Weather Forecasting Using Machine Learning

Ayokunle Oluwole, Haoyue Lin, John Geng and Tina Saravi

Project Overview

Predicting weather forecasting with Machine Learning using:

1. Spark
2. SQL Database (Railway)
3. Python Pandas
4. Python Matplotlib
5. Scikit-learn
6. Machine Learning (Time-Series Analysis)

Data Collection

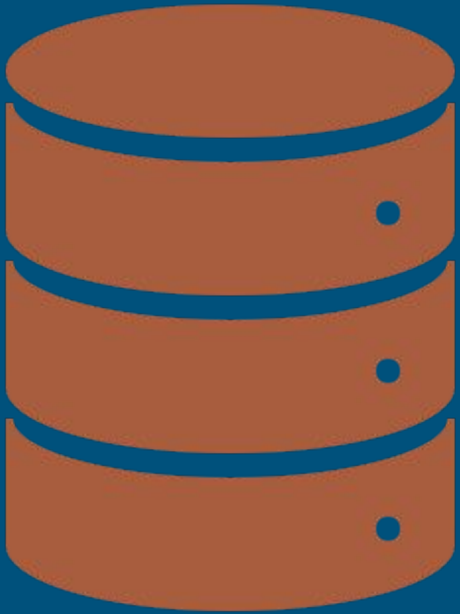
1. Climate Weather Data (Government of Canada)

   Weather Stations:

   a. Billy Bishop Airport (1840 -2006)

   b. Buttonville Airport (1986 - 2015)

   c. Toronto Pearson Airport (1937 - 2013)



Data Preparation

PySpark, is a powerful framework that was used to process and clean each CSV data.

1. Start a SparkSession and load each CSV file for cleaning.
2. Uploaded the 3 csv files to Railway to host a SQL server for transformation of dataset.
3. Load the 3 tables (Pearson, Billy Bishop and Buttonville) into spark from SQL
4. Review the loaded the 3 tables, to concatenated the 3 tables together into one DataFrame
5. Data cleaning, removed all null values and convert all the data into correct data types for further data analysis
6. Data calculation find the average temperature for each Year/Months
7. Sort the data and export to csv file

Data Exploration

- Data From the year 1900 to 2013, it lists average temperature for each month
- There were many fields that is null. In order to find the average temperature of the 3 tables, we removed all the nulls.
- We reviewed the data, removed all the unnecessary columns, from 29 columns to 4 columns
- We then furthermore bring the table in 2 columns, one column been Date-Time in "YYYY-MM" format and one column been Mean Temperature column in Integer.

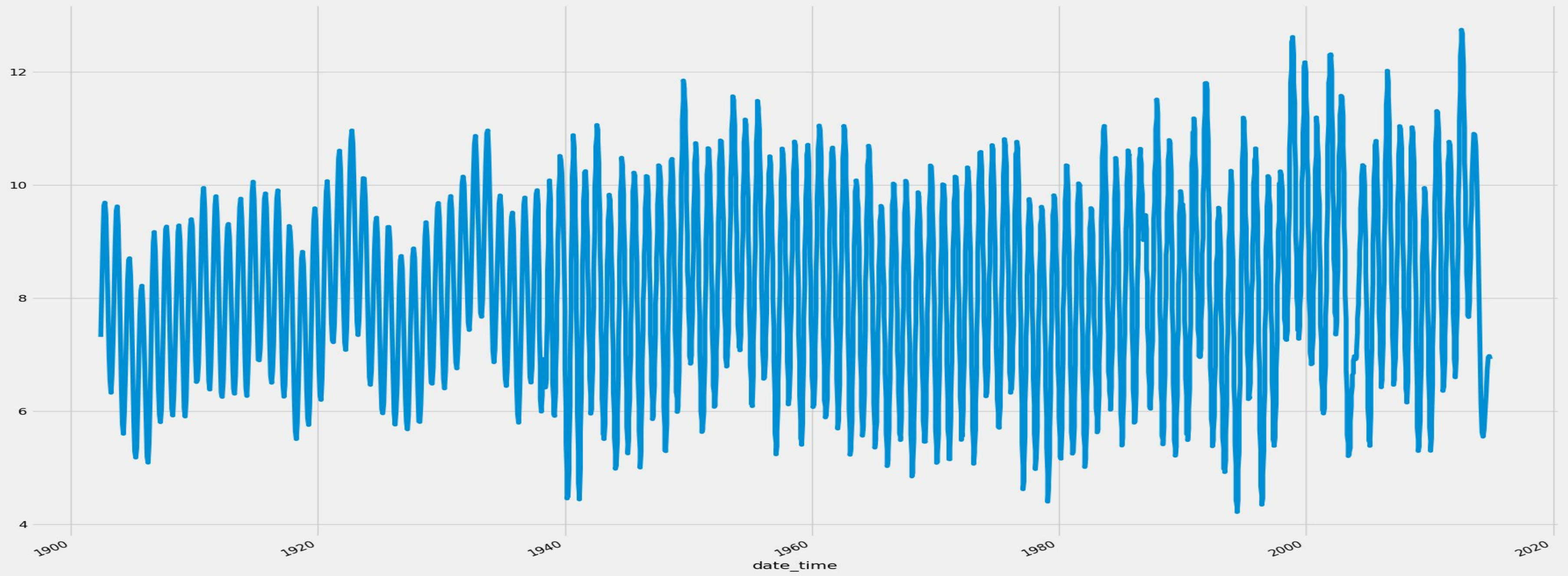
	year	month	date_time	mean_temp
0	1840	APR	1840-04-01	6.7
1	1840	AUG	1840-08-01	19.0
2	1840	DEC	1840-12-01	-4.9
3	1840	JUL	1840-07-01	19.5
4	1840	JUN	1840-06-01	15.8
5	1840	MAR	1840-03-01	1.2
6	1840	MAY	1840-05-01	12.4
7	1840	NOV	1840-11-01	2.1

