Predicting and Analysing Damage from Hurricanes and Typhoons

F21DL – Data Mining and Machine Learning
Group UG- 3

Heriot-Watt University Dubai
Submission Date: 22/11/2024
Lecturer: Dr Neamat El Gayar, Dr Radu-Casian Mihăilescu

Authored By:		**GitHub Usernames:**
Lubna Gulnaar	H00419072	lubzgulz
Thomson Thomas	H00375097	ThomsonAT
Joseph William Abdo	H00389925	DALEKCHANNEL
Joepaul Ettumanookaran	H00421027	unicornpjs
Irin Varughese	H00419396	irinmv

Github Repository: ***https://github.com/unicornpjs/Dubai_UG_3.git***

Table Of Contents

1. Introduction

Hurricanes and typhoons are one of the most common but extremely destructive natural disasters. They can often cause widespread damage to infrastructure, landscapes, and living beings. Therefore, predicting their formation and understanding the destruction they can lead to, would help for disaster preparedness and resource allocation. For our Machine Learning Portfolio, the topic of application "Hurricane Prediction and Damage Analysis," was picked wherein our two main objectives is to use data mining and machine learning techniques like clustering, decision trees, linear and logistic regression as well as convolutional neural networks (CNNs) for:

1. Prediction: Analysing numerical data to predict the formation and trajectory of hurricanes and typhoons, focusing on wind speeds, directions, and pressure.

2. Damage Analysis: Using satellite images to classify areas as "damage" or "no damage" to assess the extent of destruction to buildings, infrastructure, and landscapes.

2.Data Sets

2.1 Description

In this project two datasets were used. Key details are provided here, and more details can be found in Appendix A.

1. HURDAT Numerical Data [1]: The HURDAT Numerical Dataset, from the National Hurricane Centre, contains detailed data on Atlantic and Pacific hurricanes and tropical storms. It also provides information on various storm attributes recorded during their lifecycle.

2. Satellite Images of Hurricane Harvey [2]: The Satellite Images of Hurricane Damage dataset contains images from areas affected by Hurricane Harvey. This dataset is used to train a Convolutional Neural Network (CNN) to classify images into ‘damage’ and ‘no damage’ categories for damage detection.

2.2 Exploratory Data Analysis (EDA)

The numerical dataset was loaded as a .CSV file and analysed using methods like .info(), .describe(), and .shape() to understand its structure, data types, and statistics, including value ranges and missing data. Rows with missing values were removed or replaced with data to ensure data completeness.

A scatter plot was created for “Maximum Wind vs. Minimum Pressure” to understand the relationship between wind speed and central pressure (Fig B1 (Appendix B)). The cluster of events at the top of the graph represent high pressure events, which could mean less intense weather phenomena, as opposed to the cluster on the bottom of the graph, which could represent high intensity weather phenomena. Similarly histograms were also plotted using the values of the Atlantic and Pacific datasets. The focus of these histograms is around mainly the maximum wind and minimum pressure (see Fig B2(Appendix B)).

A correlation matrix (see Fig B3 (Appendix B)) was generated using numeric columns, followed by a heatmap visualization (see Fig B4 (Appendix B)). The correlation matrix shows the relationship between different features in a dataset, by using values ranging from **-1 to 1**. Maximum Wind vs. Minimum Pressure show a negative correlation or a low value, interpreted as Minimum Pressure decreases, Maximum Wind speed increases. Whereas, for other features comparison the values show a comparatively stronger positive correlation that suggests that wind direction and longitude and latitude data may not provide as clear an indicator of storm for prediction compared to the Maximum Wind-Minimum Pressure relationship. Similar trend is visualized in the heatmaps. Thus, this demonstrates that predicting using Maximum Wind and Minimum Pressure is statistically optimal to use for predictive models.

When considering the image dataset, preprocessing is aimed at normalizing the images to be used in machine learning applications. This included ensuring image sizes stayed regular with all data used, and with images reshaped. Random images were selected from both damage and no damage sets of images and plotted to verify the processed image data. The use of colormaps was tested to help increase the accuracy of the values in the image but was not utilized due to the lack of key features noticeable by applying the colormaps.

3. Experimental Setup

3.1 Clustering for Hurricane Characterization and Damage Recognition

The K-means clustering algorithm was implemented to group hurricane data based on various features such as wind speed and pressure, to identify patterns in the data that could help us in categorizing different occurrences of hurricanes such as tropical storms, no hurricane etc. The *Atlantic* and *Pacific* hurricane datasets were imported and concatenated to form a combined dataset. Non-numeric columns were excluded from the dataset to retain features suitable for clustering. Additionally, rows with missing values were removed. Features were standardized using the "`StandardScaler`" to ensure equal contribution to the clustering algorithm. The K-means algorithm was applied with three clusters (k = 3). The K-means algorithm was also utilized in the images dataset with two clusters (k=2), with clustering aiming to group regions of the images in a process called segmentation, which would further allow the program to read the image.

