Rylan Mattocks

WGU c964 Computer Science capstone

Temperature prediction

Contents

[Section A: 3](#_Toc102571827)

[1. Letter of Transmittal 3](#_Toc102571828)

[2. Project Recommendation 4](#_Toc102571829)

[a. Problem Summary 4](#_Toc102571830)

[b. Application Benefits 4](#_Toc102571831)

[c. Application Description 4](#_Toc102571832)

[d. Data Description 4](#_Toc102571833)

[e. Objective and hypotheses 5](#_Toc102571834)

[f. Methodology 5](#_Toc102571835)

[g. Funding Requirements 5](#_Toc102571836)

[h. Stakeholders impact 6](#_Toc102571837)

[i. Data Precautions 6](#_Toc102571838)

[j. Developers Expertise 6](#_Toc102571839)

[Section B: 7](#_Toc102571840)

[1. Problem Statement 7](#_Toc102571841)

[2. Existing System Analysis 7](#_Toc102571842)

[3. Data 7](#_Toc102571843)

[4. Project Methodology 8](#_Toc102571844)

[5. Project Outcomes 8](#_Toc102571845)

[6. Implementation Plan 8](#_Toc102571846)

[7. Evaluation Plan 9](#_Toc102571847)

[8. Resources and Costs 9](#_Toc102571848)

[a. Programming Environment 9](#_Toc102571849)

[b. Environment Costs 9](#_Toc102571850)

[c. Human Resource Requirements 9](#_Toc102571851)

[9. Timeline and Milestones 10](#_Toc102571852)

[Section C: 11](#_Toc102571853)

[Section D: 12](#_Toc102571854)

[1. Project Purpose 12](#_Toc102571855)

[2. Datasets 12](#_Toc102571856)

[3. Data Product Code 13](#_Toc102571857)

[4. Hypothesis Verification 15](#_Toc102571858)

[5. Effective Visualizations and Reporting 15](#_Toc102571859)

[6. Accuracy Analysis 15](#_Toc102571860)

[7. Application Testing 16](#_Toc102571861)

[8. Applications Files 16](#_Toc102571862)

[9. User’s Guide 16](#_Toc102571863)

[10. Summation of Learning Experience 17](#_Toc102571864)

[Section E: 17](#_Toc102571865)

[1. Sources 17](#_Toc102571866)

# 

# Section A:

## Letter of Transmittal

Rylan Mattocks

Project Manager

Indianapolis Weather Association

999 S Meridian St

Indianapolis IN, 46225

Dear Mr. Collins.

The Indianapolis Weather Association is the go-to business for temperature readings in Indianapolis. However, a major component that IWA lacks is temperature predictions. With many people living in Indianapolis and others planning on visiting, they are desperate to know what the weather will be like in the future. Currently, IWA has access to previous year’s weather data, and with this data, we would be able to develop an application to predict future temperatures. This will improve IWA’s customer relations significantly, and increase the customer base.

The proposed temperature prediction application will include a dashboard to allow users to view past weather data, as well as get a weather prediction for any specific day. This will benefit IWA by improving customer relations. The consumers will be able to see future predictions of temperature data allowing them to make adjustments to plans accordingly. This application will also increase the customer base for IWA. Since we currently do not have any way of predicting temperature data in the future, more consumers will begin using the company’s services.

The requested funding to develop the temperature prediction application is around $50,000. The majority of funding will be used for development and labor. My expertise is python, as well as, my management experience will allow me to lead a team to develop the application required to implement temperature predictions to IWA.

I am looking forward to hearing from you, and hopefully helping implement this application. Please contact me with any questions you may have.

Sincerely,

Rylan Mattocks

Project Manager

## 

## 2. Project Recommendation

### a. Problem Summary

Indianapolis Weather Association is the leading organization for weather data in Indianapolis. Every day hundreds of users access our website to view the weather for the day. Other organizations such as the National Weather Service offer weather forecasting up to seven days in the future. Without proper weather prediction, consumers will begin to look elsewhere for their weather information. The weather prediction application will aim to give users weather data that is easily accessible and offer temperature forecasting. This application aims to retain consumers as well as expand the customer base.

### b. Application Benefits

Indianapolis Weather Association currently offers daily temperature data, as well as having previous temperature data stored. With more people making plans for outdoor activities and visits to Indianapolis, our current system lacks in providing them with any information. This causes users to look towards other organizations to find the information they need.

