Predicting Paths

for Tropical Cyclones

via Machine Learning

By: Adam Scott  
ID: 1423886

**Table of Contents**

1. Letter of transmittal ……………………………………………………………….. 3

A.1. Problem Summary………………………………………………………………....4

A.2. Product Benefits ………………………………………………………………….5

A.3. Data Product Outline……………………………………………………………...6

A.4. Objective and Hypothesis…………………………………………………………6

A.5. Project Methodology ……………………………………………………………..7

A.6. Funding …………………………………………………………………………..7

A.7. Stakeholder Impact……………………………………………………………….8

A.8. Ethical and Legal Precautions……………………………………………………9

A.9. Expertise…………………………………………………………………………9

B. Executive Summary………………………………………………………………..9

B.1. Problem Summary ………………………………………………………………10

B.2. Customer Description……………………………………………………………11

B.3. Existing Systems………………………………………………………………...12

B.4. Data Systems…………………………………………………………………….13

B.5. Methodology…………………………………………………………………….14

B.6. Outcomes………………………………………………………………………..15

B.7. Implementation………………………………………………………………….17

B.8. Evaluation Plan………………………………………………………………….18

B.9. Programming Environment, Costs, and HR Requirements …………………….19

B.10. Timeline and Milestones……………………………………………………….21

C. Application Files…………………………………………………………………..21

D.1. Project Purpose………………………………………………………………….22

D.2. Datasets………………………………………………………………………….22

D.3. Product Code…………………………………………………………………….23

D.4. Hypothesis Verification………………………………………………………...27

D.5. Visuals and Data Exploration…………………………………………………..32

D.6. Accuracy………………………………………………………………………..34

D.7. Testing………………………………………………………………………….34

D.8. Installation Guide………………………………………………………………35

D.9. User Guide……………………………………………………………………...35

D.10. Final Thoughts………………………………………………………………...36

E. References………………………………………………………………………...39

**A. Letter of Transmittal**

Friday, March 3, 2023

To: Joseph Mercer

Chief Information Technology and Research Officer

11691 SW 17th St, Miami, FL 33165

Subject: Tropical Storm Prediction Model Proposal

Dear Joseph Mercer

I am writing to present our software solution for predicting the future path of tropical storms affecting the Atlantic and Gulf coasts. In recent years we have had an influx of tropical storm activity with limited preparedness available. these storms begin affecting communities along the Gulf and Atlantic Coastlines. We plan to develop a software solution to assist in preparing the general public for potential impacts from storms that may affect us all in the upcoming season. This will allow emergency management organizations to prepare in advance. This product will have the benefit of predicting the paths of storms during Hurricane season allowing media outlets to notify the public in advance.

This software will be useful by providing valuable insights and information that can guide the decision-making process. The software will perform various data analyses, such as predictive modeling and data visualization, which can help identify patterns, trends, and relationships within the data. This information will be used to make informed decisions based on the data, rather than relying solely on intuition alone.

Our solution will consist of two modules, one for viewing historical storm tracks and another for generating predictions. The historical data viewer uses the MapPlot library to display past storm tracks, providing valuable insights into the occurrence and movement of hurricanes over time. The prediction module uses machine learning techniques, specifically a Decision Tree Regressor model, to generate forecasts for four key storm attributes: latitude, longitude, wind speed, and pressure. We will also create a CSV file to record the predictions generated by the software.

To develop this tool, we will collect data from the National Oceanic and Atmospheric Administration (NOAA) and develop this software using python. We plan to use the Scikit-learn library for machine learning and the Folium library to display storm tracks on maps. Our software will be accurate, providing valuable insights into the likelihood and impact of future hurricanes.

We believe that our software is a significant contribution to our mission of keeping the public Informed and safe during natural disasters. This will enable the public and emergency management organizations to prepare and respond effectively. While this software is a beta version of a standalone application, the development and maintenance cost of this product is estimated to be $100,000 for the first year and an additional $25,000 for each year after that.

We are excited to present this solution to you and hope that it meets your expectations. Please do not hesitate to contact us if you have any questions or require further information.

Sincerely,

Adam Scott

**A.1. Problem Summary**

The United States is frequently affected by hurricanes and recently there has been an influx of the frequency of these storms. These storms can result in significant damage and loss of life spanning across the southern and eastern United States. While there are existing methods for predicting hurricanes, there is a need for more accurate and accessible information to help the general public prepare for potential impacts of storms. To address this issue, we propose developing software that predicts the future path of hurricanes affecting the US. This software will provide timely and accurate information to the public, allowing them to prepare for the hurricane season effectively and reduce the impact of these storms.

**A.2. Product Benefits**

The benefits of this software are listed below.

* Early warning: The software can predict the future path of hurricanes before they happen, providing the public with advanced notice to prepare for storms heading their way. This could help save lives and minimize damage to property.
* Improved preparedness: With advanced notice, people can take measures to secure their homes, stock up on supplies, and make evacuation plans, leading to more effective preparedness and response to hurricanes.
* Increased safety: By providing accurate and timely information, the software can help keep people safe during hurricane season.
* Cost savings: Early preparation and warning could potentially save people money in terms of property damage and loss.
* Improved decision-making: With access to reliable information about the future path of hurricanes, individuals and organizations can make informed decisions about their plans and actions during hurricane season.

This software would support the decision-making process by providing accurate and timely information to individuals, businesses, and government agencies. With access to information on the future path of hurricanes, users can make informed decisions on whether to evacuate, prepare their homes and businesses, or take other necessary actions to protect themselves and their property. This information can also help emergency management officials allocate resources and coordinate responses more effectively, potentially reducing the impact of hurricanes on affected communities. Additionally, the software can provide a long-term view of hurricane impacts, enabling users to plan and prepare for the hurricane season more effectively.

**A.3. Data Product Outline**

The application will allow users to not only view historical data of all hurricanes and their paths but will also use this data to predict storms and display the prediction on a map in real time. Additionally, it will have the ability to display data plots regarding its accuracy and the intensity of the storms it has predicted. The data being used for this project was obtained from the NOAA (National Oceanic and Atmospheric Administration). It consists of data pertaining to all storms from 1970 to present day. The data in this file documents the latitude and longitude of each storm in 6-hour intervals. It has the time and date the data was recorded as well as the wind speed and atmospheric pressure of each storm at the time. The proposed machine learning model will take into consideration the location, wind speed, and pressure of each historical storm throughout their lives and predict the path of storms that may affect the United States in the future. As more data is added to this model, the more accurate it will become.