3.2 Baseline Machine Learning Models for Hurricane Prediction

3.2.1 Decision Trees

A Decision Tree is one of the few algorithms used to create models from datasets. It achieves this by splitting the data into subsets and forming a tree based on previous questions in the branches[4]. In the primary prediction testing code two methods were implemented: the first method predicted the storm region with it being either Atlantic or Pacific. The second method determined what type of storm it was with 11 options. Both methods used the maximum wind and minimum pressure as key parameters to determine this in the model.

The first Decision Tree was made using the DecisionTreeClassifier import, which allowed for the direct creation of the Decision Tree after the data was split 70-30. After training, the model was first tested by visualising the data with a tree-like structure with a depth of 2(see Fig B5 (Appendix B)). Then, it was tested using the predict and probability prediction method, demonstrating that it could reliably predict the storm region with little issue. Additionally, (as shown in Fig B6 a (Appendix B)), two decision boundary visualisations were created to compare the Atlantic and Pacific data and classify it. The Atlantic data was shown to have a more extensive range in the minimum pressure section of the data, and it was classified with a depth of 0, while the Pacific data had a narrower region of the minimum pressure; however, it had a bit of a more extended range in the maximum wind section.

While both have a specific range, they overlap much in each other's data, which could be caused by the fact that a lot of the minimum pressure and maximum wind data may be similar. It is looking at the region of the storm, so the data may overlap in many regions. Although this happened to the first method, the second graph, as shown in Fig B6 b (Appendix B), showed a more improved visualisation of the tree, with the Atlantic data being more on the left with a depth of 1 and demonstrating a higher range in the pressure region with a lower wind speed. The Pacific data with a depth of 0 was on the right, with a higher range in the maximum wind region and a lower minimum pressure region. This data may not have overlap is that the data only looks for specific storm types, which will not have as much overlapping data as a lot of the storms will have different data.

3.2.2 Naïve Bayes

The Gaussian Naive Bayes (GNB) algorithm was used to calculate the correct and mislabelled points. In this method, the dataset was split into two different training-testing configurations, 50-50 and 70-30 split. Then, the Naive Bayes model was tested to check for number of mislabelled points by comparing the predicted labels (y_pred) with the actual labels (y_test), using the expression (y_test != y_pred).sum(), that counted the number of incorrect predictions. After training and predicting, the results from both data split configurations were compared using a bar graph to assess how our model performs with different data.

3.2.3 KNN

K-Nearest Neighbours (KNN) algorithm was applied to classify hurricane data into two categories: Atlantic (with label 0) and Pacific (with label 1) regions. The dataset was first split into training and testing sets using a 70/30 split. The KNN classifier was then trained on the selected features, including Maximum Wind, Minimum Pressure, and Low Wind NE, with the number of neighbours set to 3. After training, the model was evaluated by calculating its accuracy score on the test set and making predictions using new data points. The number of mislabelled points was identified, and was

further evaluated using the classification report, for metrics such as precision, recall, F1-score, and accuracy for both classes ("Atlantic" and "Pacific").

3.2.4 Linear Regression and Logistic Regression

Linear Regression was implemented to analyse the relationship between Minimum Pressure and Maximum Wind. The dataset was first pre-processed by converting the date column into a datetime format and setting it as the index for time-based analysis. Minimum Pressure was considered as the independent variable (feature) and Maximum Wind as the dependent variable (target).

Logistic Regression was used on the hurricane data to visualize the decision boundary and probability contours of the model (see Fig B7 (Appendix B)) and then predict its occurrence. The model was trained using features - Maximum Wind and Minimum Pressure. The target variables were encoded using `LabelEncoder` to transform categorical labels into numeric values to work with the regression model. The dataset was split into 70-30 % and the logistic regression classifier was trained to predict specific hurricane statuses. The performance of the model was evaluated using a classification report, which provided metrics such as accuracy, precision**,** recall, and F1-score. Additionally, a bar chart was generated to compare these metrics, highlighting the effectiveness of logistic regression in classifying hurricanes.

3.3 Neural Network Models for Damage Analysis

3.3.1 CNN

CNN was utilized in image processing, with the model being selected for its architectural advantages in recognizing patterns in image data. The dataset was split as done previously, with the CNN classifier being trained based on the segmented images. The images were passed through the algorithm layer by layer, first with the input layer before the algorithm extracted main features, creating links and flatting the features to fully connect the layers. This process was repeated multiple times to improve the performance of the program, after which the accuracy of the test was extracted.

4.Results and Brief Discussion

4.1 Clustering Results and Interpretation

For clustering the K-Means algorithm with k=3 was used. As shown in the plot (see Fig 1) the clusters showed distinct groupings representing low-intensity, moderate-intensity, and high-intensity storms. There are 3 clusters distributions, marked with purple, yellow and blue. The ones marked in purple represent hurricanes with moderate wind speeds and pressure. These likely represent average hurricanes. Yellow represents hurricanes with higher wind speeds and lower pressure, which likely represents high intensity hurricanes. Blue represents hurricanes with lower wind speeds and higher pressures, which are likely to represent low intensity hurricanes. These features can be used to categorize hurricanes into different intensity levels, and this can help predict hurricanes by checking wind speeds and pressure of winds and whether they form hurricanes and how intense the hurricane might be.