	mean_temp
date_time	
1900-01-01	-3.4
1900-02-01	-6.7
1900-03-01	-4.8
1900-04-01	7.5
1900-05-01	12.9

Data Analysis & Visualization

Time Series Model

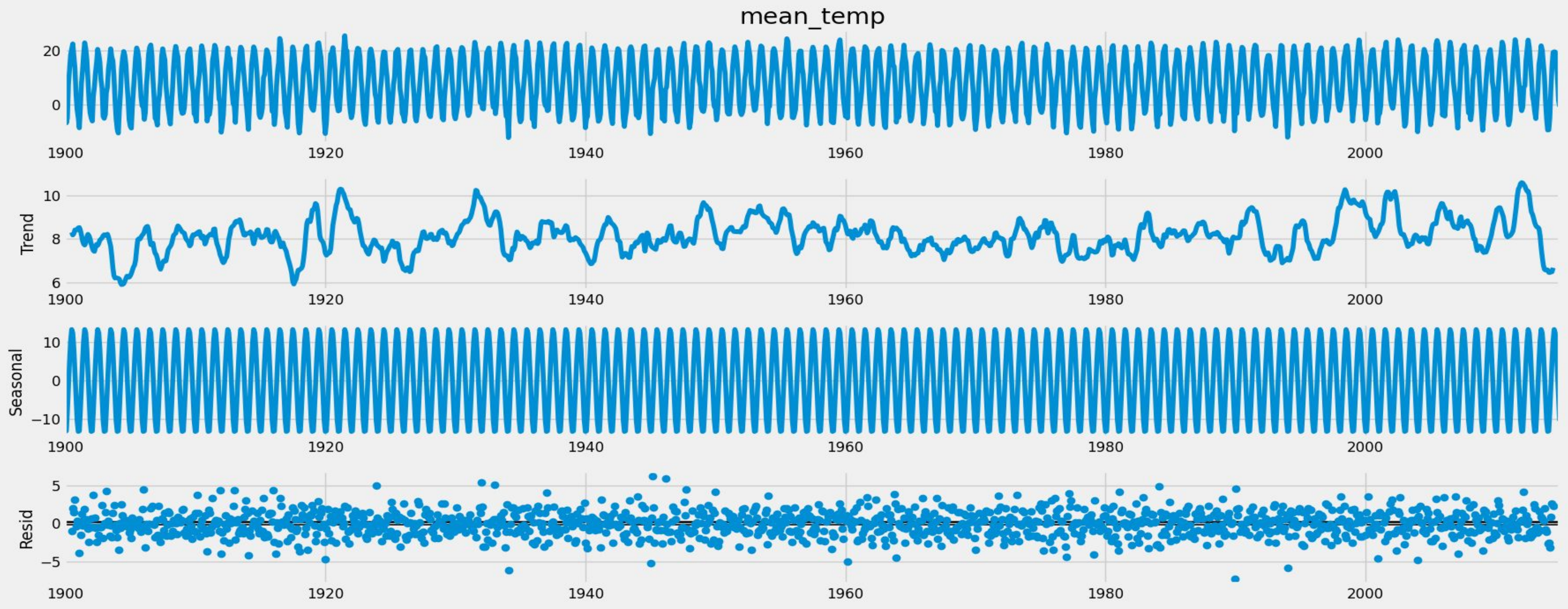
- Visualize the Mean Temperature using Time Series Data to see the patterns.
- The time-series exhibits a seasonality pattern: Low temperature in the beginning and end of the year, high in the middle of the year.



