Data Analysis: Applying Machine Learning Algorithms to Drone Data

Robert Brandl
University of Nicosia Artificial Intelligence Lab

Abstract

Machine Learning continues to grow in importance, with applications stemming from artificial intelligence to drone swarms. Drone technology allows for manipulation and control through coding, where certain relationships can be utilized to enhance drone functionality. The goal of this project was to determine correlational relationships between various variables, searching for any interesting insights by developing a code-based implementation using Machine Learning. This implementation could then be generalized and further utilized in identifying other significant and/or relevant variable correlations. Finally, Webots software, which allows for drone simulations, could be used to simulate the drones along with Machine Learning algorithms through another complementary software called flockAI.

To complete this project, the Python programming language, along with its built-in Machine Learning libraries, was used. The provided drone dataset[1] was split into training and testing data, where the predictions generated by the Machine Learning algorithms were compared to the resulting testing data. The results indicate that a proportional, correlational relationship exists between two variables, wind speed and velocity in the y-direction. These results show that the implementation developed in Python can be reused and expanded to establish further relationships between relevant drone-affecting variables. In addition, these relationships can be used to improve drone predictions and allow for greater flight maneuverability, increasing drone swarm functionality.

Introduction

Drones remain a source of investment for many companies, who recognize the value and resourcefulness of drones in tasks from delivering packages to intaking meteorological data. The premise of this project centers around utilizing specific programming techniques and data analysis to determine trends and relationships between the variables which affect drone capabilities. The central data analysis technique crucial to this project is Machine Learning,

[1] https://kilthub.cmu.edu/articles/dataset/Data_Collected_with_Package_Delivery_Quadcopter_Drone/12683453?file=26385151

which involves the use of given data to generate predictions, which can help computers learn through experience. For instance, a program can be given data, from which it splits the data into testing and training data, where the training data is run through algorithms to predict the results of the testing data. At the same time, this analysis commonly revolves around the use of independent and dependent variables. An independent variable is a variable whose value does not depend upon any other variable. On the other hand, a dependent variable changes based on some other variable, commonly the independent variable. Machine Learning algorithms seek to predict data based on the relationships between different variables, oftentimes the dependent and independent variables.

For this project, all the data utilized came from a dataset of drone data. The primary source of data was the flights.csv file, which contains a variety of information on different drone flights condensed into one spreadsheet. The information collected includes qualitative data, such as routes and dates, and extensive quantitative data, including wind speed, wind angle, battery voltage, linear acceleration, and velocity, among others. For this project, only quantitative data was used to ensure accurate and unbiased results were produced. To analyze and generate predictions with this data, the coding language Python was used. This language contains various libraries with functions that make the application of Machine Learning more simplified and easier to understand.

To complete this project, the main goal focused on producing code that establishes relationships between the variables that affect drone swarms. This included applying certain algorithms, called regression algorithms, which use training data to generate predictions to be compared to the actual results (test data). My initial analysis determined relationships through trial and error, simply using the technique of linear regression (to be explained later). However, a more technical, analytical approach was used to determine variable correlations, and an array of four regression algorithms was used to create predictions to find relationships. These algorithms were compared to each other to determine the algorithm that produced the least amount of error in its predictions. Then, a final analysis was performed and compared with the testing data, demonstrating whether or not a relationship exists.

However, this project also brings about certain challenges, common to this type of data analytics. For instance, the variables must be compared to each other to determine whether any correlation exists between them. This process relies on certain correlation algorithms within

Python, along with trial and error. In addition, my choice of regression algorithms presents the challenge of determining which algorithm would be best suited to create predictions, with the highest accuracy.

Research Background & Context

To better understand the design of my project and its evaluation, it is important to understand the algorithms and libraries used. For this project, the relevant Python libraries include NumPy, pandas, matplotlib, and sklearn. The NumPy library contains methods that allow for data manipulation, including functions for calculating the mean and median of data, used within my project. In addition, the library allows for data to be standardized, meaning the data (in array form) can be converted into proportional values between 0 and 1. The pandas library provides a way to import spreadsheet files, such as the flights.csv file, and allows data to be placed in a data storage array, called a DataFrame, for further manipulation or other uses. The matplotlib library provides for creating plots, such as creating histograms with titles and axis labels.

