National College of Ireland

Configuration Manual

MSc Research Project
Data Analytics

Likhita Sargur Yathishkumar
Student ID: x23203951

School of Computing
National College of Ireland

Supervisor: Jaswinder Singh

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Likhita Sargur Yathishkumar
Student ID:	X23203951
Programme:	MSc Data Analytics **Year:** 2025-26
Module:	MSc Research Practicum
Lecturer:	Jaswinder Singh
Submission Due Date:	27/01/2026
Project Title:	Predicting Formula 1 Podium Finish Probabilities Using Machine Learning: A Comparative Study of Pre-race and Within-race Feature Scenarios
Word Count:	1322 **Page Count:** 7

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: Likhita Sargur Yathishkumar

Date: 11/12/2025

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

Likhita Sargur Yathishkumar
Student ID: x23203951

1 Introduction

This manual is provided as a step by step guide of the experiment setup and the required information showing packages imported or used for the machine learning project involving 'Predicting Formula 1 Podium Finish Probabilities Using Machine Learning: A Comparative Study of Pre-race and Within-race Feature Scenarios'. This research work access the datasets and machine learning algorithms for the prediction. The research evaluates two machine learning algorithms: Random Forest Classifier and Light GBM Classifier, followed by class weighted sampling and a better performing model that was chosen amongst the two was then selected for SHAP and further the Probability Calibration was conducted.
This manual has all the technical details including the Software used and the Hardware requirements that are needed. This guide provides a step by step approach for the users to configure that it gives similar experimental environment and having similar results.

2 Development Environment

For this research study, the environment used is Mac OS. Specifications of both hardware and software are mentioned below in detail. And the datasets used for the project is also mentioned with the links. second section.

2.1 Hardware Requirements

- Machine: Any modern x86_64/ARM64 (Apple M-series or Intel/AMD).
- GPU is not required.
  • Example used in this project: macOS on Apple M3 for local
  • RAM: 8 GB minimum (16 GB recommended)
- Operating System: Mac OS
- Chip: Apple M3

The environment was created using the local hardware system specifications that is mentioned above and not any explicit requirements are required for this other than these.

2.2 Software Requirements

- Operating System: Mac OS or any other operating system can be used.
  Example: windows 10 and above or Ubuntu 20.04
- Programming Language: Python version 3.12.7

```
[1]: !python --version

Python 3.12.7
```

Figure 1: Python Version check

- IDE (Integrated Development Environment): Jupyter Notebook 6.5.4 and above.
- Conda: Anaconda

2.3 Libraries Imported

There are several python libraries that are required to be installed for the execution of the code, and these libraries can be installed using the pip command in either Jupyter Notebook or also directly from the terminal.

- Numpy
- Pandas
- Matplotlib
- Seaborn
- Shap
- Random
- Scikit-learn

```
: ## To Imports all the necessary libraries and global display
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from pathlib import Path

from sklearn.compose import ColumnTransformer
from sklearn.preprocessing import OneHotEncoder
from sklearn.impute import SimpleImputer
from sklearn.pipeline import Pipeline
from sklearn.metrics import classification_report, ConfusionMatrixDisplay, confusion_matrix, roc_curve, auc
from sklearn.utils.class_weight import compute_class_weight
from sklearn.ensemble import RandomForestClassifier
from lightgbm import LGBMClassifier
from sklearn.metrics import f1_score, log_loss, roc_auc_score
from sklearn.preprocessing import label_binarize
from sklearn.calibration import CalibratedClassifierCV
from sklearn.metrics import log_loss

import shap

import random
np.random.seed(42)
random.seed(42)

pd.set_option("display.max_columns", 120)
plt.rcParams["figure.figsize"] = (9, 5)
```

Figure 2: Image of Notebook file showing all the libraries used

3 Dataset

For this project the first dataset is having multiple CSV files which was downloaded directly from the Kaggle website. In this file the dataset contains information from the year 1950-2024 and it is mentioned in the title when the linked is accessed as well, as shown in the figure 2. But the URL link still shows the year as 1950-2020 where it was not updated or lags.

The second dataset that was used is the weather dataset which is also publicly available from the Kaggle website. This contains the data from the year 1950 to the year 2024.

- Multiple data set for Formula-1 Championship:

  Rao, R. (2024) *Formula 1 World Championship (1950–2020)* [Dataset]. Kaggle. Available at: https://www.kaggle.com/datasets/rohanrao/formula-1-world-championship-1950-2020/data **(**Accessed: 9 December 2025**).** [Title indicates 1950–2024; legacy URL slug reads 1950–2020.]