The proposed application will be able to address these problems. This application will be able to predict temperature readings for future dates based on the previous weather data of the last 25 years. The application will also be able to allow the user to view the previous years’ data in graphical formats. This will allow the user to see the data that the prediction is being made with as well as compare the temperatures throughout the years. This application will help retain customers by giving them the information they are seeking elsewhere.

### c. Application Description

The application will be developed using Python and its libraries. The program will be hosted on Streamlit to create the web application. The application will give the option to sign up for an account or log in to an existing account. Upon logging in, the user will be able to access multiple tabs to view our temperature data. The tabs will consist of: the average temperatures by season, the average temperature by year, the temperature each day by month, and temperature predictions. The following tabs will consist of a slider to view these temperatures by each year: seasonal temperatures and monthly temperatures. The temperature prediction will allow the user to select a month, day, and year, and will output the predicted temperature value for that date. This information will aid in consumer retention and increase the customer base.

### d. Data Description

The dataset that will be used for the application will be attained from kaggle.com. The dataset will contain the temperatures from the last 25 years in major cities across the world. The dataset will need to have Indianapolis extracted from it. The data contains the following information:

* Region
* Country
* State
* City
* Month
* Day
* Year
* Average Temperature

The data will be in CSV format. The data will need to be cleaned to remove any bad data for accuracy purposes. A big limitation of the dataset is it only goes back 25 years. This will be okay to use as a proof of concept, but to greatly increase prediction accuracy, the data should include much more temperature data. The product will be able to adapt to a larger data set with minimal to no changes.

### e. Objective and hypotheses

The objective of the temperature prediction application is to provide consumers with the ability to see temperature forecasts. This will benefit Indianapolis Weather Association’s consumer retention as well as increase its customer base. The application proposes that if given the previous years’ temperatures, then future temperatures can be predicted. Accurately predicting future temperatures for users will improve customer retainage because customers will no longer be using other services such as the National Weather Service.

### f. Methodology

The development methodology that will be used to create the temperature prediction application will be the Software Development Life Cycle (SDLC) methodology. This methodology will be used because the application does not need the increased flexibility of the agile methodology. The requirements for the application are simple and will not require adjustments after they are established. The main object of the application is to implement a temperature prediction. It would be easy to try implementing more components to the application that would be considered out of scope, and with SDLC the requirements will be clearly defined beforehand limiting any work done not in the scope of the project. The stages of the SDLC are as follows:

1. Planning and Requirement Analysis – This is where requirement analysis and risk identification occurs.
2. Defining Requirements – This is where the requirements are created and approved by the customer.
3. Designing the Product Architecture – This is where the best architecture is chosen for the project based on parameters such as risk assessment, product robustness, design modularity, budget and time constraints.
4. Building or Developing the Product – This is where the development starts based on the information from the previous stages. It will be written in Python.
5. Testing the Product – This is where the product is tested and fixed to meet quality standards.
6. Deployment in the Market and Maintenance – This is where the product is deployed and, based on customer feedback, enhanced. Also, maintenance will be done throughout the life of the project.

### g. Funding Requirements

The initial funding requirement will be $50,000. This will cover the application’s development cost. The cost will also cover the labor required by the development and testing teams. Additional requirements may be implemented in the requirements stage which could increase the cost of the project. The additional funding would cover the time increase if the requirements add work that is currently considered out of scope. Also, the architecture may be changed to accommodate certain aspects of the project. This could increase or decrease project costs depending on what is chosen.

### h. Stakeholders impact

The temperature prediction application will have a large impact on stakeholders. The main benefit will be for consumers. The application will give access to viewing previous temperature data, as well as the prediction of future temperatures for a given date. This will greatly increase customer satisfaction, as well as customer retainage. Customer satisfaction will increase because users will be able to plan for temperatures in the future. This will also keep customers using our application rather than seeking out competitors to view temperature forecasts.

### i. Data Precautions

The data used for the temperature prediction application does not violate any regulations such as HIPPA, FERPA, or PCI DSS. Once the application is deployed, users will be required to create an account to access the information. This will require them to create a username and password. This data will be hashed to avoid information being stolen in the case of a data breach.

