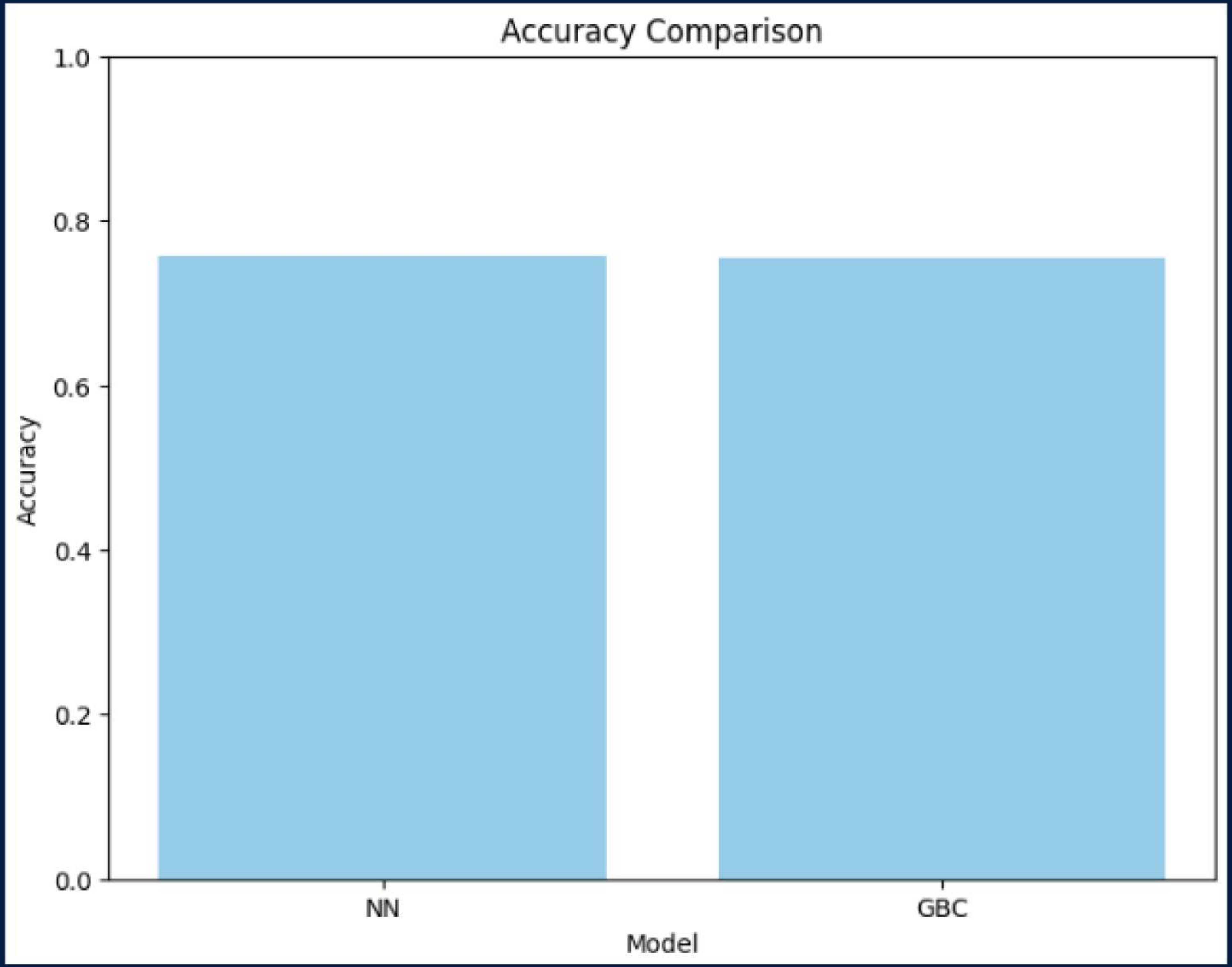


RESULT ANALYSIS

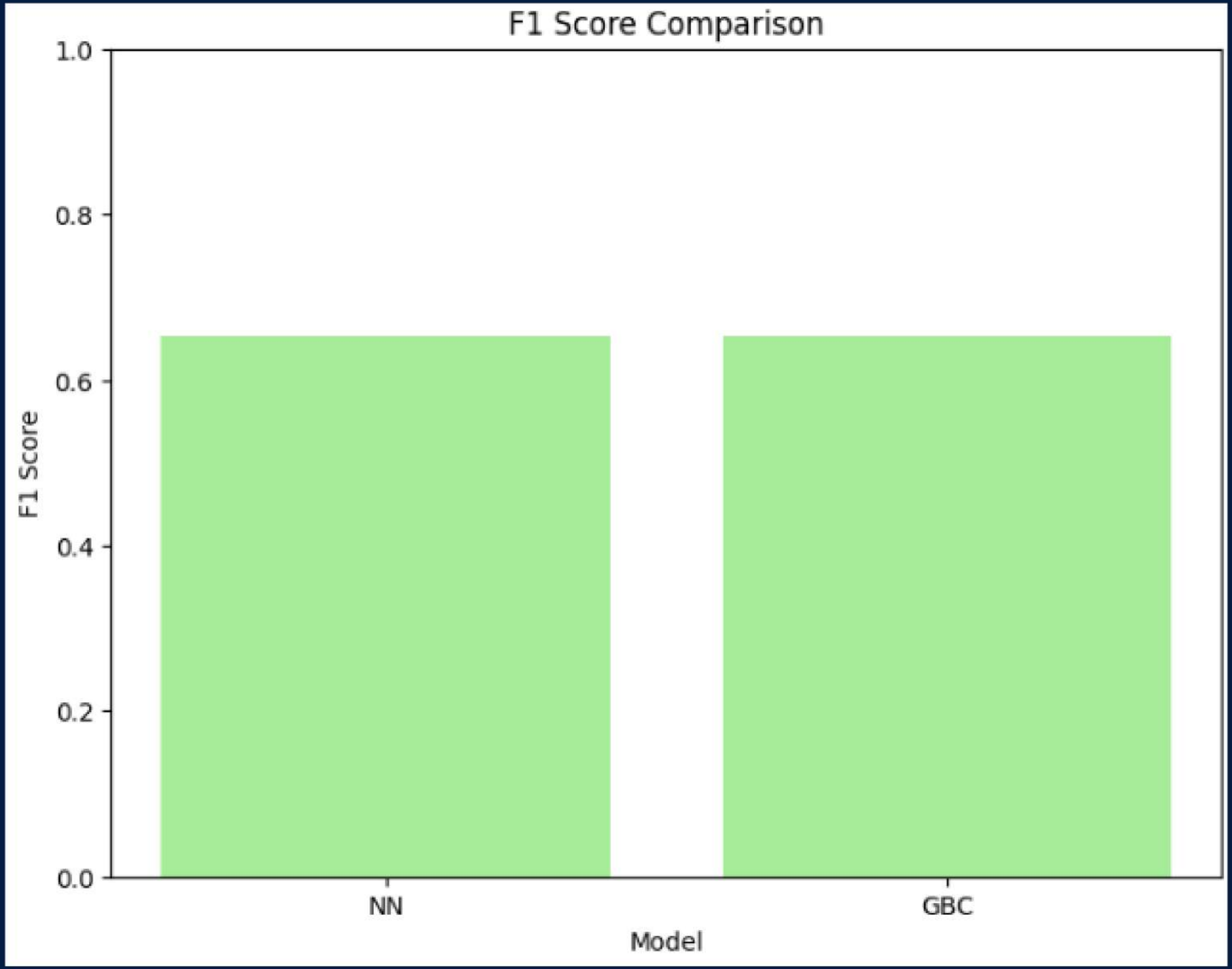
Module accuracy and results of the classification models

	Model	Accuracy	F1 Score	Time
0	NN	0.757963	0.653606	24.764408
1	GBC	0.754035	0.653732	0.037644

RESULT ANALYSIS



RESULT ANALYSIS



CONCLUSION

- If the objective is to predict earthquake magnitude directly (regression task), the Gradient Boosting Regressor (GBR) appears to be the most effective model.
- However, if the objective is to classify earthquake occurrences (classification task), the Gradient Boosting Classifier (GBC) is the preferred choice based on the provided metrics.
- Both GBR and GBC demonstrate strong performance in terms of predictive accuracy, making them suitable candidates for predicting earthquake-related phenomena based on geographical coordinates and depth.