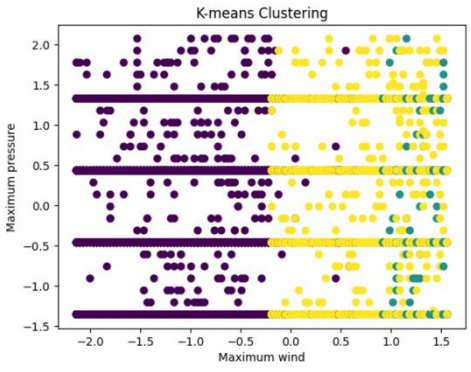


Fig 1: *Clustering using K-Means*

4.2 Predictive Model Results and Performance

The Naive Bayes classifier was done on two different train-test splits for checking for mislabelled values. A bar plot visualizing this result is shown in Fig B8 (Appendix B):

- **70-30 Split**: The algorithm showed a smaller number of mislabelled points; thus, it suggests that it might work better on the test dataset and more optimal to be used for predicting (see Fig 2).
- **50-50 Split**: A higher number of mislabelled points was observed, because of less training data which would impact poorly on the model's performance (see Fig 3).

```
print("Number of mislabeled points out of a total %d points : %d"
      % (X_test2.shape[0], (y_test2 != y_pred2).sum()))
```

```
Number of mislabeled points out of a total 22573 points : 8496
```

Fig 2: *Output showing Number of mislabelled points for a 70-30 train-test dataset using Naïve Bayes Algorithm*

```
print("Number of mislabeled points out of a total %d points : %d"
      % (X_test.shape[0], (y_test != y_pred).sum()))
```

```
Number of mislabeled points out of a total 37621 points : 14000
```

Fig 3: *Output showing Number of mislabelled points for a 50-50 train-test dataset using Naïve Bayes Algorithm*

Using KNN, the code could predict whether the storm was Atlantic or Pacific based on inputted values (Fig 4).

```
[ ] k_nearest_neighbour.predict([[20, 1002]]) # This represents a Atlantic prediction as the output came as 0

array([0])

[ ] k_nearest_neighbour.predict([[25, 1009]]) # This represents a Pacific prediction as the output came as 1

array([1])
```

Fig 4: *Output predicts whether the storm is Atlantic or Pacific based on the values*

Decision Tree classifier was coded to use Maximum Wind and Minimum Pressure as features to predict the location of the storm (Fig 5) and the different types of hurricanes using the values inputted (see Fig 6).

```
Storm_Prediction: [3]

       Latitude Longitude
22332     32.9N     86.5W
31699     25.2N     98.7W
31700     25.3N     99.0W
40358     31.7N     96.9W
43815     35.3N     89.1W
45622     32.7N     88.6W
45679     40.2N     88.7W
45680     42.2N     86.5W
46735     17.5N     92.8W
```

	Latitude	Longitude
22332	32.9N	86.5W
31699	25.2N	98.7W
31700	25.3N	99.0W
40358	31.7N	96.9W
43815	35.3N	89.1W
45622	32.7N	88.6W
45679	40.2N	88.7W
45680	42.2N	86.5W
46735	17.5N	92.8W

```
Storm_Prediction: [1]

      Latitude Longitude
0        28.0N     94.8W
1        28.0N     95.4W
2        28.0N     96.0W
3        28.1N     96.5W
4        28.2N     96.8W
...        ...       ...
15815    25.8N    173.6E
15853    32.7N    177.4W
15889    16.5N    147.0W
17059    29.9N    129.7E
17343     6.7N    169.8E

[1542 rows x 2 columns]
```

	Latitude	Longitude
0	28.0N	94.8W
1	28.0N	95.4W
2	28.0N	96.0W
3	28.1N	96.5W
4	28.2N	96.8W
...	...	...
15815	25.8N	173.6E
15853	32.7N	177.4W
15889	16.5N	147.0W
17059	29.9N	129.7E
17343	6.7N	169.8E

1542 rows × 2 columns

Fig 5: *Predicating the location of the storm*

```
[ ] tree_hurricane_identifier.predict([[20, 1002]])# based off the dataset it shows that its a Tropical Depression

array([3])

[ ] tree_hurricane_identifier.predict([[80, 0]]) #Based off the dataset from these values the array of 1 shows its a hurricane

array([1])
```

Fig 6: *Results of using Decision Tree Classifier to Predict the Hurricane Type based on Wind and Pressure values*

Similarly, Logistic Regression classifier is used in predicting hurricane types and visualizing it belonging to specific statuses (e.g., Hurricane, Tropical Storm; see Fig 7).

```
example = [[1000, 100]]  # Using Example Value
predicted_type = classifier.predict(example)
print(f"Predicted Hurricane Type : {predicted_type}")
```

```
Predicted Hurricane Type : [1]
```

Fig 7: *Results of using Logistic Regression classifier to predict hurricane type based on Wind and Pressure values*

For Linear Regression, to interpret the results, the actual, predicted values and prediction errors for maximum wind(see Fig 8) were printed. For better visualization, a scatter plot was also drawn(see Fig B10 (Appendix B)), with actual values plotted as points and the regression line overlayed in red.