Data Analysis & Visualization

Time-series decomposition

- By analyzing the individual components of the time series: trend, seasonality, and noise.
- We can see that the temperature is unstable along the observed seasonality.

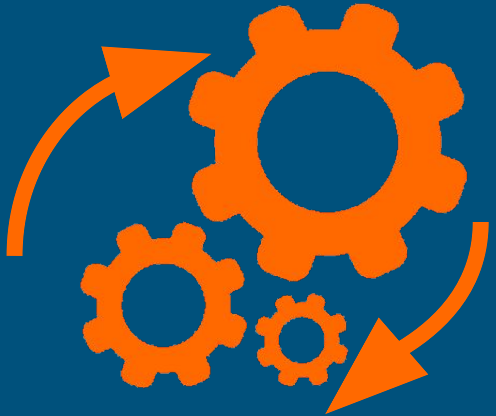


Modeling and Optimization

SARIMA Model

Stands for **Seasonal Autoregressive Integrated Moving Average**

- Closely tied to the ARIMA Model that was developed in the 1970
- ARIMA model was an extension of the ARMA model from 1950s
- ARMA model was created to capture the relation between observation at different time lag and moving average components in a time series.
- SARIMA model recognized the presence of seasonality



Modeling and Optimization

Deploying the SARIMA Model

- The parameters of the SARIMA model are denoted as (p, d, q)(P, D, Q, s), q are:
- **p, d, and q** represent the non-seasonal order of the AR, I, and MA components, respectively.
- **P, D, and Q** represent the seasonal order of the AR, I, and MA components, respectively.
- **s** represents the length of the seasonal cycle
- By estimating the appropriate values for these parameters we can use this model on historical data to predict future forecasts patterns in a time series dataset

```
for param in pdq:
    for param_seasonal in seasonal_pdq:
        try:
            mod = sm.tsa.statespace.SARIMAX(y,
                                            order=param,
                                            seasonal_order=param_seasonal,
                                            enforce_stationarity=False,
                                            enforce_invertibility=False)
            results = mod.fit()
            print('ARIMA{}x{}12 - AIC:{}'.format(param, param_seasonal, results.aic))
        except Exception as e:
            print('An exception occurred:', e)
            continue
```