### j. Developers Expertise

The developers are college graduates from Western Governors University. They have some experience with python, as well as other programming languages. They are also confident in their mathematical abilities which will be used in the creation of the linear regression model and standard deviation calculation of the data. They have a good understanding of development methodologies. They also have leadership experience in their previous job positions, which will be a valuable asset in leading the team. They have the experience required to complete the temperature prediction model in the timeframe provided.

# Section B:

## 1. Problem Statement

The temperature prediction application will benefit the Indianapolis Weather Association’s consumer retention and expansion. The application will use linear regression to predict future weather temperatures to give our consumers a temperature forecast. Our consumers will no longer need to seek out temperature forecasts from our competitors, therefore retaining our customer base. This application will also give users the ability to view previous temperature data in various formats. This, combined with the temperature prediction, will increase customer retention and provide a better customer experience which will increase customer satisfaction.

## 2. Existing System Analysis

The current system in use by Indianapolis Weather Association does not offer a forecast of any type. The lack of future temperature information is causing users to look towards competitors to fulfil their needs. The temperature prediction application will give users the temperature forecasting they are looking for, therefore increasing customer satisfaction and retention.

Indianapolis Weather Association does not offer any graphical analysis of previous temperature data. The temperature prediction application will also offer graphical views of previous temperature data. These displays will include average seasonal temperatures, average monthly temperatures, and average yearly temperatures. This will provide users with the data we use to predict future temperatures as well as the ability to observe previous temperature data in an easy to view display.

## 3. Data

The data that will be used to develop the temperature prediction application is a dataset that was found on kaggle.com. The dataset can be viewed here: <https://www.kaggle.com/datasets/sudalairajkumar/daily-temperature-of-major-cities>

The dataset includes the following information:

* Region
* Country
* State
* City
* Month
* Day
* Year
* Average Temperature

The temperature prediction application will not use all of the data in the dataset. Before using the dataset, the Indianapolis area needs to be extracted from the dataset. This is done by filtering the ‘City’ to ‘Indianapolis’. The dataset also includes unused information that is kept such as Region, Country, State, and City. This is redundant information because we are only using the temperature data for one location. The data itself is complete without any anomalies. In the future, the dataset should contain much more data to give a more accurate prediction of future weather temperatures. The application will require minimal to no changes to accommodate a larger dataset given the same format.

## 4. Project Methodology

The project will be developed using the Software Development Life Cycle (SDLC) methodology. The project should not require changes from the original requirements and does not need to implement additional components that would be viewed as out of scope; therefore, the SDLC methodology would be the best fit. The following are the phases for the SDLC methodology:

1. Planning and Requirement Analysis – The team structure will be determined during this phase, as well as, the goals and costs of the project.
2. Defining Requirements – The development team and stakeholders will meet to determine the requirements and expected performance of the application. This will include any functionalities the application should offer.
3. Designing the Product Architecture – The development team will decide on the architecture of the application during this phase. This is also where the methods for performing tasks will be defined. The team will also decide the platform the application will run on and design the user interface.
4. Building or Developing the Product – The development team will convert the previously defined features into code, making sure all components are implemented.
5. Testing the Product – The testing team will test the application, ensuring all components work as intended.
6. Deployment in the Market and Maintenance – The product will be deployed into the market for consumers to begin using. Using feedback from the consumers, the application can be improved with new features and fixed to eliminate any bugs and vulnerabilities.

## 5. Project Outcomes

The deliverables for the project will include all of the materials produced during the project’s development. The deliverables included will be cost distribution, project schedule, error documentation, and project documentation such as the software requirement specification document, the design document specification, and technical feasibility study.

The deliverables for the product will include the dataset and the temperature prediction application. The graphical displays will also be included as product deliverables.

## 6. Implementation Plan

The implementation strategy will use the Software Development Life Cycle (SDLC) methodology. The requirements and expectations will be produced by the development team along with the input from the stakeholders. The implementation will consist of four phases: First, the design and feature development; second, the prototype for the user interface; third, the development and testing; and finally deployment.

After the requirements, features, and prototype is finished, the development team will begin the start developing the application. The development team will be able to focus on completing the project without changes affecting the development process. The testing will be completed by the quality assurance team. This testing will include user acceptance testing as well. Upon completion of all development and testing, with approval from the stakeholders, the application will be deployed for use.

## 7. Evaluation Plan

The Software Development Life Cycle (SDLC) methodology will clearly state the requirements for the application before development. This will help assure all project features are implemented throughout development, and the features that are implemented are within the scope of work. Also, the requirements will make testing for missing features much easier.

