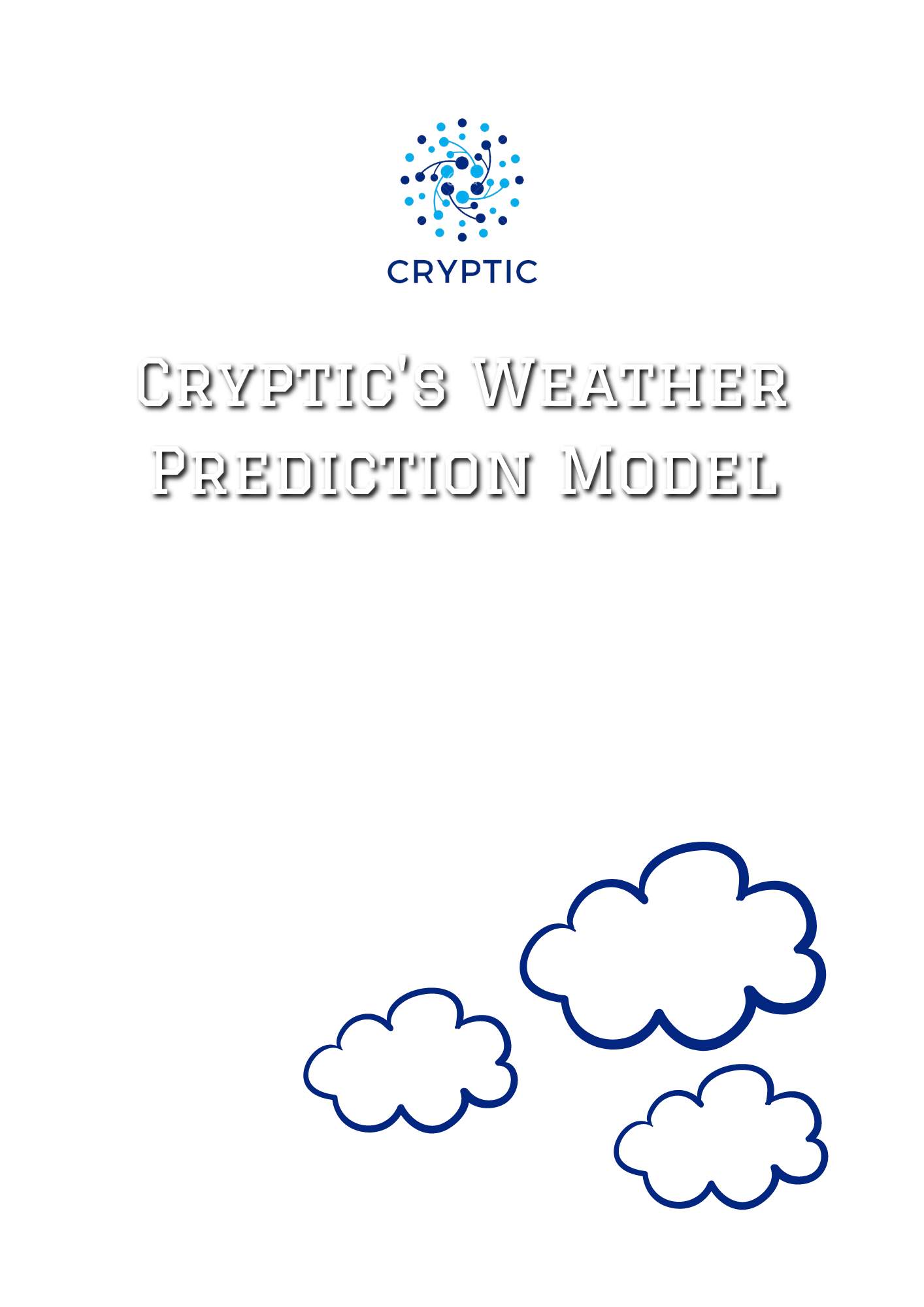
****

**Team Cryptic’s Members**

* Okampka Tobechukwu - Data Scientist (Team Lead)
* Okechukwu Joshua - ML Engineer
* Mbachu Divine - Python Developer
* Anazodo Chukwuemeka - Video Editor
* Sakshi Kodre - Python Developer

**Team Cryptic’s Mentor(s)**

* Mr. Azib - Lead Mentor

**Challenge:**

Challenge your skills on data processing and analysis by creating an accurate weather prediction model for the F1 2021 video game.

**Problem Statement:**

Points - sometimes races – in F1 are won and lost based on interpreting what the weather is going to do and acting accordingly. Race engineers don’t supply their drivers with nebulous possibilities, instead they deal in specifics, relaying precise information: how many minutes until rain starts falling; which corner it hits first; how intense; for what duration[[1]](#footnote-0). Although the final decision is down to the driver, accurate weather predictions and analysis is required to aid the driver in making the right decision and possibly winning the race.