**A.4. OBJECTIVE AND HYPOTHESIS**

The software will use advanced data analytics and artificial intelligence to analyze real-time emergency response data, including information about accidents, natural disasters, and other emergencies. By processing this data in real-time, the software will be able to help emergency responders quickly identify and respond to emergencies, potentially saving lives and reducing response times. Additionally, the software will also provide tools and resources to help educate the general public about emergency preparedness, including information about evacuation routes, emergency supplies, and other important topics. By empowering the public to better understand and prepare for emergencies, the software could also help minimize the impact of emergencies and reduce the number of lives lost.

**A.5. Project Methodology**

This project will implement the AGILE methodology. It will include 3 primary components to include front-end development regarding the GUI and visuals, back-end development pertaining to data collection, cleaning and implementation, and the machine learning model development. Using these components, the development flow will contain a planning phase, iterative phase split into 2 week sprints, a deployment phase where the software will be tested and put into use, and a maintenance phase where the software will have updates over time as well as bug fixes. Each phase will have its own life cycle including development,  
software testing, software integration, software deployment, and user feedback. Using the AGILE approach, we will be able to create this software solution quickly as well as develop new features and turn out bug fixes on a regular basis.

**A.6. Funding**

An outline of funding requirements and their cost are listed below.

1. Personnel Costs: The project will require a team of developers, project managers, and quality assurance engineers. The personnel costs for this team will be estimated at $500,000 per year.
2. Equipment and Infrastructure Costs: The project will require a development environment, cloud hosting, and other necessary infrastructure. The costs for equipment and infrastructure are estimated at $150,000.
3. Software and Licensing Costs: The project will require software licenses and other tools to support the development process. The costs for software and licensing are estimated at $50,000.
4. Testing and Quality Assurance Costs: The project will require rigorous testing and quality assurance to ensure that the software meets the required standards. The costs for testing and quality assurance are estimated at $100,000.
5. Miscellaneous Costs: The project may require additional costs such as travel, training, and other miscellaneous expenses. The miscellaneous costs are estimated at $25,000.

In total, the estimated funding requirement for this project is $825,000.

**A.7. Stakeholder Impact**

The proposed software solution will impact a variety of stakeholders. Investors will have a clear increase of revenue over the course of their involvement with this project. They will also gain credibility and appreciation from the general public as the software continues to grow and prepare countless individuals from a potential disaster. Other benefits include emergency response being able to respond more quickly overall and local governments will have advanced knowledge to prepare for these storms well in advance.

**A.8. Ethical and Legal Precautions**

The data being used in this software solution is readily available to the public and does not pose any risk with personal information. This software and data set contains no risk for ethical or legal concerns since

**A.9. Expertise**

This application will be led by a senior software developer with experience using machine learning. This individual has over 10 years’ experience in this field. Additionally, this leader will be assisted by 6 other developers with ranging experience. 3 of these developers will be individuals with over 5 years’ experience in this field as well as having in depth understanding of machine learning models, 2 of them will be developers with 2 years’ experience using web development and python library tools and one new hire in order to give this individual a learning experience allowing them to grow in this field.

**B. Executive Summary**

**B.1. Problem Summary**

Tropical storms have been a major threat to the general public in many parts of the world. The impact of these storms is often severe, resulting in damage to infrastructure, displacement of people, loss of lives, and disruption of critical services such as power, water, and transportation. The effects of these storms are exacerbated by climate change, which has led to more frequent and intense storms in recent years. Additionally, emergency response to tropical storms is often complex and challenging, with emergency responders facing numerous challenges such as limited resources, inadequate information, and unpredictable weather patterns. This can result in delayed response times and inadequate assistance to affected populations. Machine learning software using historical data has the potential to assist in improving the emergency response to tropical storms. By analyzing historical data on storm patterns and intensity this software can help emergency responders to better understand the nature of the storm and its potential impact on the affected population. This can help emergency responders to make more informed decisions and to allocate resources more effectively, resulting in a more efficient and effective emergency response. Additionally, the software can also help to prepare the general public better by providing them with more accurate and timely information on the storm and its potential impact, enabling them to take necessary precautions and to make better decisions for their safety. This machine learning software can also be a valuable tool for governmental agencies. By using historical data and real-time information, the software can help agencies predict and respond to tropical storms more effectively. This can result in better preparation, more efficient allocation of resources, and ultimately, increased public safety. Overall, the machine learning software has the potential to be a valuable asset for governmental and public agencies to assist in the efforts to prepare for and respond to tropical storms.

**B.2. Customer Description**

Our agency responsible for dealing with tropical storms and preparedness. We need a comprehensive and effective solution to better understand and respond to tropical storms. This software will provide us with the ability to analyze and visualize historical data, assess potential impacts, and make informed decisions to prepare for and respond to tropical storms. Using machine learning algorithms to analyze historical data on tropical storms, such as their paths, wind speeds, and atmospheric pressure we can identify patterns and trends that can help predict future storms. This will enable us to make more informed decisions about evacuations, sheltering, and other preparations. Overall, the software will fulfill our needs by providing us with the tools and information necessary to effectively respond to tropical storms and minimize their impact on the public and infrastructure.

**B.3. Existing Systems**

Currently, there are a few established hurricane tracking systems in place, including the National Hurricane Center (NHC) in the United States, the Joint Typhoon Warning Center (JTWC) in the Pacific, and the European Centre for Medium-Range Weather Forecasts (ECMWF). These organizations use historical data and current weather patterns to predict the path of a storm, estimate its strength, and issue warnings to the public in affected areas.

However, these tracking systems are not perfect, and there are limitations to their accuracy. For example, they may not always capture the nuances of the weather patterns that could lead to a tropical storm or hurricane. Additionally, they may not always be able to predict the intensity or exact path of the storm, which can be critical in terms of preparedness and response efforts.

Having software that utilizes machine learning to analyze historical weather patterns and current conditions could be extremely beneficial in predicting the future path of storms. This software could be used to identify patterns and trends in weather data that may indicate the formation of a tropical storm or hurricane. By analyzing large amounts of data, the software could potentially predict the path, intensity, and timing of a storm more accurately than current tracking systems.

This improved accuracy could lead to more effective preparedness and response efforts, as emergency responders and governmental agencies would have more lead time to prepare for an incoming storm. It could also help the public better understand the potential risks associated with a storm and make more informed decisions about evacuations and other safety measures. Overall, having software that predicts the future path of storms could greatly benefit both governmental agencies and the general public in terms of mitigating the impacts of tropical storms and hurricanes.