The temperature prediction application will be evaluated based on the accuracy of the predictions. The accuracy will initially be less accurate because of the amount of data being used. This will increase the error calculation of the data as well. With more data, the accuracy of the prediction will be much greater. The feedback from users will provide information about the accuracy of the predictions as well.

## 8. Resources and Costs

### a. Programming Environment

The application will use Python 3.9.12 and its libraries, GitHub, and Streamlit to develop the temperature prediction application and dashboard. These are all accessible on Windows, Mac, and Linux computers and are all open-source resources, therefore, free of cost. The programming environment will not need to budget any cost.

### b. Environment Costs

The temperature prediction application will be hosted via Streamlit by Indianapolis Weather Association. This will have no cost for hosting or licensing, which will have the environment costs budget free of cost.

### c. Human Resource Requirements

The labor costs of developing the application are expected to be $50,000. This cost could increase if features or requirements are added that are considered out of scope. The breakdown of the costs are as follows:

Phase one – Requirement identification and project planning for the prediction application as well as the feasibility study. This is mainly administrative labor. $10,000

Phase two – The development of the prototype and design of the application. $12,000

Phase three – The development and testing of the final application. This includes usability testing. $20,000

Phase four – Application deployment and maintenance such as technical support. $8,000

## 9. Timeline and Milestones

The time expected to complete the project is 24 weeks. The table below shows the timeline.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Task | Start | End | Hours | Dependencies | Resources |
| 1. Requirement Analysis | 6/1/22 | 6/2/22 | 10 | None | QA Team, Development Team, Stakeholders |
| 2. Kickoff | 6/3/22 | 6/4/22 | 10 | Task 1 | QA Team, Development Team, Stakeholders |
| 3. Planning  (**Phase 1 start**) | 6/5/22 | 6/8/22 | 30 | Task 2 | Development Team |
| 4. Requirements | 6/9/22 | 6/19/22 | 100 | Task 3 | Development Team, Stakeholders |
| 5. Design | 6/20/22 | 6/30/22 | 100 | Task 4 | Development Team |
| 6. Data Analysis (**Phase 1 end**) | 7/1/22 | 7/10/22 | 90 | Task 5 | Development Team |
| 7. User interface development (**Phase 2 start**) | 7/11/22 | 8/04/22 | 240 | Task 1-6 | Development Team |
| 8. Temperature prediction development (**Phase 2 end**) | 8/05/22 | 8/28/22 | 230 | Task 7 | Development Team |
| 9. QA Testing (**Phase 3 start**) | 8/29/22 | 9/10/22 | 120 | Task 1-8 | QA Team |
| 10. Use feedback to improve product | 9/11/22 | 9/24/22 | 130 | Task 9 | QA Team, Development Team |
| 11. Deliver Prototype | 9/25/22 | 10/09/22 | 140 | Task 1-10 | Development Team, Stakeholders |
| 12. Acceptance Testing | 10/10/22 | 10/23/22 | 130 | Task 1-11 | QA Team, Stakeholders |
| 13. Improve product with feedback  (**Phase 3 end**) | 10/24/22 | 11/4/22 | 110 | Task 1-12 | QA Team, Development Team |
| 14. Product deployment (**Phase 4**) | 11/5/22 | 11/16/22 | 110 | Task 1-13 | QA Team, Development Team, Stakeholders |
| **Total:** | 6/1/22 | 11/16/22 | 1550 |  |  |

# Section C:

See Application

# Section D:

## 1. Project Purpose

The purpose of the project was to create a temperature prediction application that used linear regression to predict future temperatures and to create graphical representations of the previous temperature data that which the predictions were based.

The temperature prediction application was created to give consumers a weather forecast, which was not available in any capacity. Users that were looking for forecasting information were seeking out that information from competitors. Upon implementation of the temperature prediction application, users were able to view weather forecasting, therefore, increasing user retention. The dashboard hosted the prediction application as well as the graphical displays of previous weather data. The prediction application could have been improved with more previous temperature data to increase the accuracy of the linear regression model.

## 2. Datasets

The dataset used in creating the temperature prediction application was attained from kaggle.com. The dataset can be viewed here: <https://www.kaggle.com/datasets/sudalairajkumar/daily-temperature-of-major-cities>. The data is stored as a CSV file. The original dataset has much more information than what is needed. An example is can be seen in figure 1 below.

