National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Vishwakh Madhu
Student ID: x21125295

School of Computing
National College of Ireland

Supervisor: Dr.Hicham Rifai

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Vishwakh Madhu
Student ID:	x21125295
Programme:	MSc in Data Analytics
Year:	2022-2023
Module:	MSc Research Project
Supervisor:	Dr.Hicham Rifai
Submission Due Date:	01/02/2023
Project Title:	Configuration Manual
Word Count:	XXX
Page Count:	5

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	
Date:	1st February 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Vishwakh Madhu
x21125295

1 Introduction

The techniques and various software and hardware specifications utilized in the research project "Comparative study of time series forecasting methodologies in predicting Crime rate" is described in this configuration manual.

This guide's parts are as follows: Section 3 provides further information on the environment setup's characteristics. Section 4 discusses the libraries required to finish this project. The whole dataset is described in Section 5. Section 6 offers information about the code repository and the models' implementation, while Section 7 evaluates the models' performance.

2 Overview

This research compares several time series models for crime prediction in the city of Chicago.

3 System Specifications

The prerequisites are as follows: the hardware and software infrastructures needed to train a model on such a large amount of data.

3.1 Hardware Requirements

Operating system : Windows 11 Home Processor: AMD Ryzen 5 3550H with Radeon Vega Mobile Gfx 2.10 GHz Installed RAM: 8GB System Type: 64-bit operating system, x64-based processor

3.2 Software Requirements

Programming Language: Python version 3.6.9 Integrated Development Environment (IDE): Anaconda Navigator's Juypter Notebook.

4 Python Libraries Required

All of the Python libraries required to carry out this research study are presented in Figure below along with the import codes. libraries for Python are installed.

```
In [1]: import numpy as np
        import pandas as pd
        import sys
        import matplotlib as mpl
        import matplotlib.pyplot as plt
        import geopandas
        from shapely.geometry import Polygon, Point
        from sklearn.preprocessing import StandardScaler
        import datetime as dt
        from scipy.stats.mstats import winsorize
        from statsmodels.tsa.stattools import adfuller
        from statsmodels.tsa.seasonal import seasonal_decompose
        from statsmodels.tsa.arima_model import ARIMA
        from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
        from tqdm import tqdm
        from datetime import datetime, timedelta
        import matplotlib.animation
        from sklearn.metrics import mean_absolute_error
        from statsmodels.tsa.statespace.sarimax import SARIMAX
        import statsmodels.api as sm
```

Figure 1: Python Libraries

5 Data Sources

5.1 Dataset

crime dataset is collected from the government's official website.the dataset contains of 22 columns and 7690564 rows. fig 2 represents the same.

	ID	Case Number	Date	Block	IUCR	Primary Type	Description	Location Description	Arrest	Domestic	...	Ward	Community Area	FBI Code	Coordinat
0	10224738	HY411648	09/05/2015 01:30:00 PM	043XX S WOOD ST	0486	BATTERY	DOMESTIC BATTERY SIMPLE	RESIDENCE	False	True	...	12.0	61.0	08B	1165074.(
1	10224739	HY411615	09/04/2015 11:30:00 AM	008XX N CENTRAL AVE	0870	THEFT	POCKET-PICKING	CTA BUS	False	False	...	29.0	25.0	06	1138875.(
2	11646166	JC213529	09/01/2018 12:01:00 AM	082XX S INGLESIDE AVE	0810	THEFT	OVER $500	RESIDENCE	False	True	...	8.0	44.0	06	Nal
3	10224740	HY411595	09/05/2015 12:45:00 PM	035XX W BARRY AVE	2023	NARCOTICS	POSS: HEROIN(BRN/TAN)	SIDEWALK	True	False	...	35.0	21.0	18	1152037.(
4	10224741	HY411610	09/05/2015 01:00:00 PM	0000X N LARAMIE AVE	0560	ASSAULT	SIMPLE	APARTMENT	False	True	...	28.0	25.0	08A	1141706.(

5 rows × 22 columns

Figure 2: Dataset

6 Project implementation

The data set has been read into the Python environment using Pd.read fuction() on the Anaconda Navigator after being chosen.

```
: data=pd.read_csv("Crimes_-_2001_to_Present.csv")

: data.head()
```

Figure 3: Loading dataset

7 Modelling

Several models that have been proved useful for predicting and forecasting future values have been utilized in this stage.