- Weather Dataset information from Kaggle:

  Kostyrya, M. (2024) *Formula 1 Weather Info (1950–2024)* [Dataset]. Kaggle. Available at: https://www.kaggle.com/datasets/mariyakostyrya/formula-1-weather-info-1950-2024

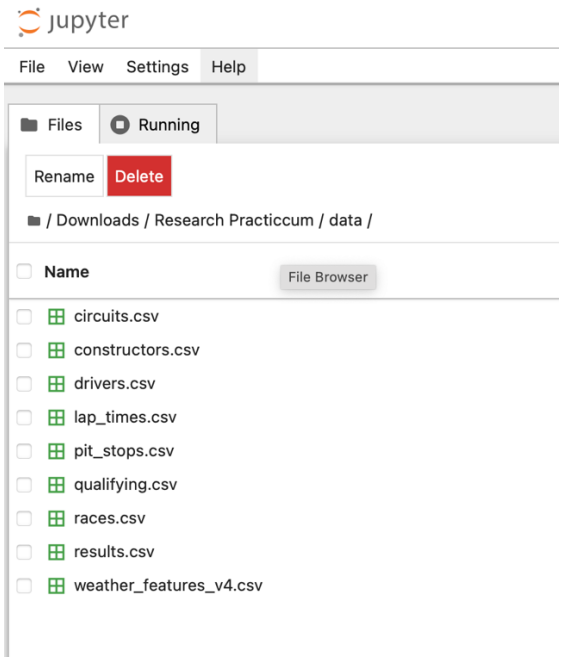

Figure 3: All the dataset file in the local

4 Project Code File

For this project only one single Jupyter Notebook file was used. From importing the libraries, loading the dataset, EDA, Model training and Results are all conducted in this single file.

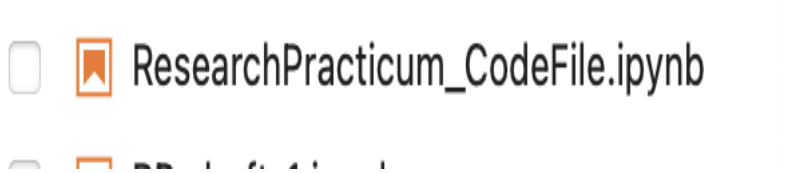

Figure 4: Python file used

Run the notebook file from starting till end, It will:

- Load and merge F1 Tables with weather.
- Check for the path where the dataset and CSV files are stored. That is important to map the path to where the files are present in the local.
- To Build the Pre-race and Within-race feature sets. The within-race is a retrospective one.
- Train the two modles - Random Forest and Light GBM.
- Choose the winner by validation Macro F1 as the primary and Log loss as the secondary terms of valuation.
- Evaluate on test.
- The graphs and Confusion matrix, ROC curve and top 10 drivers probability images will be seen.
- Run SHAP for the selected model and scenario using the validation derived in the previous steps.
- Save or take screenshots of the tables and images if required for further.

5 Data Preparation

In this study, from extracting the data that is loading all the datasets from CSV file. And Data Pre-processing like handling missing values, Exploratory data analysis to understand the distribution of the variables and the target variable, and Feature engineering everything was done.

```
## To Load all the datasets including weather
DATA_DIR = Path("/Users/likhita/Downloads/Research Practiccum/data")

races        = pd.read_csv(DATA_DIR / "races.csv", parse_dates=["date"], dayfirst=False, infer_datetime_format=True)
results      = pd.read_csv(DATA_DIR / "results.csv")
drivers      = pd.read_csv(DATA_DIR / "drivers.csv", parse_dates=["dob"], dayfirst=False, infer_datetime_format=True)
constructors = pd.read_csv(DATA_DIR / "constructors.csv")
qualifying   = pd.read_csv(DATA_DIR / "qualifying.csv", low_memory=False)
circuits     = pd.read_csv(DATA_DIR / "circuits.csv")
pit_stops    = pd.read_csv(DATA_DIR / "pit_stops.csv")
lap_times    = pd.read_csv(DATA_DIR / "lap_times.csv")
weather      = pd.read_csv(DATA_DIR / "weather_features_v4.csv")  # new: load now
```

Figure 5: Image showing all the Datasets that are imported from local path including weather file

After all the datasets are loaded from the local path and they are done successfully the below message will be displayed.

```
Loaded: races=(1125, 18), results=(26759, 18), drivers=(861, 9), constructors=(212, 5), qualifying=(10494, 9), circuits=(77, 9), pit_stops=(1
1371, 7), lap_times=(589081, 6), weather=(1125, 6)
```

Figure 6: Image showing all the files after loading.