	Actual Maximum Wind	Predicted Maximum Wind	Error
0	80	51.927395	28.072605
1	80	51.927395	28.072605
2	80	51.927395	28.072605
3	80	51.927395	28.072605
4	80	51.927395	28.072605
...	...	...	...
49100	55	52.131997	2.868003
49101	55	52.132414	2.867586
49102	50	52.132831	-2.132831
49103	45	52.132831	-7.132831
49104	45	52.133249	-7.133249

[49105 rows x 3 columns]

(a)

	Actual Maximum Wind	Predicted Maximum Wind	Error
0	45	50.295555	-5.295555
1	45	50.295555	-5.295555
2	45	50.295555	-5.295555
3	45	50.295555	-5.295555
4	45	50.295555	-5.295555
...	...	...	...
26132	35	47.637965	-12.637965
26133	30	47.624704	-17.624704
26134	30	47.622052	-17.622052
26135	25	47.619399	-22.619399
26136	20	47.616747	-27.616747

(b)

Fig 8: *Actual Wind values, Predicted Wind values and Prediction Errors (a) Atlantic (b)Pacific*

The values for the intercept, coefficient, R^2, and Mean Squared Error (MSE) were printed to understand the linear regression model's performance its ability to predict Maximum Winds (see Fig 9).

```
Linear Regression Score (R^2): 1.3113275888154696e-05
Intercept: 51.92739459193165
Coefficient: [0.00020857]
Mean Squared Error: 766.2620365261442
```

(a)

```
Linear Regression Score (R^2): 0.0027135141849548017
Intercept: 50.295555340485386
Coefficient: [-0.00265229]
Mean Squared Error: 640.0682472016902
```

(b)

Fig 9: *Values for the intercept, coefficient, R^2, and Mean Squared Error (MSE) (a) Atlantic (b) Pacific*

Based off the performance of the multiple models using the classification report it demonstrates that the K-Nearest Neighbour (KNN) has a better overall performance with precision, recall, f1 score and accuracy. Although the KNN model has a better overall performance, it is very computationally heavy with larger models due to the need to calculate the distance between each dataset, which can lead to a slower prediction model [5]. During the coding, this was experienced when trying to use the KNN *pipe* import and the *NeighborhoodComponentsAnalysis* import, which ended in a long computational issue where other types of KNN training couldn't be achieved using the KNN import. The second-best predictive model was the Decision Tree, which achieved a similar accuracy score as KNN, but it had a lower score in the other sections (See Fig 10, 11, 12).

BY_Pred

	precision	recall	f1-score	support
Atlantic	0.68	0.79	0.73	14677
Pacific	0.45	0.31	0.37	7896
accuracy			0.62	22573
macro avg	0.56	0.55	0.55	22573
weighted avg	0.60	0.62	0.60	22573

DT_Pred

	precision	recall	f1-score	support
0	0.68	0.90	0.77	14677
1	0.52	0.20	0.29	7896
accuracy			0.66	22573
macro avg	0.60	0.55	0.53	22573
weighted avg	0.62	0.66	0.60	22573

KNN_Pred

	precision	recall	f1-score	support
0	0.71	0.81	0.75	14677
1	0.51	0.37	0.43	7896
accuracy			0.66	22573
macro avg	0.61	0.59	0.59	22573
weighted avg	0.64	0.66	0.64	22573

(a)

BY_Pred

	precision	recall	f1-score	support
0	0.65	1.00	0.79	14677
1	0.00	0.00	0.00	7896
accuracy			0.65	22573
macro avg	0.33	0.50	0.39	22573
weighted avg	0.42	0.65	0.51	22573

(b)

Fig 10: *Evaluation matrices for Experimental Method for Predicting Hurricane on Atlantic(0) and Pacific (1) data using (a) Naive Bayes, Decision Tree and KNN (b) Logistic Regression*

BY_Pred

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1439
1	0.98	1.00	0.99	6468
2	0.87	1.00	0.93	8341
3	0.94	0.61	0.74	5016
4	0.00	0.00	0.00	34
5	0.26	0.94	0.41	863
6	0.00	0.00	0.00	178
7	0.00	0.00	0.00	88
8	0.00	0.00	0.00	102
9	0.00	0.00	0.00	43
10	0.00	0.00	0.00	1
11	0.00	0.00	0.00	0
accuracy			0.83	22573
macro avg	0.25	0.30	0.26	22573
weighted avg	0.82	0.83	0.81	22573

