North American Weather Prediction Readme

1. Problem Statement
    a. The goal of the NA weather prediction project was to take a dataset of 5 years of hourly recorded weather data from 30 north american cities and see if a machine learning model could predict the type of weather that would be present given similar weather data. Having a solution to this problem would let almost anyone predict the weather ahead of time and incorporate that data into their personal or business decisions.
2. Acceptance Criteria
    a. Measurements of success for this project were an accuracy of above 80%, and a balance of precision and recall scores. The solution should also be containerized and easily deployable in a cloud environment. Overall the project was a success because the models achieved accuracies in the low 90s.
3. Technical Details
    a. Overview
        i. For this project it was decided to use two random forest classifier models to achieve the predictions wanted. First the input features get fed into a binary classification model that predicts whether or not the weather will be sunny. If the result is sunny it returns that prediction. If the weather is predicted to not be sunny, then the same input features get fed into a multi classification model that predicts all other types of weather. The final prediction is then returned.
    b. Feature Engineering
        i. The engineered features are the daily, monthly, and yearly rolling averages of the humidity, temperature, and

pressure features. These engineered features provided the needed boost to get to an acceptable level of accuracy. All averages were engineered by computing the rolling average of the respective time length over the entire dataset.

c. Feature Description

i. **Humidity:** Measure of humidity in % relative humidity.

ii. **Temperature:** Measure of temperature in Kelvin.

iii. **Pressure:** Measure of pressure in hectoPascals.

iv. **Wind Direction:** Measure of wind direction in degrees.

v. **Wind Speed:** Measure of wind speed in miles per hour.

vi. **Latitude:** Measure of position in degrees.

vii. **Longitude:** Measure of position in degrees.

viii. **Month:** The month the measurements were taken in.

ix. **Day:** The day of the month that the measurements were taken.

x. **Hour:** The hour of the day that the measurements were taken in.

xi. **Humidity Daily Average:** Rolling daily average of humidity.

xii. **Humidity Monthly Average:** Rolling monthly average of humidity.

xiii. **Humidity Yearly Average:** Rolling yearly average of humidity.

xiv. **Temperature Daily Average:** Rolling daily average of temperature.

xv. **Temperature Monthly Average:** Rolling monthly average of temperature.

xvi. **Temperature Yearly Average:** Rolling yearly average of temperature.

xvii. **Pressure Daily Average:** Rolling daily average of pressure.

xviii. **Pressure Monthly Average:** Rolling monthly average of pressure.

xix. **Pressure Yearly Average:** Rolling yearly average of pressure.