Later followed by further steps as mentioned before then followed by EDA and Feature engineering steps will be done.

```
         std_lap_ms  num_pitstops  total_pit_ms  avg_pit_ms  first_pit_lap  \
0      20990.356845           NaN           NaN         NaN            NaN
1      20212.298423           NaN           NaN         NaN            NaN
2      18957.199698           NaN           NaN         NaN            NaN
3      19640.823974           NaN           NaN         NaN            NaN
4      21826.401776           NaN           NaN         NaN            NaN
...             ...           ...           ...         ...            ...
26754   7251.481525           4.0       92946.0     23236.5           12.0
26755   7168.780388           3.0       78297.0     26099.0           23.0
26756  12734.899632           1.0       41144.0     41144.0            6.0
26757  10855.300667           1.0       24941.0     24941.0            3.0
26758           NaN           NaN           NaN         NaN            NaN

       last_pit_lap  temp_c  precip_mm  wind_speed
0               NaN    37.1        0.0        17.9
1               NaN    37.1        0.0        17.9
2               NaN    37.1        0.0        17.9
3               NaN    37.1        0.0        17.9
4               NaN    37.1        0.0        17.9
...             ...     ...        ...         ...
26754          55.0    26.3        0.0        16.3
26755          29.0    26.3        0.0        16.3
26756           6.0    26.3        0.0        16.3
26757           3.0    26.3        0.0        16.3
26758           NaN    26.3        0.0        16.3

[26759 rows x 69 columns]
```

Figure 7: After merge the final dataset

6 Model Building

This section covers the list of models that were used in the study, training the models and evaluation of the models using performance metrics.

```
[24]: ## Chronological split
train_df = df[df["year"] <= 2019].copy()
val_df   = df[(df["year"] >= 2020) & (df["year"] <= 2022)].copy()
test_df  = df[df["year"] >= 2023].copy()

print(f"Split sizes | train={train_df.shape}, val={val_df.shape}, test={test_df.shape}")
```

```
Split sizes | train=(24620, 68), val=(1220, 68), test=(919, 68)
```

Figure 8: Chronological Split

Models use the fixed random seeds inside the notebook to get the identical results across the runs.

Models used:

- Random Forest Classifier

- Light GBM Classifier

7 Evaluation

Below are the mentioned performance metrics that were used to evaluate the performance of the models. The Metrics used for evaluation are:

- Macro F1
- Accuracy
- Log Loss
- Brier
- Macro-AUC OVR

```
=== RF (pre_race) – Test ===
              precision    recall  f1-score   support

           0      0.936     0.885     0.910       781
           1      0.103     0.152     0.123        46
           2      0.156     0.109     0.128        46
           3      0.432     0.761     0.551        46

    accuracy                          0.803       919
   macro avg      0.407     0.477     0.428       919
weighted avg      0.830     0.803     0.813       919
```

Figure 9: Performance Metrics for RF (pre-race) for Test

```
=== LGBM (pre_race) – Test ===
              precision    recall  f1-score   support

           0      0.947     0.877     0.911       781
           1      0.165     0.348     0.224        46
           2      0.111     0.087     0.098        46
           3      0.429     0.587     0.495        46

    accuracy                          0.797       919
   macro avg      0.413     0.475     0.432       919
weighted avg      0.840     0.797     0.815       919
```

Figure 10: Performance Metrics for LightGBM (pre-race) for Test

```
=== LGBM (within_race) — Test ===
              precision    recall  f1-score   support

           0      0.941     0.927     0.934       781
           1      0.158     0.196     0.175        46
           2      0.154     0.130     0.141        46
           3      0.463     0.543     0.500        46

    accuracy                         0.831       919
   macro avg      0.429     0.449     0.438       919
weighted avg      0.839     0.831     0.835       919
```

Figure 11: Performance Metrics for RF (Within-race) for Test

```
=== RF (within_race) — Test ===
              precision    recall  f1-score   support

           0      0.924     0.939     0.931       781
           1      0.184     0.152     0.167        46
           2      0.182     0.087     0.118        46
           3      0.545     0.783     0.643        46

    accuracy                         0.849       919
   macro avg      0.459     0.490     0.465       919
weighted avg      0.831     0.849     0.838       919
```

Figure 12: Performance Metrics for LightGBM (Within-race) for Test

```
{'logloss_raw': 0.5360630527007408, 'logloss_cal': 0.5104815070547608, 'brier_raw': 0.06747816670347229, 'brier_cal': 0.050742041991224736}
```

Figure 13: Calibrated Search Metrics

8 Results and Visualizations

The notebook shows the figures which can be saved or taken screenshot of and store where feasible. Example one can store like this - to outputs/figures/ and tables to outputs/tables/ to identify easily. Key images that are included in the Report include:

- Confusion matrices for both Pre-Race and Within-Race scenario.
- ROC curves per scenario.
- Top-10 podium probability graph for each scenario.
- SHAP summary for the selected class
- Reliability graph for Calibrated Curve

For all the graphs and tables of the evaluation and results, refer the Report which has all the required images and explanation.