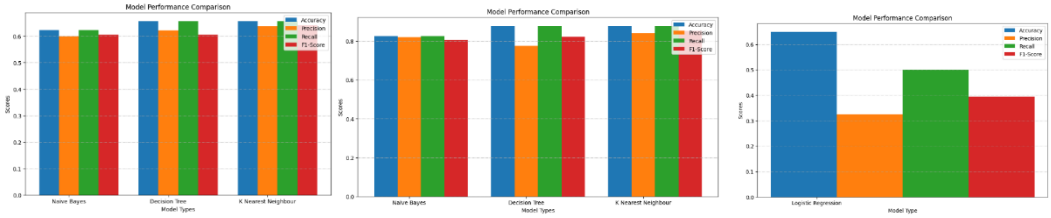
DT_Pred

	precision	recall	f1-score	support
0	0.00	0.00	0.00	1439
1	0.98	1.00	0.99	6468
2	0.87	1.00	0.93	8341
3	0.79	1.00	0.88	5016
4	0.00	0.00	0.00	34
5	0.00	0.00	0.00	863
6	0.00	0.00	0.00	178
7	0.00	0.00	0.00	88
8	0.00	0.00	0.00	102
9	0.00	0.00	0.00	43
10	0.00	0.00	0.00	1
accuracy			0.88	22573
macro avg	0.24	0.27	0.25	22573
weighted avg	0.78	0.88	0.82	22573

KNN_Pred

	precision	recall	f1-score	support
0	0.44	0.14	0.21	1439
1	0.98	1.00	0.99	6468
2	0.88	0.99	0.93	8341
3	0.85	0.92	0.88	5016
4	0.00	0.00	0.00	34
5	0.44	0.38	0.41	863
6	0.00	0.00	0.00	178
7	0.00	0.00	0.00	88
8	0.00	0.00	0.00	102
9	0.00	0.00	0.00	43
10	0.00	0.00	0.00	1
11	0.00	0.00	0.00	0
accuracy			0.88	22573
macro avg	0.30	0.29	0.29	22573
weighted avg	0.84	0.88	0.85	22573

Fig 11: *Evaluation matrices for Experimental Method for Predicting Different Types of Hurricanes using Naive Bayes, Decision Tree and KNN*



(a) (b)

Fig 12: *Evaluating and comparing the models in terms of appropriate metrics (a) Naive Bayes, Decision Tress and KNN (b) Logistic Regression for Different Experimental Methods*

4.3 Neural Network Results and Performance

This confusion matrix shows the performance of a classification model predicting **"damage"** and **"no damage"** in areas affected by hurricane shown in our image dataset. Thus, the model performs well overall (see Fig 13). The accuracy of the test was also extracted, with the final output returning 86.8%.

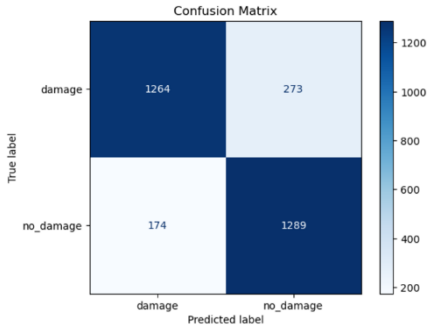


Fig 13: *Confusion Matrix of Image Datasets*

5. Conclusion

The objectives of the coursework were met by using various machine learning techniques to predict and analyse hurricane behaviour based on the key features. Clustering with **K-Means** managed to group hurricanes into three distinct intensity categories. It helped us learn patterns that differentiated winds for different storm severity levels. Predictive models- **Naive Bayes**, **Decision Trees**, **KNN**, **Linear and Logistic Regression**, were implemented to classify hurricanes and write other methods and evaluation was done to analyse their performance.

References

[1] NOAA, “Hurricanes and Typhoons, 1851-2014,” *Kaggle.com*, 2014. https://www.kaggle.com/datasets/noaa/hurricane-database/data (accessed Nov. 19, 2024).

[2] “Page Restricted,” *Kaggle.com*, 2024. https://www.kaggle.com/datasets/kmader/satellite-images-of-hurricane-damage/code (accessed Nov. 19, 2024).

[3] N. US Department of Commerce, “Saffir-Simpson Hurricane Scale,” www.weather.gov. https://www.weather.gov/mfl/saffirsimpson

[4] “Decision Tree,” *CORP-MIDS1 (MDS)*. https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree/

[5] Samarpit Nandanwar, “Understanding K-Nearest Neighbours: A Comprehensive Guide,” DEV Community, Aug. 27, 2024. https://dev.to/samarpitnandanwar/understanding-k-nearest-neighbors-a-comprehensive-guide-lm9 (accessed Nov. 22, 2024).

APPENDIX A: DETAILED EXPLANATION OF THE DATASETS

A1: NUMERICAL DATA

These datasets contains the following key columns: “Date and Time” is specified for when the data was recorded that helps in understanding storm progression , “Event and Status” that is used for classification (e.g., tropical cyclone, hurricane), “Latitude and Longitude” which would help us in mapping the storm trajectory, “Maximum Wind” records the maximum wind speed (in knots) and “Minimum Pressure” measuring the central pressure of the storm (in millibars) . These have been used by us as target variables for predicting hurricanes. Additionally, “Cyclone Size Metrics” provides info about spatial extent (e.g., NE, NW, SE, SW) and strength (e.g., low, moderate, and high winds).