Most importantly, the sklearn library provides functions for creating training data, regression techniques, etc. The library contains the MinMaxScaler and SimpleImputer, allowing for data to be properly scaled and missing data to be filled in with median or average data. In addition, the library contains the four regressors used for my project, the LinearRegressor, RandomForestRegressor, GradientBoostingRegressor, and KNeighborsRegressor. This library also contains the method called train_test_split, which provides a way to separate the x and y values of two variables into training data (used with the regression functions) and testing data (which is compared to the predictions).

As previously stated, regression algorithms were used as a method for generating predictions. Regression analysis has many applications but mainly searches for relationships between variables and creates predictions. The Linear Regressor is a Machine Learning algorithm that uses continuous features and produces an output, assigning variable weights based on the y=mx+b equation. In this project, x and y are the variables being tested, m is the slope of the relationship, and b is the constant when x=0. The Random Forest Regressor makes use of decision trees that classify new objects by “voting” based on similarities between trees and the new data. The Gradient Boosting Regressor also uses decision trees, where each new tree is created based on the residual error (analogous to the “negative gradient”). The K-Neighbors

Regressor is a Machine Learning algorithm for both classification and regression. For regression, it focuses on central points, noting the distance of each point from the central points, and places each point in specific groups. Afterward, it identifies new points based on its k closest neighbors and the distance formula.

Another tool useful for this project is the simulator, Webots. Webots provides a graphical interface for visualizing natural scenes and environments, along with various forms of technology, including drones. Drones can be added to the interface, along with motion detection sensors and other sensors to simulate a real drone. In addition, the drone can be controlled using code in any language, such as Python. To make the process of integrating Machine Learning results into Webots, the Artificial Intelligence Lab at the Univesity of Nicosia has developed a technology called flockAI. This software can be added into Webots and allows for Machine Learning algorithms to be easily run. In turn, drone algorithms can be run and tested on simulations of drones to determine their effectiveness. Therefore, the goals of this project can be extrapolated and directly used to control a drone's sensor system, movements, and other characteristics.

Design & Implementation

To reach my intended goals, I began by researching the various Python libraries and Machine Learning algorithms described in the Research Background section above. From there, I decided to examine some relationships through trial and error. I created a python file and began graphing x and y variables using the properties listed in the flights.csv file, searching for noticeable trends. Through this process, I noted possible relationships between wind speed and velocity in the y-direction, as well as velocity in the x-direction and orientation in the x-direction. Then, I created two python files, drone_LinReg_WindSpeedVelocityY.py and drone_LinReg_VelocityXOrientationX.py, evaluating these two potential relationships.

In both files, I evaluated the variables using Linear Regression, a technique that I had researched and had prior knowledge about. The process of analysis within each file was similar. First, I used the pandas library to import the flights.csv file, then visualized the data using a scatterplot. Then, I used the sklearn library to split the data into training and testing data and created a variable to represent the LinearRegressor, fitting it with the x and y training data. For the first file, the x values represent wind speed and the y values represent velocity in the y-direction. For the second file, the x values represent velocity in the x-direction and the y values

represent orientation in the x-direction. Using the Regressor, I generated predictions using the testing x data. Finally, I calculated the residual values or the difference between the actual results and the predicted results. These values were graphed to determine the overall trends, and the error between the predicted and actual values was found. After completing these initial files to better understand the Python libraries, I began to plan for a more objective, unbiased approach that did not rely on trial and error.

For the completed, final project, I focused specifically on the impact of wind speed on other properties of the drone. First, the file droneDataRegressionAnalysis.py, imports the data from the flights.csv file and seeks out correlations between wind speed and all other variables. Next, I chose a variable with a high correlation with wind speed, velocity in the y-direction, to continue my regression analysis, where wind speed represents the x variable and velocity in the y-direction represents the y variable. Then, I split the data into training and testing data and created a function to determine the mean absolute error (MAE), the absolute difference between the actual y values and the predicted y values. Using that function, I used the median value of the y values, calculated using the NumPy library, to generate a baseline value for the MAE.

The next stage of my implementation involved preprocessing the drone data. While not entirely necessary at this stage, this step will provide full accuracy and prove useful for potential future research. Using the sklearn library, any missing values were replaced with median values for the training and testing x values (based on the SimpleImputer). From there, a MinMaxScaler was used from the sklearn library to standardize the data. This converts all training and testing x values into proportional values in the range between 0 and 1. Finally, the training and testing y values were converted into vector values using the NumPy library (between -1 and 1), completing all necessary processing.