**B.4. Data Description**

This software requires a large data set for it to be in any way reliable. Fortunately, the NOAA has a data set that has logged tropical storms and their features since 1851 to present day. However, the data from that long ago is somewhat unreliable so for the purposes of this software solution, the data being used will span back to 1970. The data in this CSV file (which is regularly updated) contains information about each named storm, its path including latitudes and longitudes, wind speeds at every data point and atmospheric pressure. This data was recorded at 6-hour intervals and has information regarding each storm through the storm’s entire life cycle.

**B.5. Methodology**

The project development methodology for this program will follow an AGILE approach, each sprint will last three weeks. Below is a description of each phase of the project

**Concept Assessment** This phase will focus on assessing how realistic the goals of the project are as well as finalizing the project scope as well as developing the business and GUI design requirements. Additionally, the team will work to identify key historical data sets to be used in developing the predictive algorithms and create a roadmap while finalizing the scope of this project.

**Iteration -** During this phase, the actual software application will be developed using the chosen requirements and framework from the previous phase. This phase will be broken down into several sprints, each building upon the previous one. These sprints are explained below.

**Sprint 1 –** Gather all required data sets and required software while establishing which algorithms would likely be best to use in this scenario using an analysis of the historical data

**Sprint 2 -**  Software will be developed to include machine learning algorithms and the GUI. We will focus on major requirements and functionality primarily and then move on to additional features and quality of life enhancements

**Sprint 3 -**  being evaluated by users, grey box testing will be implemented and major bugs will be fixed in this phase as well as any final additions to features and optimizations being made.

**Sprint 4 –** The program will be sent to stakeholders for testing in a beta environment in order to get their feedback and opinions on the software solution

**Sprint 5 -** User feedback and assessment on the program will be documented as well as any bugs that users report will be reviewed and documented for the next iteration of the product

**Deployment –** The program will be put into production and released for full use. IT infrastructure will be evaluated with the intent to gather more resources to include required hardware, software, and licenses needed for deployment

**Maintenance -** Continued support for new bugs or modifications will be given after the initial release. Maintenance will shadow the iteration phase as described above with ongoing refinement of the predictive algorithms based on feedback and real-world usage.

**B.6. Outcomes**

The software solution will be easy to operate and interpret by anyone using this tool. It will allow users to view the future path of current storms and will change as historical data on current storms are added to the database. This will give the users a live view of what may be to come and give warning to the populations in the potentially affected areas. This product will come with both project deliverables and product deliverables. These components are listed below.  
**Project Deliverables**

1. Summary of hardware, software, and licensing agreements
2. Development schedule
3. User and Installation Guide

**Product Deliverables**

1. Front-end application code
2. Back-end application code
3. User authentication code
4. Machine learning model code
5. Updated CSV data files used for predictions

**B.7. Implementation and Outcomes**

**Implementation:**

After the product has been delivered to stakeholders, the implementation plan will include several key components. Firstly, the standalone software app will need to be deployed and configured on the appropriate infrastructure, which will depend on the specific requirements and resources available. This may include cloud-based solutions such as AWS or Azure, or on-premises servers or data centers.

**Rollout Phases:**

The rollout phases will include two main stages: deployment and maintenance. During the deployment phase, the initial release of the software will be rolled out to users, with a focus on ensuring a smooth and seamless transition. This may involve conducting user acceptance testing, performing compatibility testing, and ensuring that all necessary configurations and integrations are in place.

Once the initial rollout is complete, the maintenance phase will begin. This phase will involve ongoing support and maintenance of the software, including bug fixes, updates, and upgrades. Feedback from stakeholders will be collected and analyzed, and changes or improvements to the software will be made as necessary.

**Continuous Testing:**

Continuous testing will be an important aspect of the implementation plan, with the goal of ensuring that the software is functioning correctly and meeting the needs of stakeholders. Automated testing tools and scripts will be used to perform testing on a regular basis, while manual testing may also be necessary for more complex issues or scenarios.

**Distribution:**

Distribution of the software will be managed through a centralized portal, which will be accessible to authorized users. Security and authentication measures will be implemented to ensure that only authorized users can access the software, and that data and information are kept secure and confidential. User training and support will also be provided to ensure that stakeholders are able to use the software effectively and efficiently.

**Outcomes:**

* The standalone software app will provide accurate predictions of the path of future tropical storms, helping individuals and government agencies to better prepare for and respond to potential disasters.
* The app will reduce emergency response times, save lives, and minimize damage to property and infrastructure.
* Continuous testing and maintenance will ensure that the app continues to function properly and meet the evolving needs of users over time.

**B.8. Evaluation Plan**

Validating and verifying that the developed data product meets the requirements and subsequently the needs of the customers is crucial for ensuring that the product is effective and successful. There are several methods that can be used to validate and verify the product, including:

1. User Acceptance Testing (UAT): UAT involves testing the product with end-users to ensure that it meets their needs and expectations. This can involve conducting surveys, focus groups, or one-on-one interviews with users to gather feedback and identify any areas that need improvement.
2. Functional Testing: Functional testing involves testing the product to ensure that it meets the functional requirements outlined in the project specifications. This can involve testing individual features or components of the product to ensure that they are working as expected.
3. Performance Testing: Performance testing involves testing the product to ensure that it meets performance requirements, such as speed and responsiveness. This can involve testing the product under different workloads to ensure that it can handle peak usage and is scalable.
4. Security Testing: Security testing involves testing the product to ensure that it meets security requirements and is protected against cyber threats. This can involve testing the product for vulnerabilities, conducting penetration testing, and implementing security measures to protect against attacks.
5. Regression Testing: Regression testing involves testing the product after updates or changes have been made to ensure that existing features and functionality have not been impacted. This can involve testing the product against a set of predefined test cases to ensure that it is working as expected.

By using a combination of these methods, developers can ensure that this product meets the requirements and needs of stakeholders. This helps to ensure that the product is effective and user-friendly leading to a successful product launch and satisfaction

**B.9. Programming Environment, Costs, and HR Requirements**

**Programming Environment**

Version of Python – Version 3.9

Libraries: pandas, folium, sklearn, numpy, pickle, tkinter, mpl\_toolkits, matplotlib

Version of Operating System – Windows 10 Pro x64

Hardware Description: CPU – Ryzen 7 2700X 8 Core Processor

GPU – Nvidia RTX 2060 Super

Memory – 16GB RAM

**Costs**

* Amazon Cloud hosting: $25,000
* Windows Server 2020 License: $24,000
* PyCharm Professional: $250 per developer, per year
* Payroll overhead: $120,000 (4 Software developers, full time per year)
  + 2 part time developers for 6 months: $15,000

First year Total: $185,000

**Human resources**

The required team for this project requires 4 full time developers and 2 part time developers. This work will be split between them. The part time developers will be required for the first 6 months of this project, essentially until initial completion of the products development. After that, 4 full time developers will work to maintain and continue to improve this software solution. This will include labor to correct bugs in the software and future development of features to be added to this software.