7.1 ARIMA and SARIMA modeling

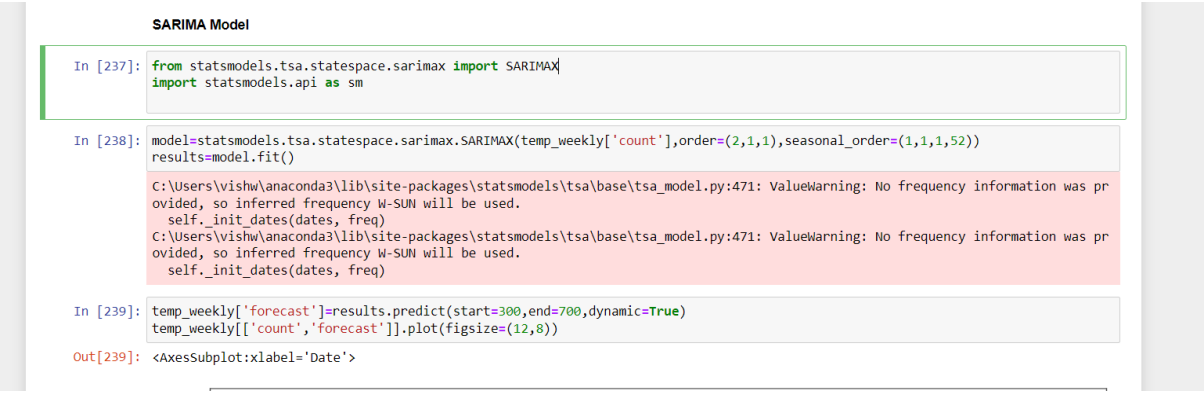

Figure 4: SARIMA Model

7.2 XG Boost

XG Boost model is implemented. shown in fig 5

7.3 Random Forest Model

Random Forest model is implemented. shown in fig 6

```
In [318]: reg = xgb.XGBRegressor(base_score=0.5, booster='gbtree',
                                 n_estimators=10000,
                                 early_stopping_rounds=50,
                                 objective='reg:linear',
                                 max_depth=3,
                                 learning_rate=0.01)
          reg.fit(X_train, y_train,
                  eval_set=[(X_train, y_train), (X_test, y_test)],
                  verbose=100)

          [03:59:21] WARNING: C:/buildkite-agent/builds/buildkite-windows-cpu-autoscaling-group-i-030221e36e1a46bfb-1/xgboost/xgboost-ci-
          windows/src/objective/regression_obj.cu:213: reg:linear is now deprecated in favor of reg:squarederror.
          [0]     validation_0-rmse:765.24505     validation_1-rmse:590.36827
          [100]   validation_0-rmse:290.04452     validation_1-rmse:206.90839
          [200]   validation_0-rmse:124.19117     validation_1-rmse:95.08138
          [300]   validation_0-rmse:74.70517      validation_1-rmse:89.40953
          [304]   validation_0-rmse:73.82987      validation_1-rmse:89.72653

Out[318]: XGBRegressor(base_score=0.5, booster='gbtree', callbacks=None,
                       colsample_bylevel=1, colsample_bynode=1, colsample_bytree=1,
                       early_stopping_rounds=50, enable_categorical=False,
                       eval_metric=None, feature_types=None, gamma=0, gpu_id=-1,
                       grow_policy='depthwise', importance_type=None,
                       interaction_constraints='', learning_rate=0.01, max_bin=256,
                       max_cat_threshold=64, max_cat_to_onehot=4, max_delta_step=0,
                       max_depth=3, max_leaves=0, min_child_weight=1, missing=nan,
                       monotone_constraints='()', n_estimators=10000, n_jobs=0,
                       num_parallel_tree=1, objective='reg:linear', predictor='auto', ...)

In [319]: fi = pd.DataFrame(data=reg.feature_importances_,
                            index=reg.feature_names_in_,
                            columns=['importance'])
          fi.sort_values('importance').plot(kind='barh', title='Feature Importance')
```

Figure 5: XG Boost Model

```
In [356]: finaldf = finaldf.reset_index(drop=True)
          test_length=162
          end_point = len(finaldf)
          x = end_point - test_length
          finaldf_train = finaldf.loc[:x - 1, :]
          finaldf_test = finaldf.loc[x:, :]
          finaldf_test_x = finaldf_test.loc[:, finaldf_test.columns != 'count']
          finaldf_test_y = finaldf_test['count']
          finaldf_train_x = finaldf_train.loc[:, finaldf_train.columns != 'count']
          finaldf_train_y = finaldf_train['count']
          rfe = RFE(RandomForestRegressor(n_estimators=100, random_state=1))
          fit = rfe.fit(finaldf_train_x, finaldf_train_y)
          y_pred = fit.predict(finaldf_test_x)

In [357]: finaldf_test_y

Out[357]: 515    4662
          516    4292
          517    4474
          518    4369
          519    3313
                 ...
          672    3302
          673    3361
          674    3380
          675    3227
          676    1767
          Name: count, Length: 162, dtype: int64

In [358]: y_true = np.array(finaldf_test_y)
```

Figure 6: Random Forest Model

7.4 Prophet Model

Prophet model is fitted. shown in fig 7

In [371]:
```
validation_data = temp_weekly.drop(train_data.index)

print(f'training data size : {train_data.shape}')
print(f'validation data size : {validation_data.shape}')

train_data = train_data.reset_index()
validation_data = validation_data.reset_index()
```

```
training data size : (585, 2)
validation data size : (146, 2)
```

In [372]:
```
model = Prophet()
model.fit(train_data)
```

```
INFO:prophet:Disabling weekly seasonality. Run prophet with weekly_seasonality=True to override this.
INFO:prophet:Disabling daily seasonality. Run prophet with daily_seasonality=True to override this.
C:\Users\vishw\anaconda3\lib\site-packages\prophet\forecaster.py:896: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
```

Out[372]:
```
<prophet.forecaster.Prophet at 0x18bc03e5df0>
```

In [373]:
```
prediction = model.predict(pd.DataFrame({'ds':validation_data['ds']}))
y_actual = validation_data['y']
y_predicted = prediction['yhat']
y_predicted = y_predicted.astype(int)
mean_absolute_error(y_actual, y_predicted)
```

```
C:\Users\vishw\anaconda3\lib\site-packages\prophet\forecaster.py:896: FutureWarning:

The frame.append method is deprecated and will be removed from pandas in a future version. Use pandas.concat instead.
```

Figure 7: Prophet

References