After completing the preprocessing, I began to utilize the various regression algorithms. First, I created a function that takes a model, trains it, and evaluates it based on the testing data. This function fits the model to the x and y training data, generates predictions from the testing x data, and calculates the MAE using these predictions and the testing y data. Then, each regression algorithm was initialized and inputted into the function, returning the MAE value. Each MAE was printed out and placed into a pandas library array (DataFrame). Next, I plotted each of the MAEs in a histogram to compare and determine which algorithm was the most effective in generating predictions. After completing this process, I observed that the Random

Forest Regression algorithm produced the lowest MAE, and used that algorithm to create a new set of predictions. Finally, I computed the residual errors for all data points and plotted them using the matplotlib library to create a histogram, allowing for a final evaluation of the model's accuracy.

All in all, the design and implementation of my project rely on Python libraries and functions to achieve unbiased and accurate results. Each step of the design process allows for simplifications to complete commands by using premade functions within Python. While choosing an initial variable to work with is often subjective, the relationships between variables can be generated using mathematical correlations. In addition, the use of libraries like matplotlib allows for easy visualization through histograms and scatterplots.

Evaluation

For the first code file, drone_LinReg_VelocityXOrientationX.py, I performed trial and error, generating various scatterplots to visualize the plots of different variables. I determined that a potential relationship was present between the velocity in the x-direction and orientation in the x-direction, displayed in Figure 1. To determine whether the correlation truly exists and create accurate predictions, I utilized linear regression and training/testing data to create a set of predictions. After developing and visualizing the residual data, I determined that a relationship does exist between these two variables. I determined that the mean squared error, an accurate measure of the error between the predicted and actual results, was around 0.05 every run of the algorithm. This low error indicates that the linear regression performed on the training data resulted in accurate predictions. In addition, Figure 2 shows the histogram created to visualize the residual errors, demonstrating how most of the residual errors (for the various data points) are centered around 0, many of which are exactly 0. Therefore, the algorithm created accurate predictions, establishing the relationship that an increase in velocity in the x-direction results in an increase in the orientation in the x-direction.