Modeling and Optimization

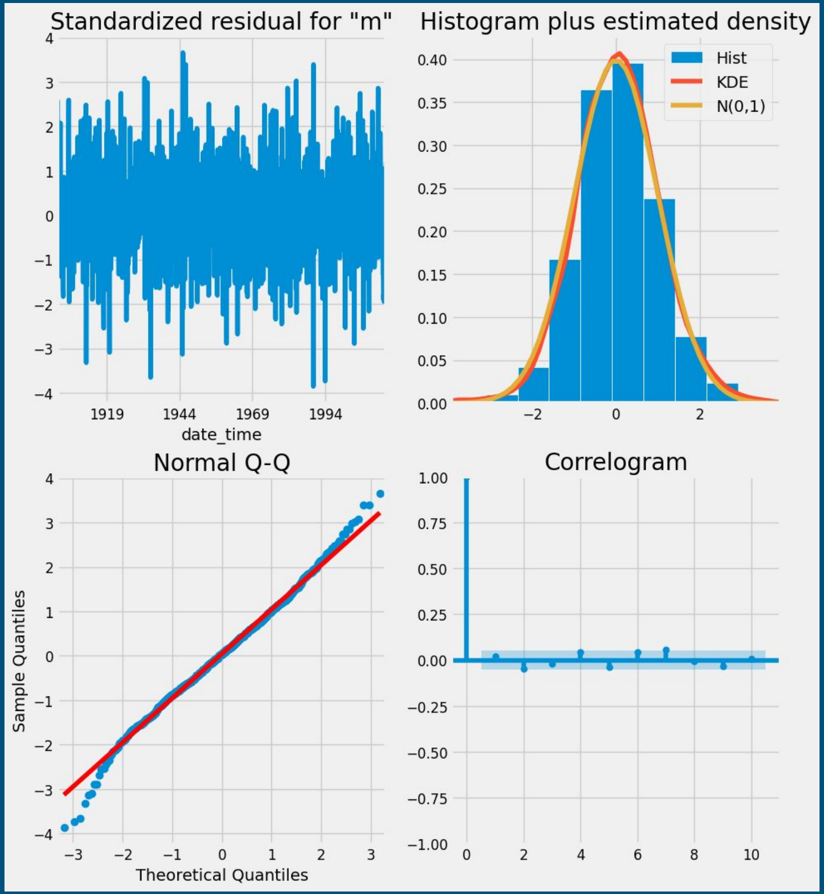
- Optimized by using **Grid Search** technique for parameter selection
- AIC (Akaike Information Criterion) measures how well the model fits in the data. The model with the lowest AIC score is considered to be the best-fitting model
- The parameters SARIMAX (1,1,1)x(1,1,1,12) yielded the lowest AIC value of 5468.76
- Mean squared error of 3.24 is the error is the difference in temperature by 3.24 degrees celsius

```
[32] ARIMA(1, 0, 0)x(1, 0, 0, 12)12 - AIC:6382.456914089306
     ARIMA(1, 0, 0)x(1, 0, 1, 12)12 - AIC:5526.7200342408405
     ARIMA(1, 0, 0)x(1, 1, 0, 12)12 - AIC:6005.880277189616
     /usr/local/lib/python3.10/dist-packages/statsmodels/base/model
       warnings.warn("Maximum Likelihood optimization failed to "
     ARIMA(1, 0, 0)x(1, 1, 1, 12)12 - AIC:5475.855096781233
     ARIMA(1, 0, 1)x(0, 0, 0, 12)12 - AIC:8029.825462734481
     ARIMA(1, 0, 1)x(0, 0, 1, 12)12 - AIC:7529.501221802674
     ARIMA(1, 0, 1)x(0, 1, 0, 12)12 - AIC:6395.104588045029
     ARIMA(1, 0, 1)x(0, 1, 1, 12)12 - AIC:5456.289718401156
     ARIMA(1, 0, 1)x(1, 0, 0, 12)12 - AIC:6377.287295897335
     ARIMA(1, 0, 1)x(1, 0, 1, 12)12 - AIC:5509.490417112486
     ARIMA(1, 0, 1)x(1, 1, 0, 12)12 - AIC:6000.969903767579
     ARIMA(1, 0, 1)x(1, 1, 1, 12)12 - AIC:5457.622592401087
     ARIMA(1, 1, 0)x(0, 0, 0, 12)12 - AIC:7864.791858056271
     ARIMA(1, 1, 0)x(0, 0, 1, 12)12 - AIC:7560.999257701889
     ARIMA(1, 1, 0)x(0, 1, 0, 12)12 - AIC:6806.459491433872
     ARIMA(1, 1, 0)x(0, 1, 1, 12)12 - AIC:5850.439062335381
```

The parameters with the lowest AIC score is ARIMA(1, 1, 1)x(1, 1, 1, 12)12 - wit
5468.76434141344.