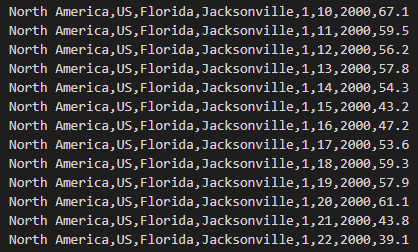


Figure 1

The data for Indianapolis was extracted from this data before being imported into the application. Some unnecessary data was left in the dataset including Region, Country, State, and City. The Indianapolis data can be seen below in figure 2.

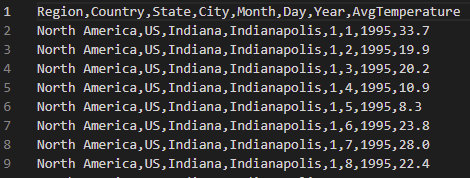


Figure 2

A column was added to the dataset called ‘Season’; this column was added to store a variable for which season each day falls. The snippet of code can be viewed in figure 3 below.

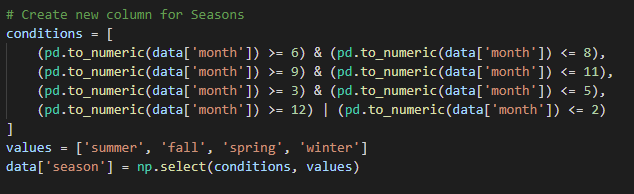


Figure 3

The dataset was checked before it was imported for bad data, such as incorrect information or no information at all, and the dataset was found to have no such anomalies. The graphical displays of the data used datasets created from the original dataset. The new datasets created only extract the information that is used in the graphical representations.

## 3. Data Product Code

The predictive model used to create the temperature prediction application was linear regression. The regression model used the python external libraries math and pandas. To create the linear regression model, the following mathematical equation was used: ŷ = bX + a. In this formula, ‘ŷ’ is the expected temperature value for the given date, ‘b’ is the sum of products divided by the sum of squares, ‘X’ is the year of the predicted date, and ‘a’ is the average temperature minus ‘b’ times the average year. The following code snippet shown in figure 4 demonstrates how this was completed.

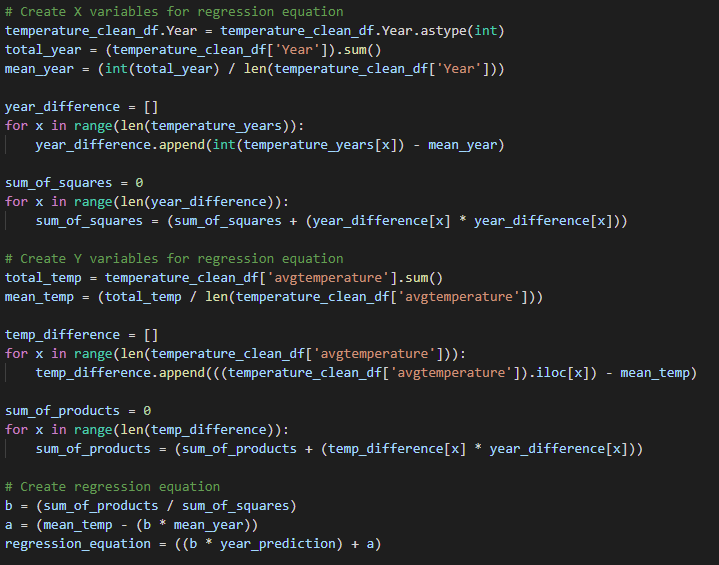


Figure 4

The temperature prediction is also displayed with a possible deviation. This deviation is calculated using root mean square error. This was calculated using the python external libraries NumPy and math. The code snippet shown in figure 5 below shows how this was calculated.

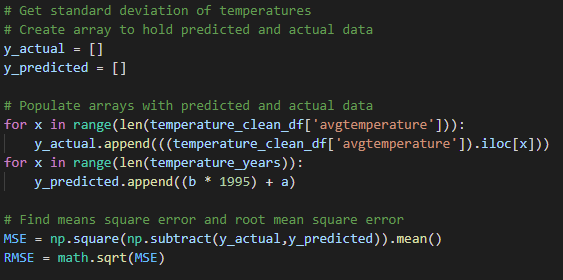


Figure 5

The descriptive data products created are as follows: a bar graph displaying the average seasonal temperature per year, a line graph showing the average temperature each day by month per year, a line graph showing the average temperature each day per month over the last 25 years, and a scatter plot with a trend line showing the average temperatures of each year since 1995. The values used in each graphical representation are added to a pandas data frame, and the data frame is then used as an input into an Altair chart. Figure 6 below shows the creation of the scatter graph with the trend line of the yearly data.