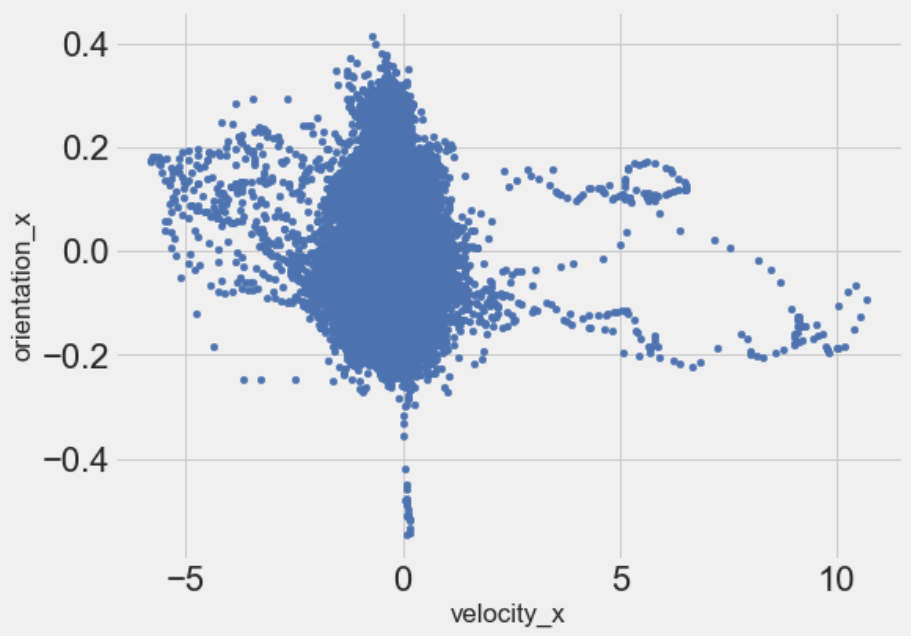

Figure 1: Drone Velocity in X-direction vs. Drone Orientation in X-direction

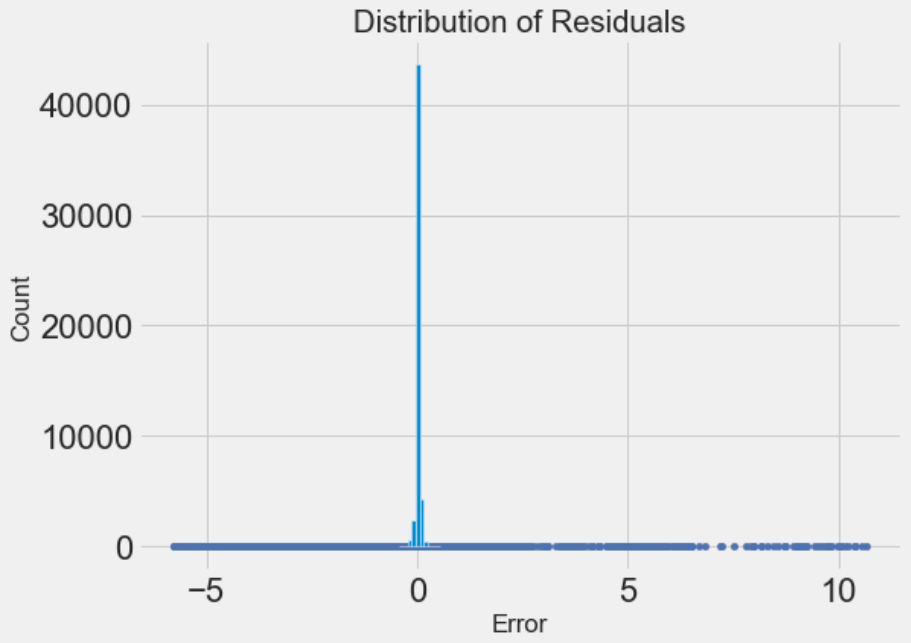

Figure 2: Residual Error from Linear Regression on Velocity in X-direction vs. Orientation in X-direction