**B.10. Timeline and Milestones**

**Milestone 1: Sprint 1 - Data Gathering and Analysis**

Start Date: April 1, 2023

End Date: April 21, 2023

Duration: 3 weeks

Dependencies: None

Resources Assigned: 2 full-time developers, 1 part-time developer

**Milestone 2: Sprint 2 - Development of Machine Learning Algorithms and GUI**

Start Date: April 24, 2023

End Date: May 12, 2023

Duration: 3 weeks

Dependencies: Milestone 1

Resources Assigned: 3 full-time developers, 2 part time developers

**Milestone 3: Sprint 3 - User Evaluation and Testing**

Start Date: May 12, 2023

End Date: June 2, 2023

Duration: 3 weeks

Dependencies: Milestone 2

Resources Assigned: 2 full-time developers, 1 part-time developer

**Milestone 4: Sprint 4 - Stakeholder Testing**

Start Date: June 2, 2023

End Date: June 23, 2023

Duration: 3 weeks

Dependencies: Milestone 3

Resources Assigned: 2 full-time developers, 1 part-time developer

**Milestone 5: Sprint 5 - Documentation and Bug Fixes**

Start Date: June 23, 2023

End Date: July 14, 2023

Duration: 3 weeks

Dependencies: Milestone 4

Resources Assigned: 3 full-time developers, 2 part-time developers

**Milestone 6: Final Review and Deployment**

Start Date: July 14, 2023

End Date: July 31, 2023

Duration: 2 weeks

Dependencies: Milestone 5

Resources Assigned: 4 full-time developers, 2 part-time developers

**C. Application Files**

These files are included with the submission in the attached zip folder. All files and appropriate data sets are included. Installation instructions and instructions for use are listed below in section D.8.

**D. 1. Project Purpose**

The purpose of this project is to develop a hurricane prediction system that can accurately forecast the path of Tropical Storms using the , allowing for early warning and preparation measures to be taken. The system will utilize machine learning algorithms trained on historical hurricane data taken from the NOAA to provide accurate and timely predictions. By helping to mitigate the potential damage and loss of life caused by hurricanes, this project aims to increase the safety and resilience of communities in hurricane-prone areas in an attempt to decrease the number casualties caused by natural disasters and limit the amount of damage and cost for disaster recovery.

**D.2. Datasets**

The data used for this project was taken from the NOAA website (NOAA. (2022, August 4)). This data spans back as far as 1851 but due to the high probability of unreliable data from this time era, the data has been cleaned and only used data going back to 1970 spanning to today. this data includes entries of every named storm with data being tracked in 6-hour intervals. Each entry contains data regarding the wind speed, pressure, as well as latitude and longitude of the storm at that time. This data was taken and cleaned using the panda’s library to ensure that it would be useable in this machine learning model. To ensure the integrity of this data, it has been saved and backed up in a CSV file. 80% of the data was used for training and the other 20% was used for testing and validation.

**D.3. Product Code**

The machine learning model used for this project is a “decision tree regressor” model. The decision tree regressor works by recursively splitting the data into subsets based on the input features, so that each subset is more homogeneous with respect to the target variable. The model uses a metric called mean squared error (MSE) to evaluate the quality of each split and chooses the feature that minimizes the MSE as the next decision node. The process continues until a stopping criterion is met, such as a maximum tree depth or a minimum number of samples per leaf node.

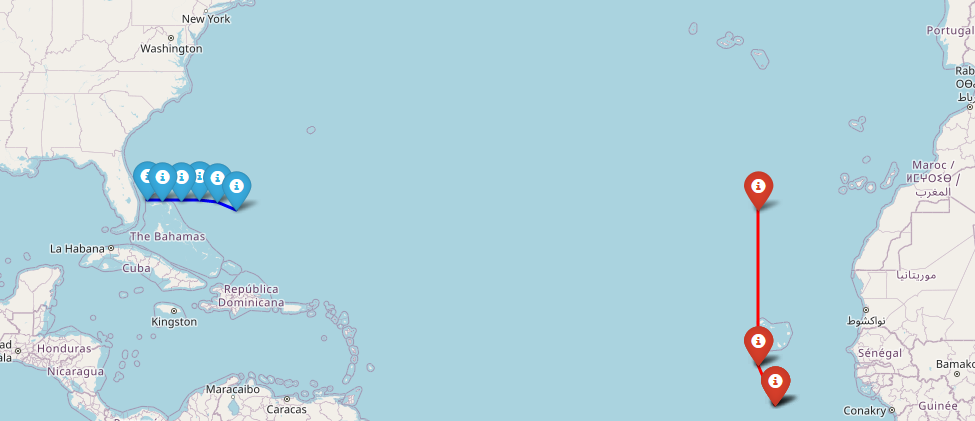
During the training phase, the model learns the optimal decision tree by fitting the data to the target variable. During the testing phase, the model applies the learned decision tree to new data to make predictions.

Overall, decision tree regressors are simple and interpretable models that can handle non-linear relationships between the input features and the target variable which is why I chose this model for this project. That being said, they can be prone to overfitting and may not generalize well to new data.

**D.4. Hypothesis Verification**

My hypothesis was that I would be able to use decision tree regression models to predict the paths of storms that may affect the United States. This turned out to be more daunting and complicated of a task than I previously imagined it to be. I hadn’t taken into consideration the amount of data it would actually take to predict the path and intensity of actual storms. However, this task was still able to be accomplished in some way. Using the latitude and longitude values in the historical data I was definitely able to predict paths of storms that could happen but by no means is it entirely accurate. This is because the conditions and weather patterns require a fundamental understanding of advanced mathematics like chaos theory and even then, the predictions would still be unreliable due to constant fluctuations in weather patterns that are not predictable. That being said, this model predicts the path of storms using the latitudes and longitudes of historical storms and makes a very unreliable prediction of the pressure and the maximum wind speeds of each predicted storm. The low reliability of these predictions are also due to the model using a vast amount of data. This causes the predictions to skew and be largely biased toward the data and not on the storm it is predicting. Blow are visuals demonstrating the prediction models output as well as the storm data used through one iteration of the program. The blue markers indicate the historical data of the real storm and the red markers indicate the path of the predicted storm.

