

## **Project 4 Proposal**

### **Predictive Temperature Modeling of Weather Patterns Using NOAA Station Data**

**Group Member: Daniel Marquez, Lailah Libay, Zahra, Peiran (Diana) Cao**

#### **1. Introduction**

Our world is becoming increasingly dynamic, and with the effects of climate change, understanding weather patterns and predicting them has never been more important. Our project aims to demystify machine learning by building a predictive model using weather data from the NOAA station USW00064775(ROME GRIFFISS AIRFIELD, NY US)

#### **2. EDA (exploratory Data analysis)**

We will leverage Python's Pandas library to call the API and preprocess the data, preparing it for our machine-learning model. This will involve handling missing values, normalizing numerical data, and converting categorical data into a machine-readable format.

We will acquire data directly from the NOAA API and store it in a SQL database hosted PostgreSQL, ensuring that the full model utilizes direct connection, complying with the project requirements. The data will contain various weather patterns, including average daily wind speed, maximum temperature, precipitation, and more.

```
datatype = ["PRCP", "TAVG", "AWND", "TMIN", "TMAX"]
```

```
PRCP: Total precipitation.  
TMAX: Maximum temperature.  
TMIN: Minimum temperature.  
TAVG: Average temperature.  
AWND: Average wind speed.
```

### 3. matrixes

```
PRCP: Total precipitation. -DV  
TMAX: Maximum temperature. -IV  
TMIN: Minimum temperature. -IV  
TAVG: Average temperature. -IV  
AWND: Average wind speed. - IV
```

### 4. Initial hypothesis

Based on the weather data obtained from the National Oceanic and Atmospheric Administration (NOAA) for

Rome Griffiss Airfield, NY, US, our initial hypothesis is as follows:

"We predict that weather parameters at Rome Griffiss Airfield, specifically temperature and precipitation, have a significant correlation with wind speed. More specifically, we hypothesize that:

1. Wind speed increases with a drop in temperature, suggesting that colder weather is generally windier. This is based on the principle that temperature differences drive air movement, resulting in wind.
2. Higher wind speeds correspond to higher levels of precipitation. This may be due to the role of wind in transporting moisture from one region to another, which can result in increased precipitation under the right conditions.
3. Extreme temperature events (both maximum and minimum) may be associated with unusual wind speed patterns. This could be due to the large temperature differentials that drive these extreme events, leading to increased wind activity.

Our hypothesis will be tested by investigating the correlation coefficients between these weather parameters. We will also consider seasonal variations, which could

influence these relationships, by conducting a month-wise or season-wise analysis. We note that this hypothesis assumes a linear relationship between these parameters, which may not always hold true."

Please note that this is an initial hypothesis, and further in-depth statistical analysis is required to confirm or disprove it. For example, linear regression models or machine learning algorithms could provide a more nuanced understanding of the relationships between weather parameters.

## Napkin Drawing

The screenshot shows a Jupyter Notebook interface with the following content:

- Left sidebar:** A file explorer showing a project structure with files like `__init__.py`, `Resources`, `api.py`, `api_utils.py`, `data_loader.py`, `data_processor.py`, `data_validator.py`, `data_writer.py`, `README.md`, `UTOR-WRT-DATA-PT-11-2022-U-LOLD`, and `Business Intelligence Data Analyst - Career Portal.pdf`.
- Main area:** Contains Python code for fetching weather data from an API. The code includes a loop for each month from 2010 to 2019, fetching data for parameters: `PRCP`, `TAVG`, `TMIN`, `TMAX`, and `WVAP`. The data is stored in a list of dictionaries.
- Handwritten notes:**
  - At the top: "FOR api for each month from 2010 - 2019".
  - Below the code: "insert into a table -> PostgreSQL".
  - At the bottom left: "df = pd.read\_sql()".
  - At the bottom right: A table with columns: `date`, `PRCP`, `TAVG`, `TMIN`, `TMAX`, and `WVAP`. Below the columns are the abbreviations: `D.V.`, `I.V.`, `I.V.`, `I.V.`, `I.V.`, and `I.V.`.