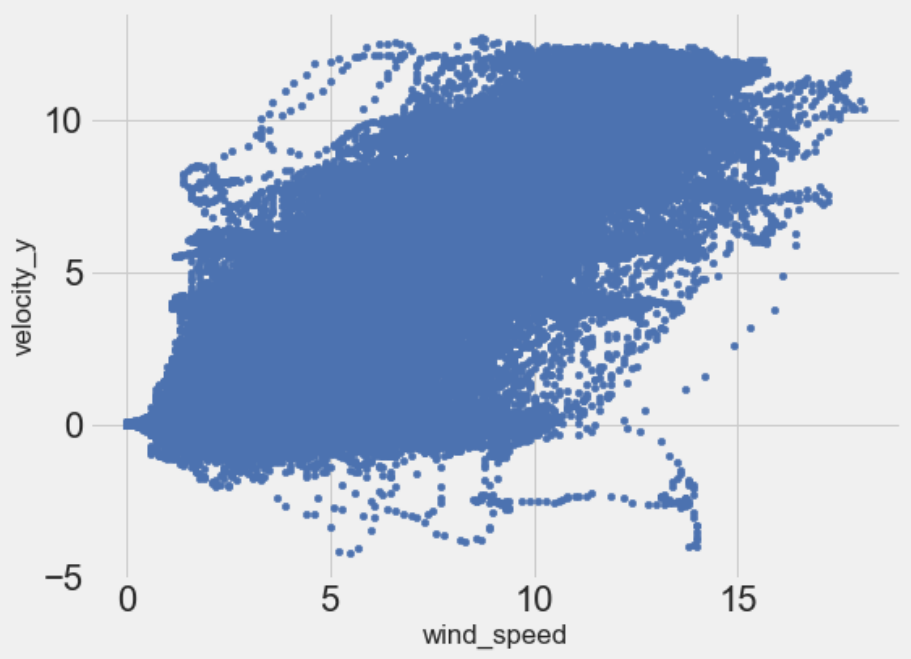

Figure 3: Drone Wind Speed vs. Drone Velocity in Y-direction

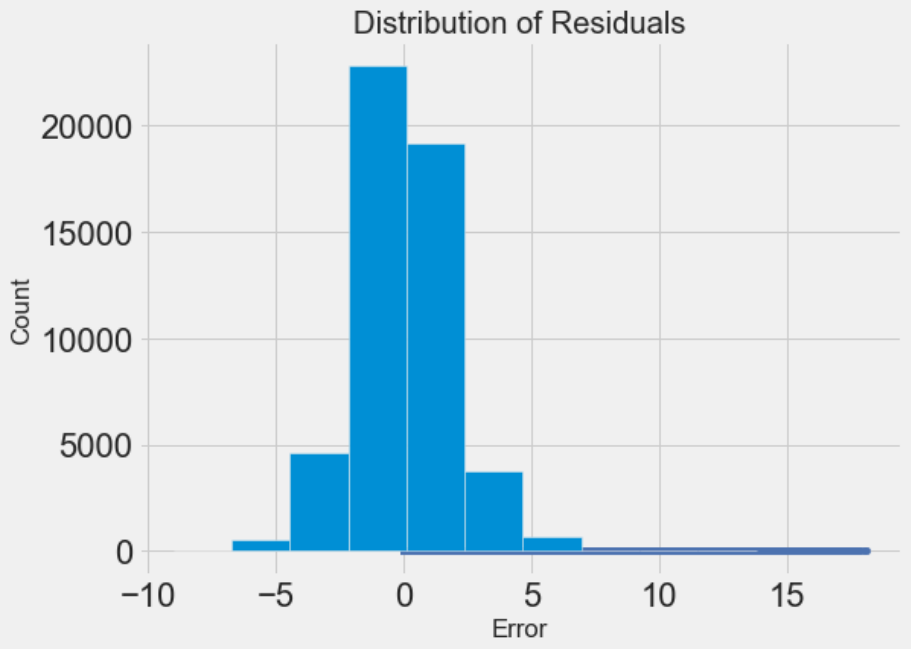

Figure 4: Residual Error from Linear Regression on Wind Speed vs. Velocity in Y-direction

Similarly, the second code file, drone_LinReg_WindSpeedVelocityY.py, was created with the same methodology, with different and less accurate results. For this file, I visualized additional scatterplots and determined to find a relationship between wind speed and velocity in the y-direction, shown as a scatterplot in Figure 3. After completing the linear regression, the mean squared error between predicted and actual results was approximately 1.8 each run. This indicates a lower accuracy than the first attempted linear regression. The relationship between these two variables is not perfectly linear, as corroborated by the histogram of residuals. Figure 4 displays a histogram that shows much of the residual errors are centered around 0, but a large number of outlier values are present. Overall, this method of analysis yielded less accurate results, prompting further analysis through a more quantitative method, seen in my final coding process.