**Actual Paths of the Random Selected Storms Used for Predictions**



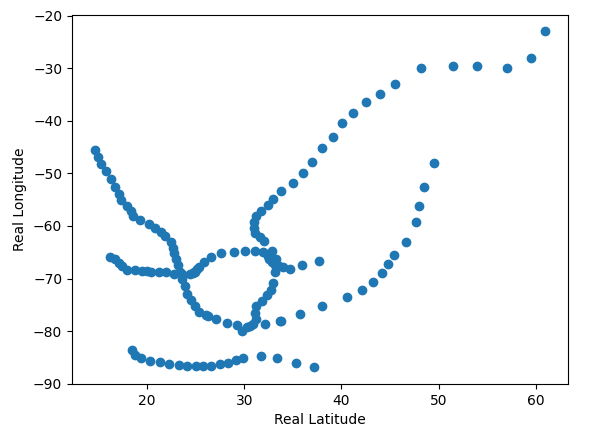
I do believe that this should still qualify this project for acceptance because, while it does not predict how I originally intended, it does in fact make predictions. This model has a lot of potential for future optimization and improvements. This program has the potential to make better and far more accurate predictions, though a lot more time and effort will be required as well as an advanced understanding of weather patterns to get it to a point that it has the capability to make somewhat accurate predictions based on real time data, not just historical data. However, this program is developed with the capability to take data from a current storm and make predictions. Overall, I believe that my hypothesis was correct. The capability to predict storms is entirely possible however, the data cannot be entirely reliable to give definitive results in a real-world scenario in its current state.

**D.5. Visuals and Data Exploration**

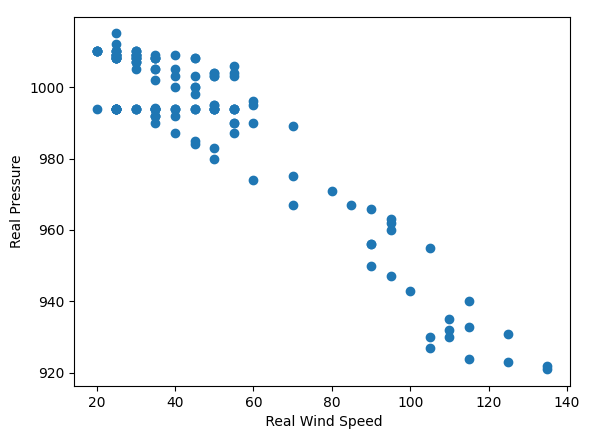
This application takes in data pertaining to historical hurricanes and uses that data for training and prediction to output a list of coordinates, wind speeds, and pressures into a CSV file that are then referenced and plotted on a map the data within this CSV file. An example of the data these CSV files contain through one iteration of the program is shown below. For clarity, the ID column represents the state of one storm over time. Each row represents the same storm at different times throughout the storm’s life.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | lat | lon | windSpd | pressure |
| 0 | 21.2 | -64.5 | 35 | 1009 |
| 1 | 22.1 | -64.5 | 35 | 1009 |
| 2 | 22.9 | -64.5 | 35 | 1009 |
| 3 | 22.9 | -60.7 | 50 | 1009 |
| 4 | 22.9 | -61.4 | 50 | 1009 |
| 5 | 22.9 | -61.4 | 50 | 1009 |
| 6 | 22.9 | -61.7 | 50 | 997 |
| 7 | 22.9 | -61.7 | 50 | 997 |
| 8 | 22.9 | -62.2 | 50 | 996 |
| 9 | 22.9 | -62.2 | 50 | 996 |

Additionally, below are the results through one iteration of the prediction model shown on a scatter plot. This diagram displays latitude vs longitude of the historical data

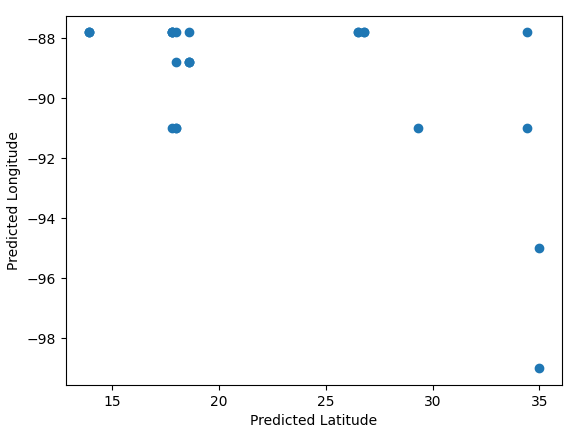


By looking at this scatter plot, it is clear that there is mostly a positive correlation between the real storm latitude and longitude values. Using this data, it is possible to infer the shape this storm’s path creates. Below is another scatter plot of the real data representing the relationship between the wind speed and barometric pressure.

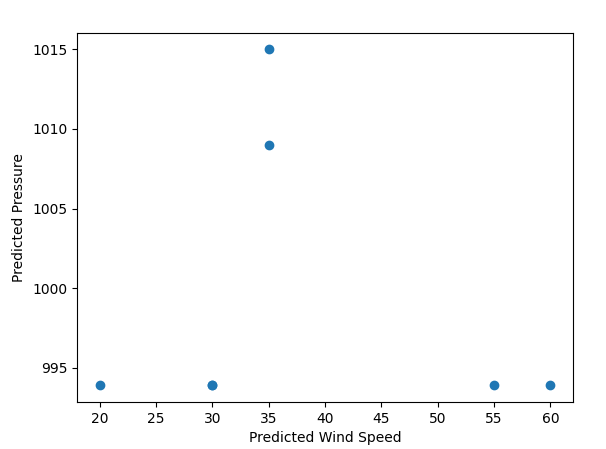


As shown above, there is a negative correlation between the wind speed and pressure. This is reliable data because the wind speed and pressure tend to decrease through the life of each storm since they eventually weaken and dissipate. Let it be known that the real-world data being used here is only half of the entire storms data set. This is because I wanted to train this model on entire storms and then use that trained model to predict a storm using only half the referenced storms data. This is why you only see the wind speeds and pressures subsiding. If it were the entire storms data set you would see additional data showing the wind speed and pressure rising to its peak as well.

Now, we move on to the predicted values and view the correlation between them as we did with the real storm data. The below scatter plots show the predicted storms latitude vs longitude as well as wind speed vs pressure.