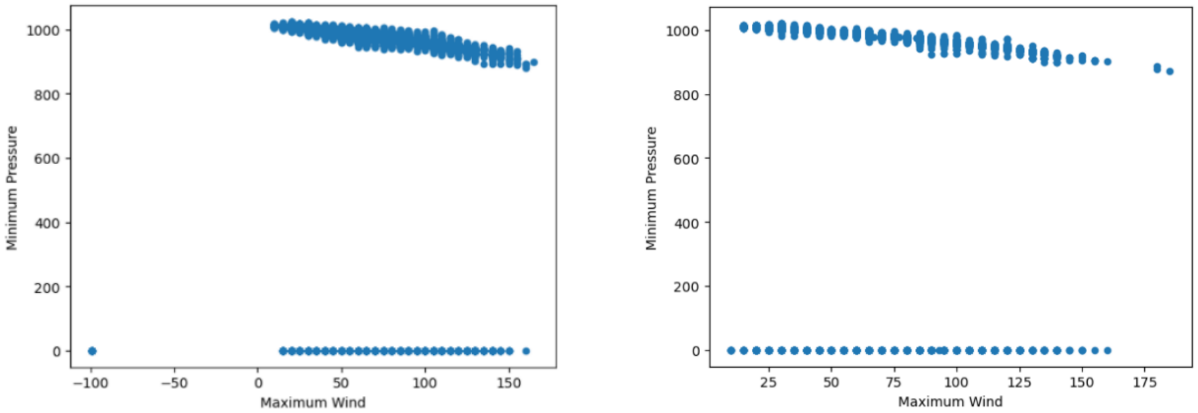
As mentioned above, for this project two target we focused on are—**Maximum Wind** and **Minimum Pressure**—to predict the occurrence and behaviour of hurricanes. By identifying the strength and patterns in wind speeds and pressure, it can be classified whether a storm meets the criteria for a hurricane. For example, high winds with speed 74 mph (64 knots) with high drop in pressure typically indicate hurricane formation [3]. The various algorithms used by us to create predictive models will use these variables, along with other info, to predict the likelihood and intensity of hurricane formed.

A2: IMAGE DATASET

The dataset contains satellite images showing locations in Texas in both urban and rural scenarios, with the dataset consisting of ‘damage’ and ‘no damage’ scenarios in these regions. This allows the code to distinguish between areas damaged by Hurricane Harvey and those that remain unaffected. The images in the dataset were formatted as .jpeg files, with a resolution of 128 by 128 pixels. While this may seem low quality, the images were taken by satellite, and basing the image processing code on this dataset allows for further applications using similar inputs, allowing for a greater adaptability in disaster zones.