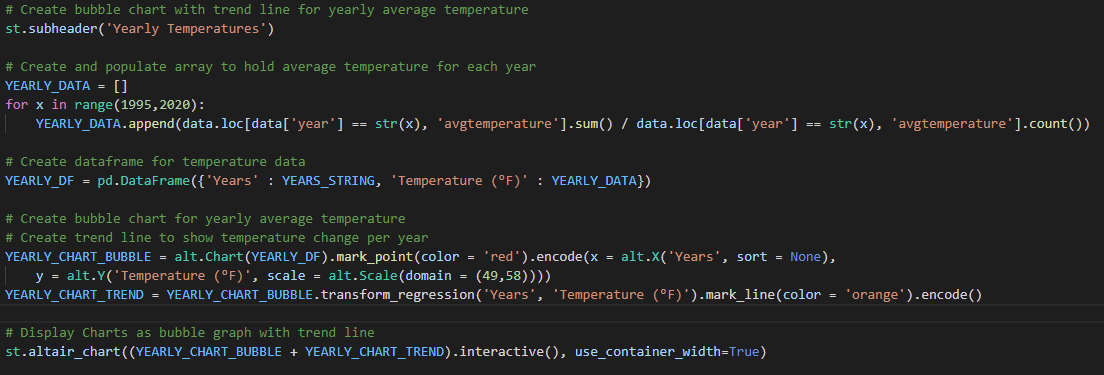


Figure 6

The source code is provided with the submission as well as held on GitHub and hosted on Streamlit.

## 4. Hypothesis Verification

The initial hypothesis predicted that if given the previous years’ temperatures, then future temperatures can be predicted. This prediction would improve customer retention and expand the customer base for Indianapolis Weather Association. The accuracy of this hypothesis will be discussed later in the accuracy analysis. The hypothesis can be partially verified but will require a larger dataset to give a more definitive answer. The current dataset can give a less accurate prediction of future weather temperatures, and the farther away from the current date, the less accurate the prediction will become. With more data, the predictions will be able to more accurately predict temperatures.

## 5. Effective Visualizations and Reporting

The application has three categories for its graphical displays: seasonal, monthly, and yearly. The seasonal display is a bar graph displaying the average seasonal temperature per year. The monthly displays are a line graph showing the average temperature each day by month per year and a line graph showing the average temperature each day per month over the last 25 years. The yearly display is a scatter plot with a trend line showing the average temperatures of each year since 1995.

The seasonal bar chart shows the average temperature per year by the seasons. The seasons are defined as winter, spring, summer, and fall. The date ranges are selected based on the meteorological seasons: winter is from December 1st through February 28th/29th, spring is from March 1st through May 31st, summer is from June 1st through August 31st, and fall is from September 1st through November 30th. There is a slider to view this temperature each year from 1995 to 2019. This view is helpful to identify when temperature increases and decreases will most likely begin.

The first monthly line graph shows the temperature of each month as a line, beginning on the month’s first day and ending on its last. This view is more compact than viewing the temperature as a long line starting from January 1st through December 31st. This compact view makes it easier for consumers to view the data. The first line graph has a slider to change the year from 1995 to 2019. The second monthly line graph is the same setup as the first, except the temperature for each day is the average temperature of that day since 1995.

The yearly scatter plot and trend line shows the average temperature of the year since 1995 plotted on the graph. A trend line is then placed over the data to show the expected increase or decrease in temperatures in the future. With the data given, the temperature is expected to increase slightly. The trend line relies heavily on the data of previous years, therefore, the accuracy of the line is not perfect. This will be discussed more in the accuracy analysis, but with more data to give the system, this trend line could be used to view possibly increases in temperature per year for Indianapolis.