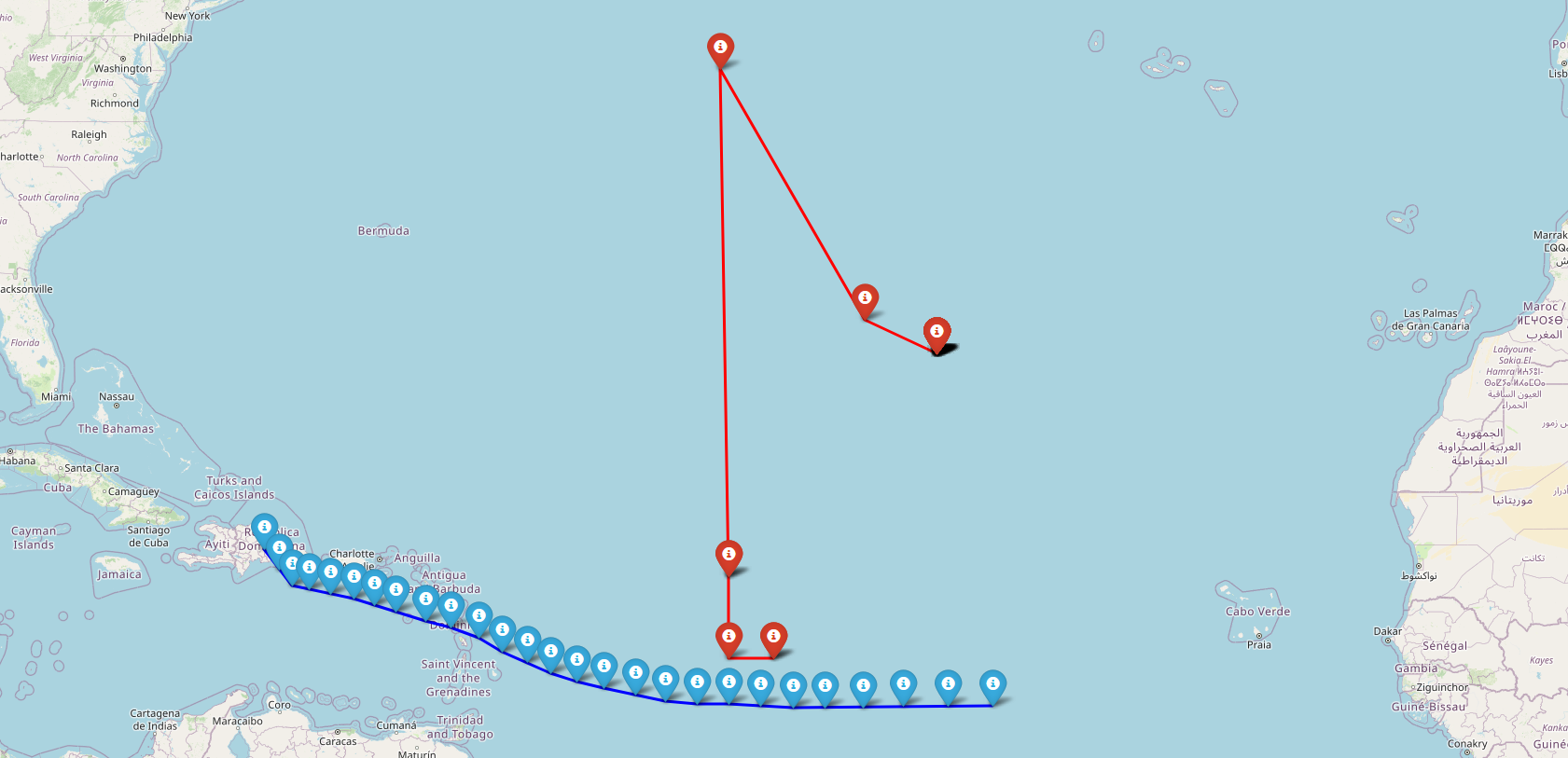


This scatter plot of the predicted storm shows a slight negative correlation between these values. As stated before, it is evident that this prediction is not entirely reliable while it is accurate (We will discuss the accuracy of this prediction in section D.6.). The reason this data look as skewed as it does is because it is heavily biased toward the massive amount of data it was trained on. Below is the correlation between the wind speed and pressure of the predicted storm.

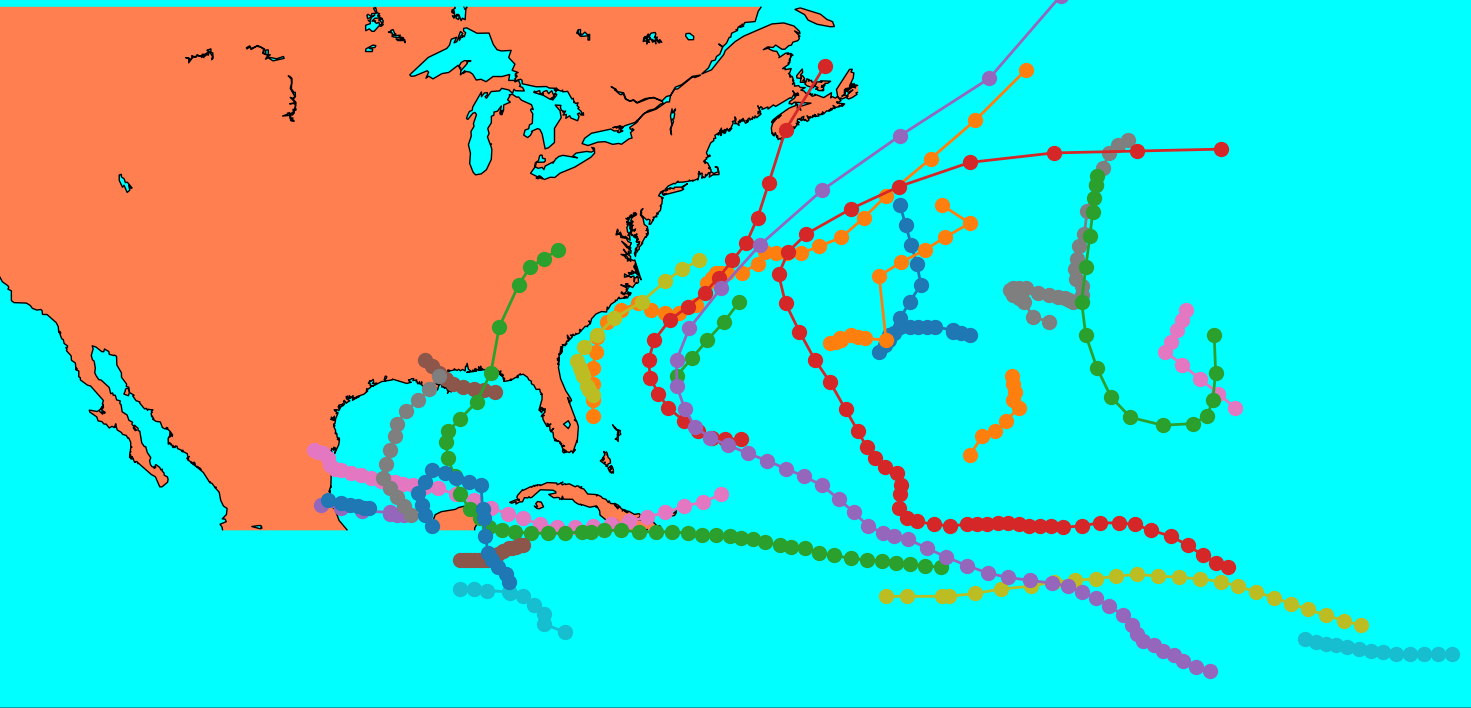


The correlation between wind speed and pressure for this prediction specifically is relevant and accurate. It essentially shows that through the life of the storm the wind speed and pressure start low, then peak, before finally subsiding again which is exactly what you find in real world scenarios.