d. Model Selection

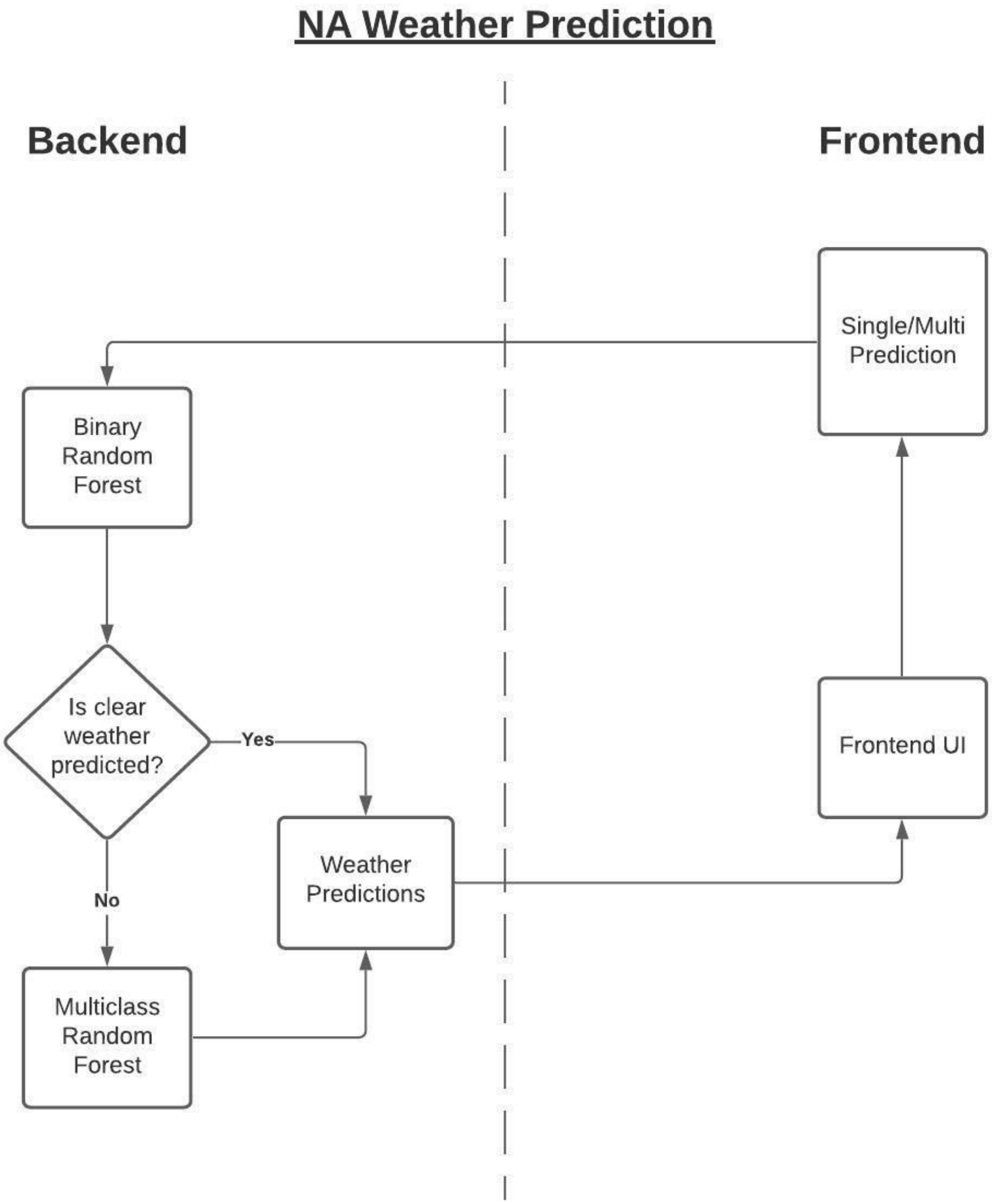
i. At first, due to the sheer amount of data available, about 1.3 million records, a neural network was implemented but the training time was very long and the accuracy, even with some feature engineering, was not high enough.

ii. Next, a random forest model was tried. This type of model had a much faster training time and had a much higher accuracy that was acceptable for the project.

iii. Hyper-parameters:

1. n_estimators: tried with values of 10, 100, 1000. A value of 1000 for the increased accuracy while still having a sustainable training time.
2. n_jobs: determines how many processors are used for computation. This value was -1 so that all available processors aided in the computation.
3. Random_state: Controls both the randomness of the bootstrapping of the samples used when building trees. Value of 52 was arbitrarily chosen.
4. All other hyper-parameters have their default values.

4. Dataflow Diagram



5. How to Use
    a. Deployment Design
        i. The way the project is setup is that there are two docker containers that will run all of the code using the flask framework. The first docker container will function as a backend and be responsible for training, saving, and loading the models, as well as receiving input features and return predictions based on them. The second container will function as a frontend and will be responsible for receiving and sending all user input to the backend container as well as receiving the predictions returned from the backend container and creating some data visualizations based on the labels that the user provided.
    b. Project setup
        i. Go to https://github.com/dnorris823/NA_Weather_Prediction and obtain the project.
        ii. If you do not have Docker, install docker.
    c. How to set up containers
        i. Backend Container
            1. Navigate via the terminal to the ML backend folder and type the following commands:
            2. docker build -t mytag .
            3. docker run -d -p 80:80 --name mycon mytag
        ii. Frontend Container
            1. Navigate via the terminal to the Web App folder and type the following commands:
            2. docker build -t mytag_f .
            3. docker run -d -p 90:90 --name mycon_f mytag_f
    d. How to navigate UI
        i. Single_prediction

    1. Enter in each individual value, paying close attention to the unit of measurement needed.
    2. Click on the “Predict” button and in a few moments your prediction should appear below.

ii. Multi Prediction

    1. **NOTICE**: please limit the number of records in the files you wish to upload to under 10,000. Any more than that and it will take more than a few minutes to return your predictions.
    2. First, in the upload features section, choose a file that contains all of the records of features you wish to predict on.
    3. Click on the upload button in the same section
    4. In the upload labels section, choose a file that contains all of the records of labels you wish to compare your predicted labels with.
    5. Click on the upload button in the same section.
    6. Click on the “Make Predictions” button.
    7. Depending on the amount of records in your files and the hardware you are running on, this may take up to 10 minutes.
    8. Once the predictions appear you should be able to see a list of your predictions, your accuracy based on the labels you provided, a classification report, a confusion matrix, and a frequency chart of your predicted values.