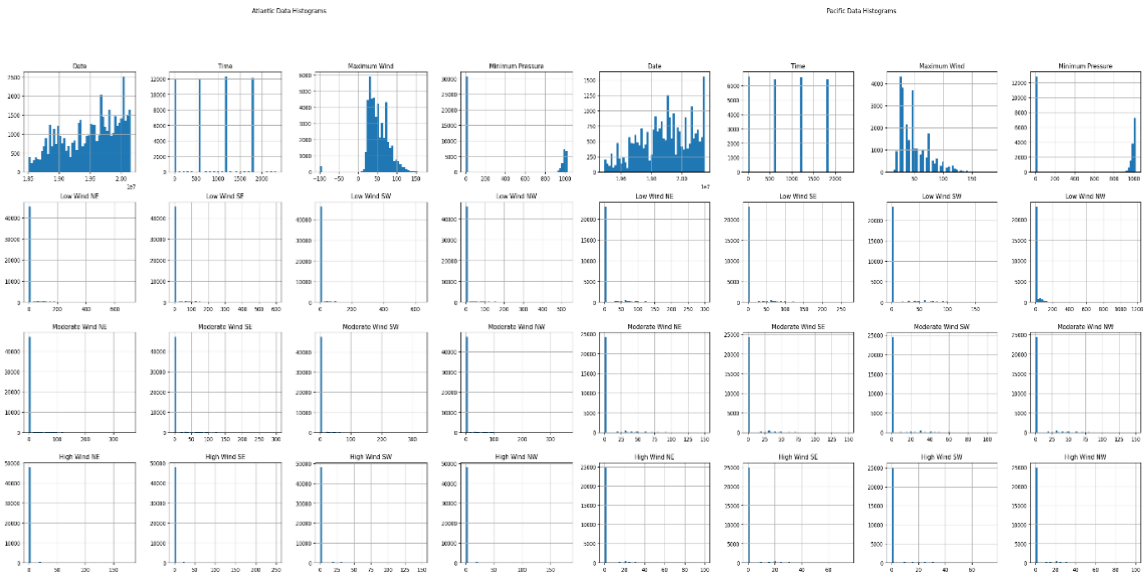
APPENDIX B: FIGURES AND GRAPHS

Fig B1:Scatterplots (a) Atlantic (b) Pacific



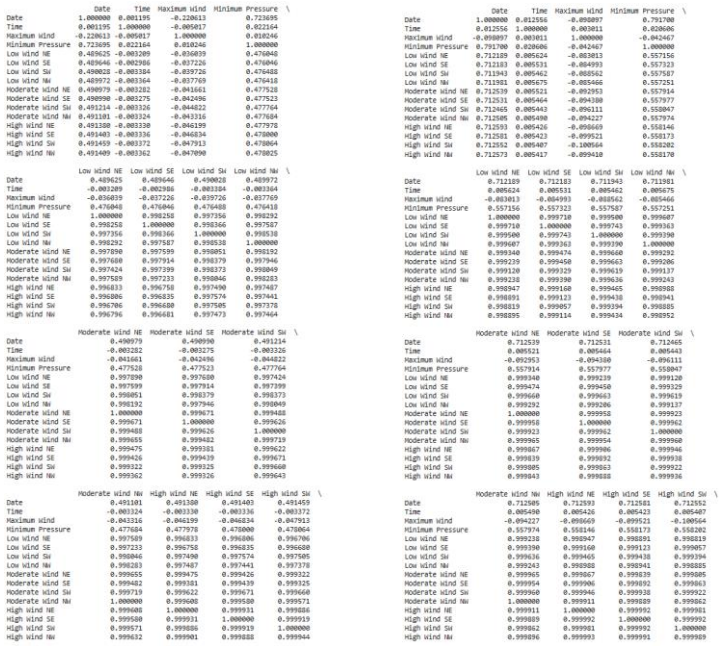
(a) (b)

Fig B2: Histograms (a) Atlantic (b) Pacific



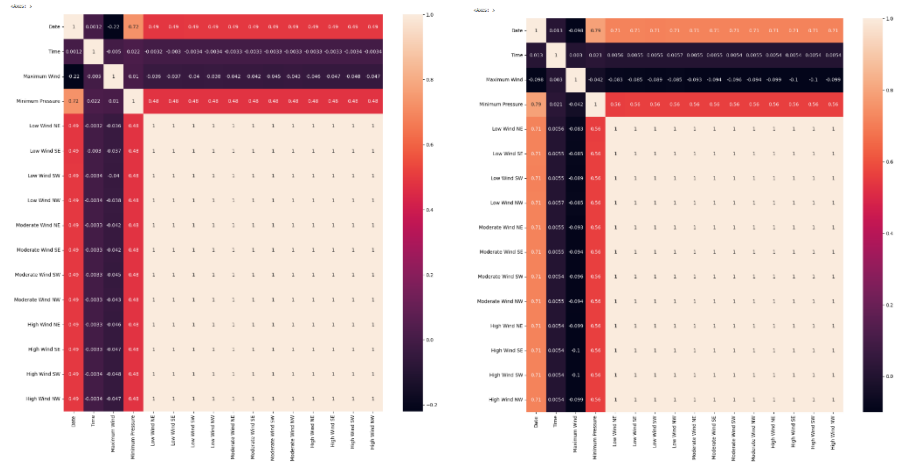
(a) (b)

Fig B3: Correlation Matrix: (a) Atlantic (b) Pacific



(a) (b)

Fig B4: Heat Map (a) Atlantic (b) Pacific



(a) (b)

Fig B5 : Decision Tree for classifying Atlantic and Pacific Storms

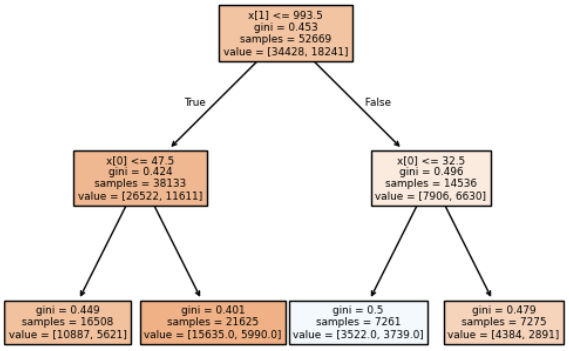
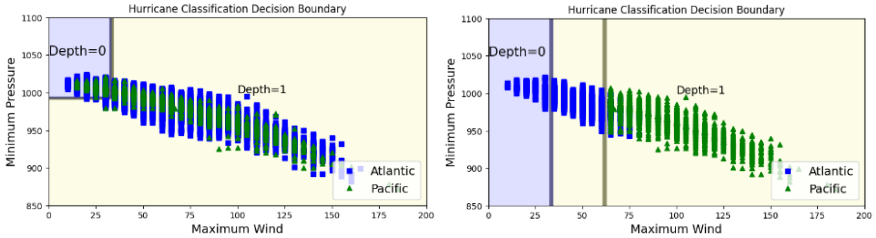


Fig B6 : Decision Boundary for classifying Atlantic and Pacific Storms



(a) (b)

Fig B7 : Using Logistic Regression to Plot Hurricane Classification Probability

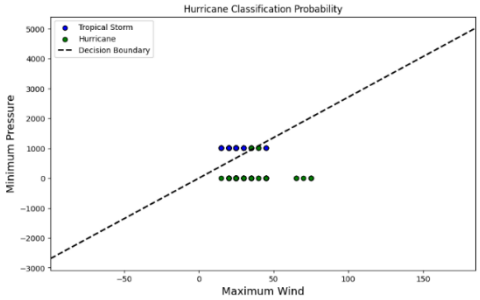


Fig B8 : Bar Graph showing Correct and Mislabelled Values using Naive Bayes Algorithm