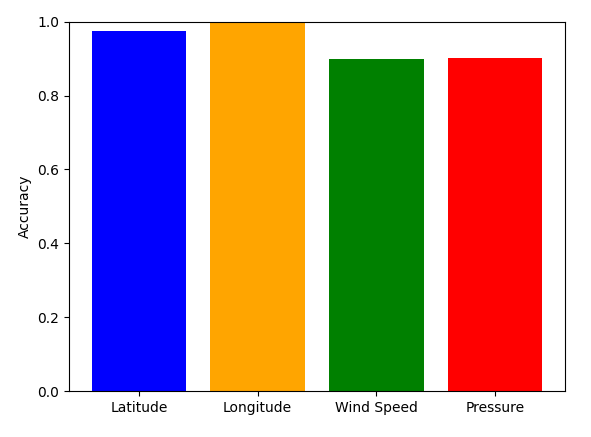
The map shown below is displaying the paths of these two storms. The markers in blue indicate the storm used for predictions and the red markers indicate the path of the predicted storm. It should also be noted that each marker on this prediction map has the ability to display the wind speed and pressure simply by clicking on any marker that is displayed. As you will see, the path of the predicted storm tends to be biased toward the trained data.



As a reference point I believe that it is important to view that the predictions are based off of. The model has the capability to display the maps of all historical hurricanes in a given year by selecting the year from a dropdown menu. One years’ worth of storm data, the year 1976 in this case, is displayed below. I believe that this exemplifies my inference that the predictions this model uses for training causes it to be heavily biased since there are 46 years’ worth of data that trains the model.



**D.6. Accuracy**  
The accuracy of this model is **s**urprisingly good, in my opinion, even though the predictions are quite skewed, this shows promise for the model to be viable with further (and significant) modifications made to it. through this one iteration of the prediction model, the accuracy of each variable (Latitude, Longitude, Wind Speed, and Pressure) is quite high and shows around 90% to 100% accuracy. It must be said though, that this is only one iteration of the model. There are times where the accuracy of these factors are not very high since the accuracy does fluctuate through each iteration of the program. It is entirely dependent on which storm the model chooses to use as a data set which is entirely random through each iteration. The graph displaying accuracy is shown below.



**D.7. Testing**

This application designed to be user friendly and simple to use overall. Testing was a simple task once all aspects were implemented. All that needed to be done was to feed the model historical data and view the results. This was made even simpler after I implemented visual representations that displayed correlations between the real-world data and prediction data. I could use these visuals to compare and contrast the data and verify that the model was predicting accurately and reliably. I was also able to compare the data to the output in the console in the IDE.

**D.8. Installation Guide**

**Hardware required:**

6 core CPU, 3.2 GHz or more

8GB RAM

Minimum 64GB Storage

**Software Required:**

Python 3.6

Python modules/libraries including pandas, folium, sklearn, numpy, pickle, tkinter, mpl\_toolkits, matplotlib

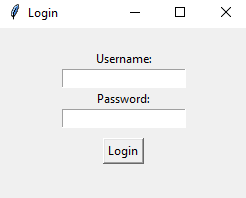
Import: webbrowser, random, csv, os, warnings

**Installation**

Place all included program files in a new folder then open Python and create a new project in the new folders directory. Finally, Install/import all required libraries.

**D.9. User Guide**

Upon running you will be met with a log in screen like the one below

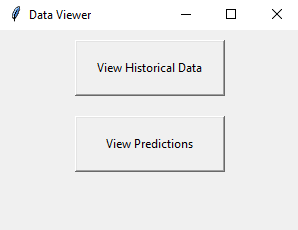


There is only one login/password and it is case sensitive though, more can be added if need be.