## 6. Accuracy Analysis

As stated earlier in the hypothesis verification, the accuracy of the predictive model is less ideal than preferred. The accuracy is lacking because the amount of data used to predict the future temperature is not enough to give an accurate prediction. The farther into the future the prediction is the less accurate it will be. The deviation of the data is also fairly large. This is also because the amount of data used is not enough. Although the accuracy is not as high as we would want, it still offers a decent prediction.

As stated earlier in the effective visualizations and reporting, the accuracy of the trend line shown in the yearly scatter plot also suffers from the same lack of data as the prediction model. The same accuracy problems and solutions can be said for this. Eliminating some of the unnecessary data from the dataset would allow for adding more data to be added to the dataset without increasing response time.

## 7. Application Testing

Application testing was completed throughout the creation of the product. Testing was done upon completion of a feature to confirm the application performed correctly. Upon completion of every feature, the quality assurance testing was completed. This confirmed that all features were added to the project, and all features performed as expected per the requirements documentation. After the quality assurance testing was completed and the project was approved by the QA team, the acceptance testing was completed. This test confirmed with the Indianapolis Weather Association and its stakeholders that the application performed up to their standards and that nothing was needed to be added to meet the requirements and expectations.

## 8. Applications Files

The application files are held on a GitHub repository and hosted on Streamlit. The files can be viewed here: <https://github.com/RylanMattocks/CapstoneProject>. The repository includes the following files:

* Insert file names and what they do here
* README.md is a file explaining how to access the application from GitHub.
* Dbdata.db is a database file containing the login information for users. The data is hashed before it is entered into the database.
* Indianapolis\_temp.csv is the CSV file containing the temperature data information.
* Invalidenteries.log is a log file containing the errors for when invalid input is entered. This includes the login, prediction data, and when invalid data is trying to be accessed.
* Requirements.txt is a text file required by the hosting service Streamlit. It holds the external libraries that need to be imported as well as the version they are running.
* Wgumain.py is the python file containing the code used for the temperature prediction application

## 9. User’s Guide

The application is hosted on Streamlit. The web application can be accessed using the following steps:

1. Proceed to [https://share.streamlit.io/rylanmattocks/capstoneproject/main/wgumain.py](https://protect-us.mimecast.com/s/mblsCXDXNoS4ORvqwT6ZSaG?domain=share.streamlit.io)
2. On the left-hand side, there will be a sidebar with a menu to log in or signup.
   1. To signup, select the signup dropdown.
      1. Enter a username and password then press the sign up button
   2. To log in without creating an account use testapp as the username and password
3. Proceed to the login menu
4. Enter the username and password into the fields and select the checkbox
   1. Make sure the checkbox stays selected, if it is unchecked you will be logged out
   2. You will see a prompt saying you have logged in as your username.
5. From here you have four dropdown options.
   1. The seasonal tab will show you the graph of temperatures by season.
      1. You can move the slider to change the year the data will show.
   2. The monthly tab will show two line graphs depicting the temperature data of each month per day.
      1. The slider will change the year of the top graph, the bottom graph is the average of all years.
   3. The yearly tab will show the average temperature of each year with a trend line displaying the predicted temperature growth.
   4. The prediction tab will allow you to see a predicted temperature for a given date.
      1. Enter the month, day, and year that you want the temperature prediction for, and press enter. (You will receive error messages for non-existent days)

## 10. Summation of Learning Experience

The capstone project made use of all my knowledge and more concerning programming and machine learning. My previous experience in programming from previous WGU classes such as the artificial intelligence and software classes helped me in developing the application. Other WGU classes as well as my work experience helped in choosing development methodologies and project planning. Many new concepts were a struggle for me, and I had to seek information to complete the project. Throughout the development, I noticed many things that I could have done better. I created more data frames than I needed to display the graphical data. I had to change some displays after creating them to display the data more efficiently once I was more proficient with how to create the displays. I learned a lot about data manipulation. Initially, I used many extra steps to obtain the data needed from the dataset, but by the end of the project, I had learned better and more efficient ways to obtain that data. I still have very little knowledge about the Streamlit web application, but I did learn enough throughout the project to be able to display the graphical data. The lack of strict requirements and outlining for the project forced me to come up with the backbone for the project. I believe this project helped me greatly in understanding how to accomplish developing a project as well as searching for the information needed.

# Section E:

## 1. Sources

Srk. “Daily Temperature of Major Cities.” *Kaggle*, 5 June 2020, https://www.kaggle.com/datasets/sudalairajkumar/daily-temperature-of-major-cities.