Keeping in mind the points mentioned above, we have also designed our own weather prediction model, based on the data retrieved from the F1 2021 game, and we will discuss it in a moment.

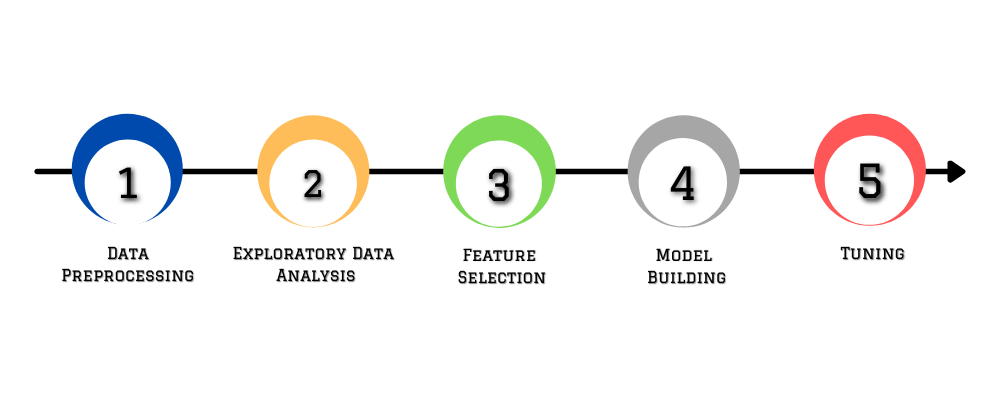
**Overview of the dataset**

The dataset consists of 3,572,328 rows(entries) and 59 columns. These data points were extracted from the Formula 1 video game developed by Code Masters.

The target feature (Weather) and some other features had been encoded. There are a total of 7794197 null values in the dataset and 3 unique data types namely: datetime, int and float.

**Approach:**

Our Approach towards this problem followed the sequence in the flowchart below:



* **EXPLORATORY DATA ANALYSIS**

We made use of exploratory data analysis to analyze and investigate the data set and summarize its main characteristics also employing data visualization methods. We made use of both univariate and bi variate visualizations. The visualization tools employed for our EDA are: Heat-maps, Box plot and histograms which are discussed below.

* **ROW AND FEATURE SELECTION**

As mentioned earlier the original data set was made up of 3,572,328 rows and 59 columns. At the end, we ended up with 99960 rows and 18 features. The question that comes to mind is: How was the data set reduced to this? What are the reasons for the row and column reduction? This questions are answered in the next few lines

Row selection

1. Features with less than 3 unique entries were dropped using a custom function.
2. All rows that had missing values for target variables were dropped from the dataset AI model.
3. Since our model was to predict 5 time steps into the future Rows that had “NUM\_WEATHER\_FORECAST” less than 5 were dropped.Rows that had more than 5 were trimmed to the Time frame window of 5

The dataset contained 1159663 duplicated entries. Features such as SESSION\_UID, PLAYER\_CAR\_INDEX and TIME\_STAMP also had duplicates in the dataset. How we handle this duplicated entries is key in determining the accuracy of our model.Duplicated and redundant entries were dropped from the dataset using a custom function at the same time preserving the class ratio of the target features

FEATURE SELECTION

Selecting the best features to use in training a model is a very tricky one and needs to be approached meticulously as it improves the machine learning process and increases the predictive power of machine learning algorithms.

We employed the use of linear correlation wrapped in heat-maps to get insight into how different features in the dataset correlates with the target features.

We made use of linear correlation to select the best features.

From the heat-maps, it was seen that the same features that correlated best with the first target output (Weather) also correlated well with the second target variable(Rain percentage). These features were retained and the rest were dropped. A threshold of 0.08 was used, any feature that had its absolute correlation value with any of the target variable less than this threshold was dropped.

* **MODEL BUILDING AND TUNING**

MODEL BULDING

Before building the model we scaled the data using minmax scaler and then split the dataset into 3: train data(xx timestamps), validation data(2000 timestamps) and test data(1 timestamps).

We employed a Feed Forward Dense neural network functional API on keras using tensor flow as the backend. We had 1 input layer and two output layers (one for classification i.e weather and the other, regression i.e rain probability)

Our network graph contained 3 hidden dense layers with 64,32 and 16 neurons respectively. We employed equal loss weights to balance the contribution of each of the losses from the two layers because they were within same range of values I.e 0-1. In this regard we gave them equal importance.

Early stop Callbacks was used to prevent the model from overfitting the data.

Tuning

bayesian optimization from python bayes opt library was used to search for best performing hyper parameters.

We were able to achieve the following on testing our model with a validation data

Weather Prediction Accuracy=91.25%

Rain Percentage Mean Absolute Error = 0.02

1. An Extract from the Official F1 website (link: <https://www.formula1.com/en/latest/features/2016/9/how-weather-forecasting-works-in-f1.html>) [↑](#footnote-ref-0)