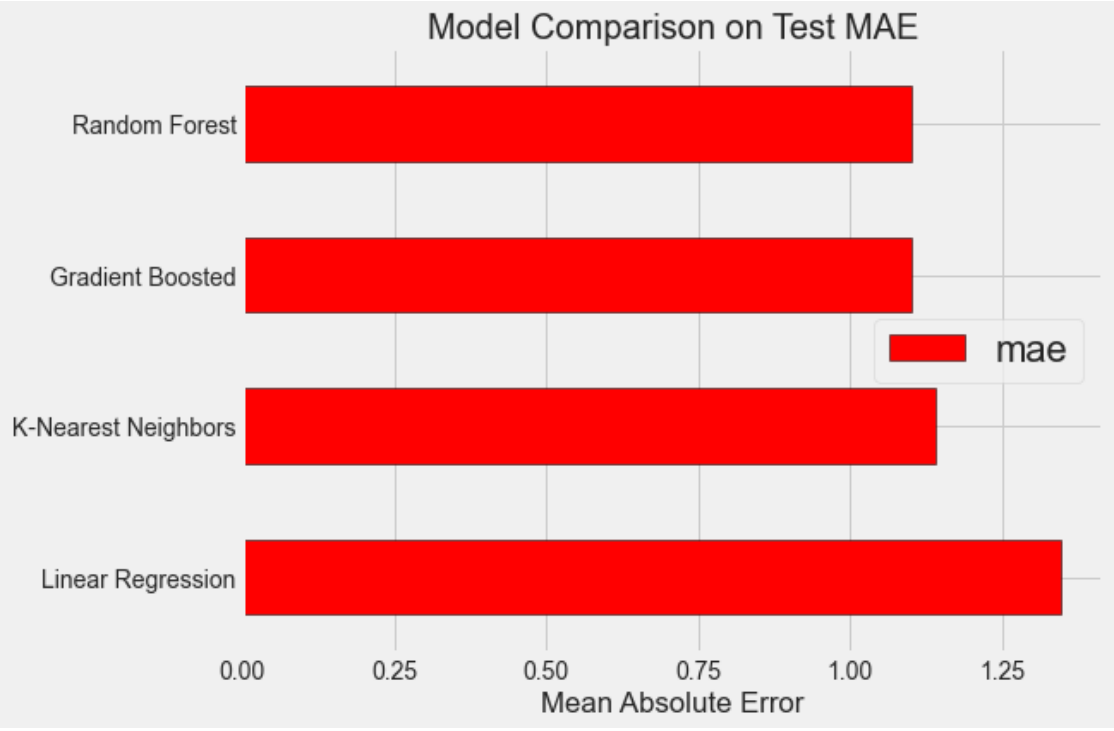

Figure 5: Mean Absolute Error for Different Regression Algorithms

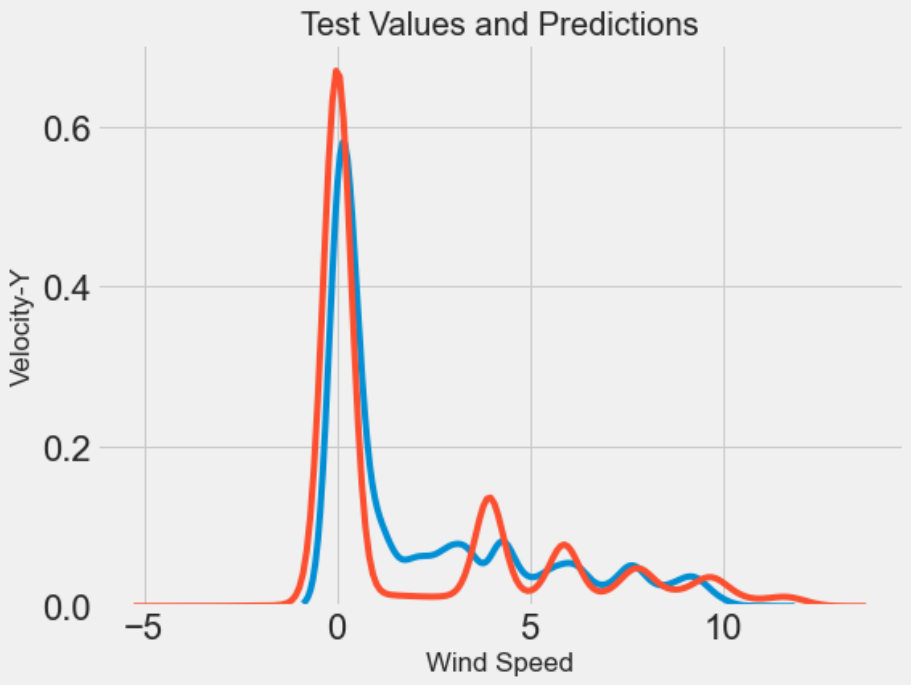

Figure 6: Comparison of Actual Drone Data (Orange) vs. Predictions (Blue)