Login: Admin

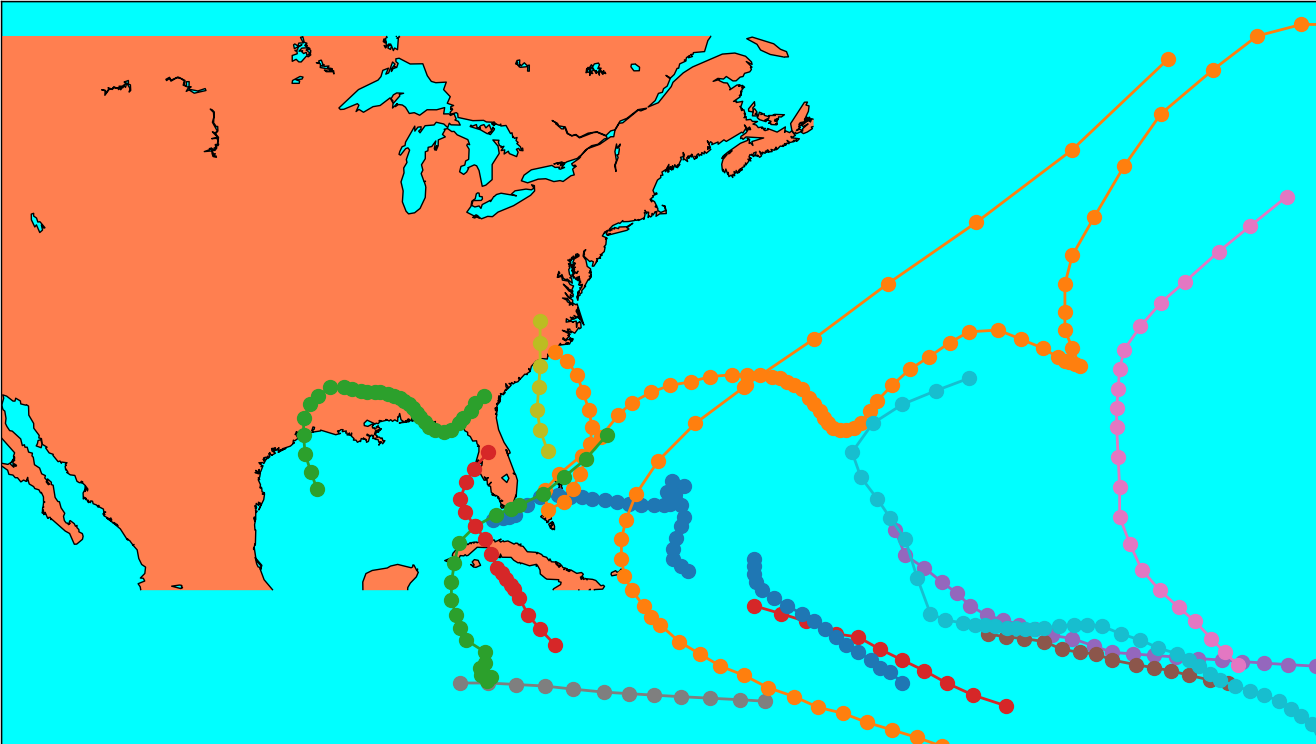
Password: P@s$w0rD

From here you will be met with the login screen below

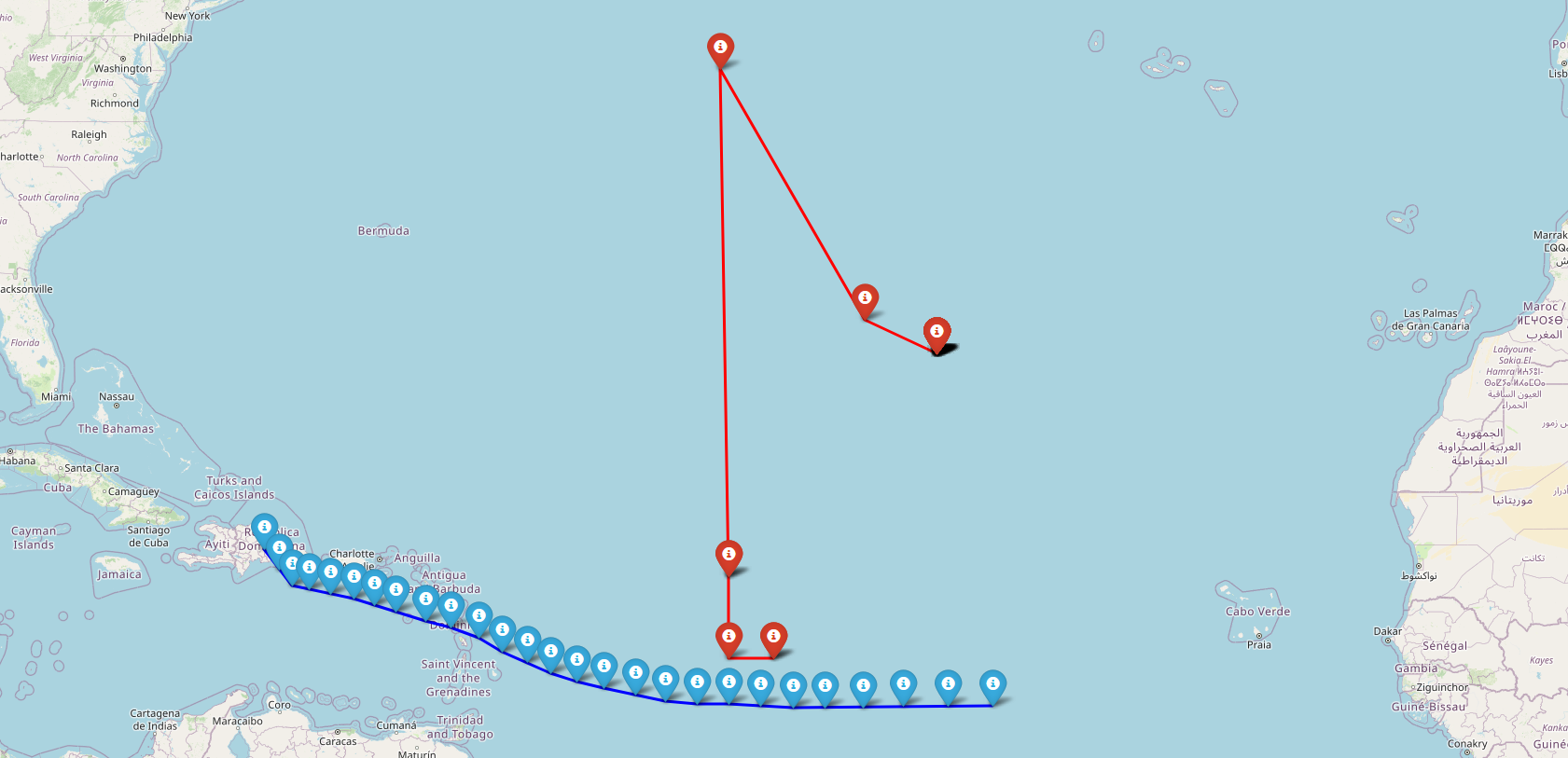


From here if you select “View Historical Data” it will generate a new screen with a dropdown menu asking for you to select a year. After selecting a year, you will select “Plot Hurricanes” and it will display all the storms for that year which will look like the figures below





Clicking on “View Predictions” will run the main program and predict storms. It will select one random storm to run the predictions against and display the predicted storm and the historical storm on the same map which is opened as an HTML file. Additionally, it will generate charts of the current iteration of the program. These charts display the correlation between real and predicted data as well as a graph showing how accurate this specific iteration of the program was. The map will look like the one below but the charts will resemble the ones shown in the review of data above. It should also be noted that each marker on this prediction map has the ability to display the wind speed and pressure simply by clicking on any marker that is displayed.



**D.10. Final Thoughts**

Overall, this project was more of an undertaking that id previously imagined. Having little to no knowledge of machine learning and regression models. It was a challenge that truly tested my abilities while teaching me a lot that I hadn’t known before. The topic I decided on was a complicated one that definitely didn’t help with the challenging aspect of this project. This was because I hadn’t considered how difficult it would be to get reliable predictions with the dataset I chose. I now understand though possible ways I could move forward with this project in the future to improve the predictions overall. I do believe that this challenge has helped me understand the concepts within this degree program, not only with programming but with writing executive summaries and evaluating my performance as a whole. My abilities were indeed tested and I do think that I’m better for it in the end.

**E. References**

1. NOAA. (2022, August 4). *Historical Hurricane Tracks*. Historical hurricane tracks. Retrieved March 4, 2023, from https://coast.noaa.gov/hurricanes