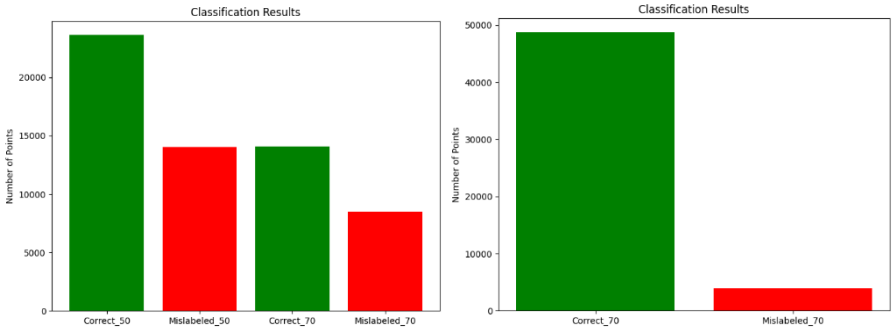


Fig B7 : Using Logistic Regression to Plot Hurricane Classification Probability

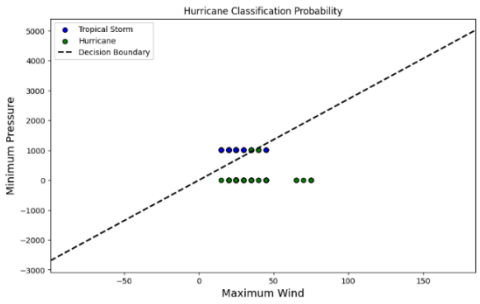
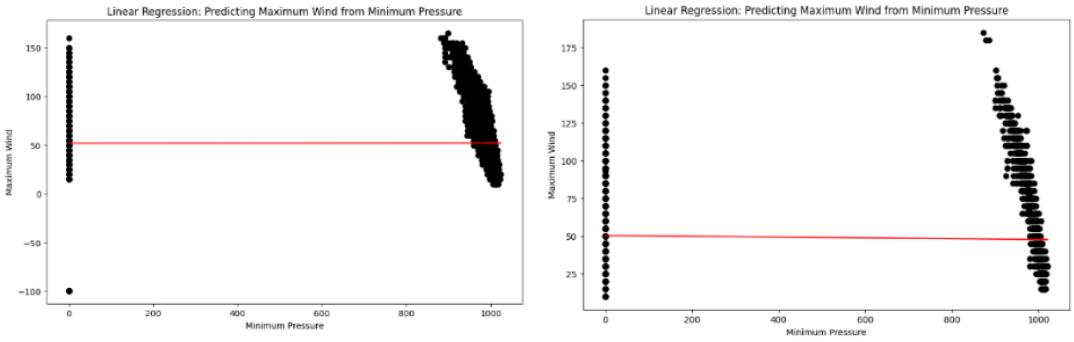


Fig B10: Scatter Plot (a) Atlantic (b) Pacific



(a) (b)

Fig B11: *Graph Plotting the Training and Testing Accuracy of Image Processing Model*

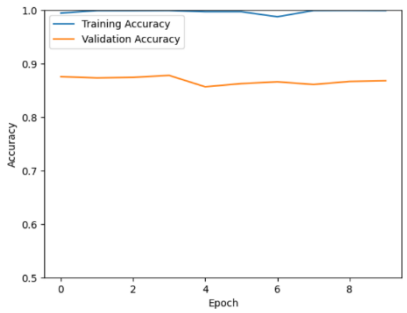
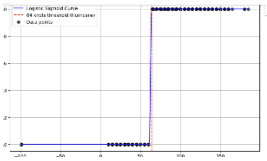


Fig B12 : Encoding different types of hurricanes

```
'EX': 0,    # No Storm
'HU': 1,    # Hurricane
'TS': 2,    # Tropical Storm
'TD': 3,    # Tropical Depression
'WV': 4,    # Tropical Wave
'LO': 5,    # Low Pressure
'SS': 6,    # Subtropical Storm
'DB': 7,    # Disturbance
'SD': 8,    # Subtropical Depression
'ET': 9,    # Extratropical Transition
'PT': 10,   # Post Tropical
'ST': 11    # Storm
```

B13 : Probability of Hurricane using Wind Speed



APPENDIX C : LIMITATIONS AND CHALLENGES

As mentioned in the results section the KNN model although it might have had a better performance overall it did have some issues where due to the large dataset the KNN model could not load in specific code and was not able to train using other KNN imports and were only able to use the regular KNN model. ,

Some issues were faced with the coding due to the fact that all the models are meant for purely numerical values but there were a lot of non-numeric values to deal with in the datasets, as well as values that had trailing white space, and some columns had values that had numeric values followed or preceded by a non-numeric value, which would also not work with the code. Some lines of code were added to strip white spaces and remove non-numeric columns to deal with these issues.

During the splitting of the data to have a training dataset and a testing dataset, we did not have enough values to split the data evenly.