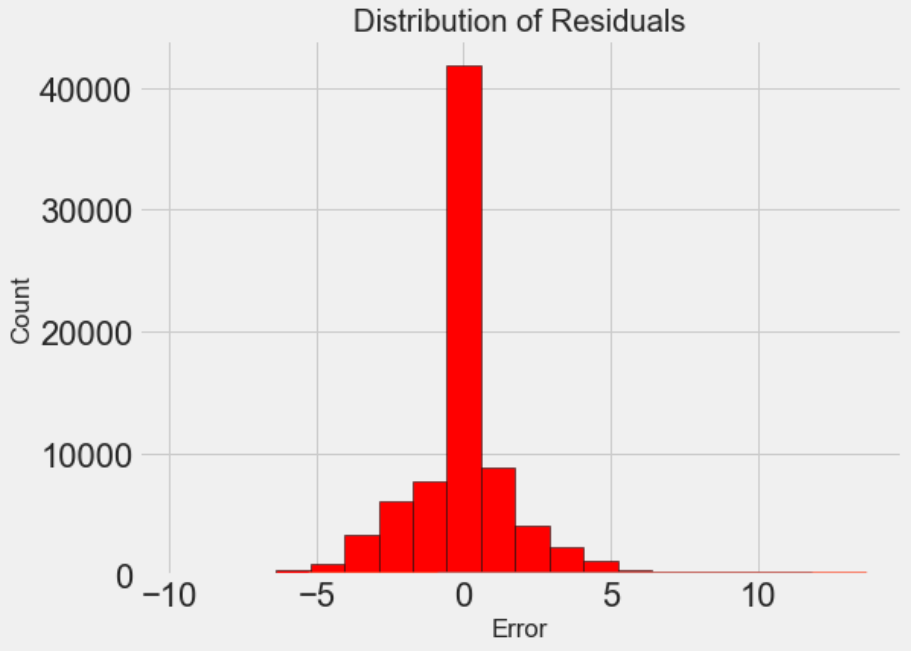

Figure 7: Residual Errors for Random Forest Regression on Wind Speed vs. Velocity in Y-direction

To continue with the variables analyzed above, I began a final Machine Learning analysis in a file called droneDataRegressionAnalysis.py. First, I used the wind speed variable as my starting variable, and with the pandas library, determined the correlations between wind speed and all the other variables. The results show that wind speed has a 0.84 correlation between wind speed and velocity in the y-direction. Therefore, some form of correlational relationship must exist between the two variables. Next, I determined the MAE between the median y-value and the predicted y-values in the test set of data, which was 2.295. This represents a relatively high MAE, and the goal of the regression algorithms was to create more accurate predictions with a lower MAE.

After the preprocessing of the data was completed, I fitted and evaluated all four regression models to the training data, creating predictions and determining the MAE for each. The MAE results for each were as follows: linear regression was 1.3488; random forest regression was 1.1015; gradient boosted regression was 1.1017; k neighbors regression was 1.1421. These results were then graphed using a histogram, seen in Figure 5, which shows that the most accurate regression algorithm, based on MAE, was the random forest regression algorithm. This algorithm slightly outperformed the gradient boosted regression algorithm by 0.0002, creating the most accurate predictions.

After completing this analysis, I used the predictions from the RandomForestRegressor to create a series of residual errors. However, I first graphed the actual results and the predictions using a curved graph, seen in Figure 6, where the actual results are plotted in orange and the predicted results are plotted in blue. This figure shows that the predicted results closely fit the actual results in various portions of the graph, although there are a few notable discrepancies. Then, I plotted these residual values onto a histogram, displayed in Figure 7, where the vast majority of residuals fell around 0, where the actual and predicted values were exact or almost exact. Overall, these results demonstrate that this Machine Learning algorithm is fairly accurate. Another appropriate demonstration of the accuracy of the results is the difference between the baseline MAE and random forest regression MAE: 1.1935. This represents a decrease of just over 52%, indicating the predictions were more accurate than just using the baseline value or the median test y-value. All in all, my objective calculations establish that a general relationship exists between wind speed and velocity in the y-direction, where a greater wind speed often indicates a greater velocity in the y-direction. However, it is important to note that this

correlational relationship proves helpful in predicting future drone flights, but does not directly prove any form of causation, which was not necessary for this project.

Conclusion

After completing my data analysis using Machine Learning, it is clear that relationships exist between the variables that affect the flight of a drone. External factors such as wind speed can affect internal factors like velocity. My research determined that a correlational relationship exists between wind speed and velocity in the y-direction. After preliminary research and trial and error, this relationship was quantified through regression and data analysis. The implementation of this project can be extrapolated and utilized to find further relationships between variables affecting drones. These results can be put to use in drone detection and drone programming, allowing the drone to generate preliminary results depending on the wind speed (create corresponding velocity-y values). Further research can utilize the Webots and flockAI software to replicate the drone's conditions and make proper use of Machine Learning algorithms. This will allow the drone to predict and anticipate certain capabilities, such as when it needs to slow down and what direction it needs to move.

However, further research can focus on greater accuracy through parameterization, where the regression algorithm is further adjusted and manipulated to be more precise in generating predictions. In addition, more algorithms can be tested to determine further ways to increase prediction accuracy. All in all, the use of Machine Learning allows for correlations to be easily identified between different variables, and the creation of reliable predictions that has wide-reaching applications, including drone functionality.