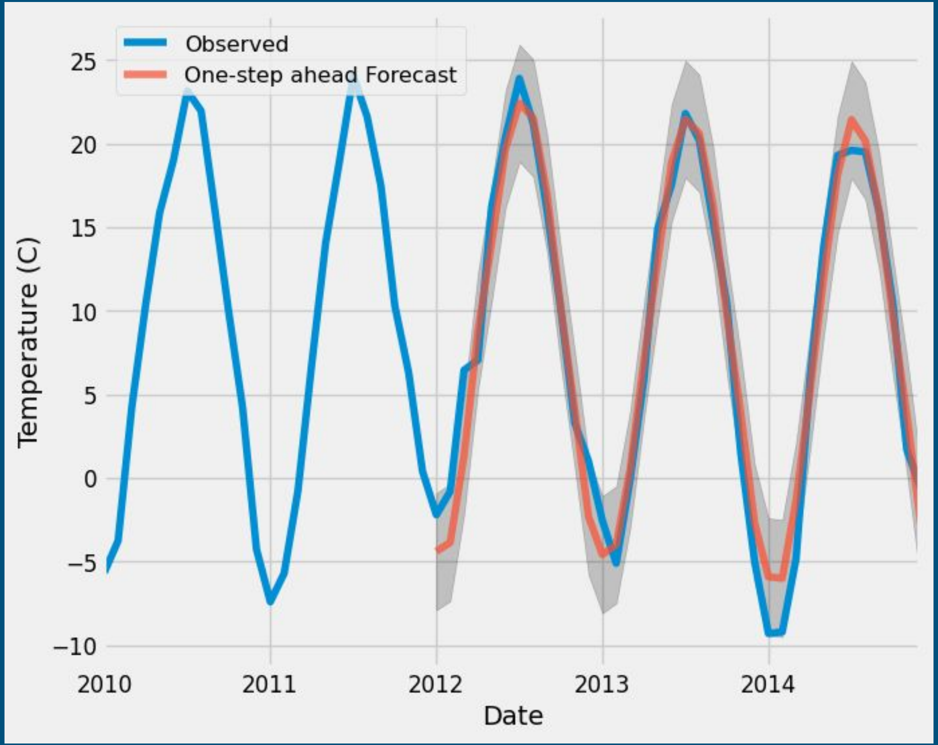
Model Diagnostics

1. Residuals of the model are uncorrelated
2. Normally distributed with zero mean
3. Exhibit no obvious Outliers



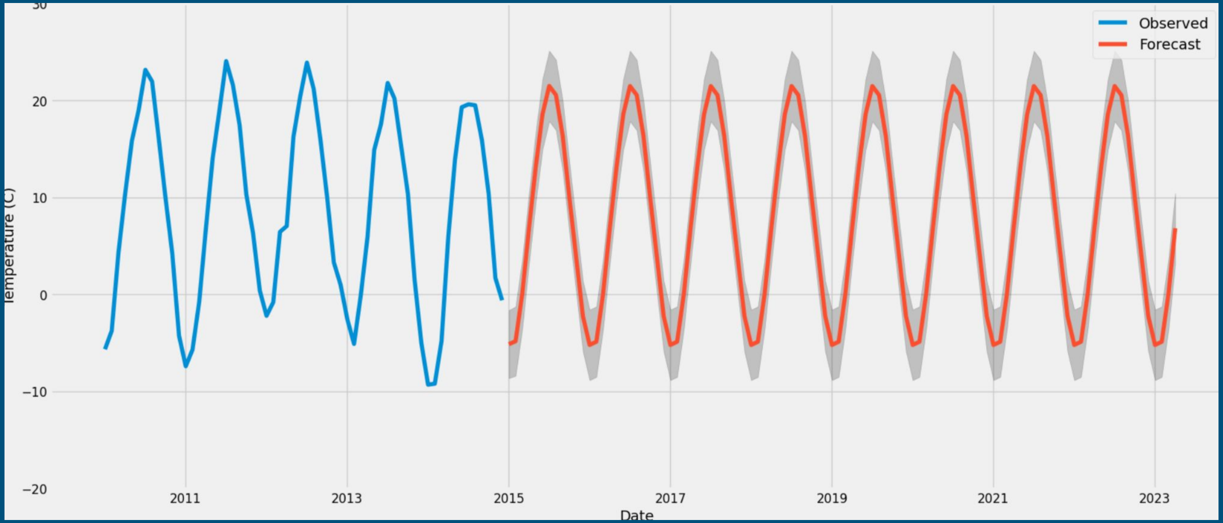
One-Step Ahead Forecast vs Observed

- The forecasted data shows a remarkable alignment with the observed data
- Successfully captures the seasonal patterns
- Mean Squared Error (MES): 3.235
- Root Mean Squared Error (RMSE): 1.799



Forecasting into the Future

- Remote PostgreSQL server via Railway.app
- Seasonal ARIMA (SARIMA) from Statsmodels library



Resources

https://climate.weather.gc.ca/historical_data/search_historic_data_e.html

Pierre, Sadrach. “A Guide to Time Series Forecasting in Python” *Medium,* 4 November 2022, https://builtin.com/data-science/time-series-forecasting-python

Li , Susan. “An End-to-End Project on Time Series Analysis and Forecasting with Python.” Medium, 8 July 2018, towardsdatascience.com/an-end-to-end-project-on-time-series-analysis-and-forecasting-with-python-4835e6bf050b.