Weather Forecast

Sanchana Mohankumar
Gauri Damle

Introduction

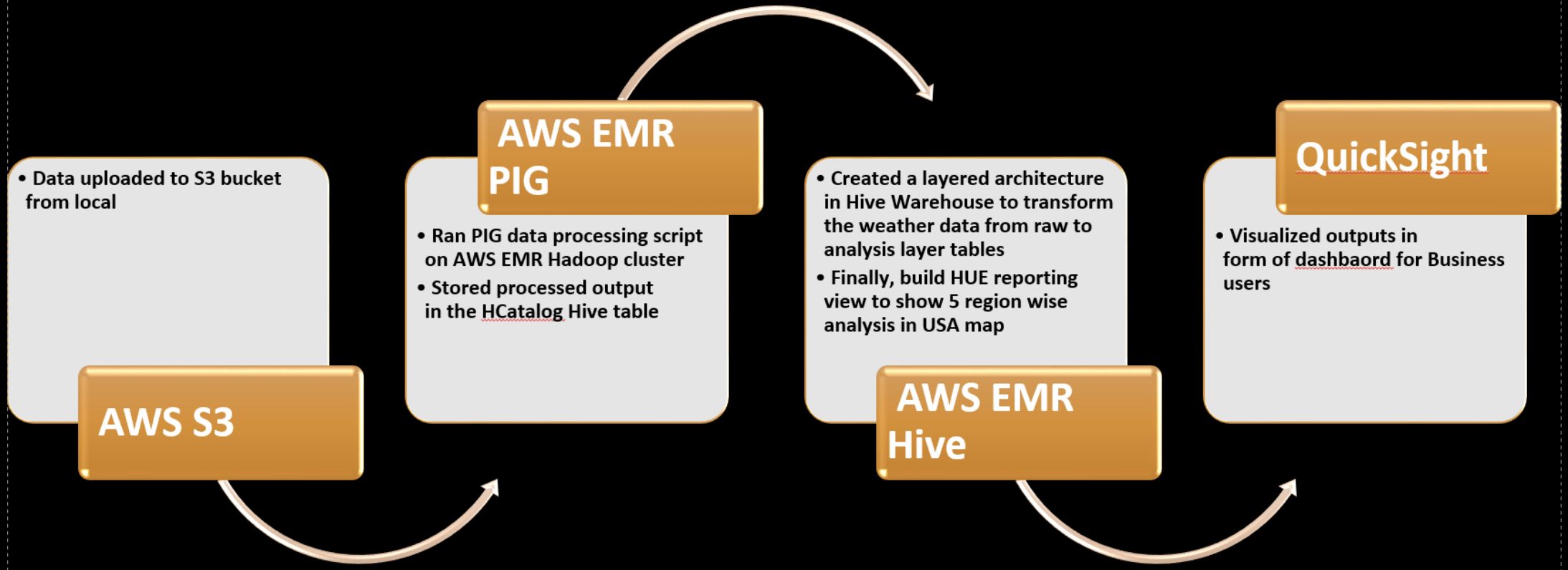
- **The aim of this project is to provide accurate and timely weather forecasts for a specific location or region leveraging efficient parallel processing techniques**

- **As per survey by National Weather Service 95% Americans use forecasts for making informed decisions**
- **A study by the National Center for Atmospheric Research found that improving weather forecasts could increase crop yields by up to 3.4%**

Data

- Extracted daily data from 1973 to 2023 for all states of US from visual crossing platform
- Utilized Latitude Longitude for all states of US from Kaggle
- Considered major columns such as state, temp, dew, humidity, precipitation, snow, windspeed, solarradiation, uvindex, datetimevalue, statecode, latitude, longitude and city

Major Task 1 -Workflow



Performance Analysis

Program	Time Taken	Program type
Step1	26 seconds	Hive Script - DDL
Step2	4 minutes 14 seconds	PIG Script - Data Processing
Step3	36 seconds	Hive Script - Analysis

Config 1 - 1 Primary and 6 core Nodes cluster

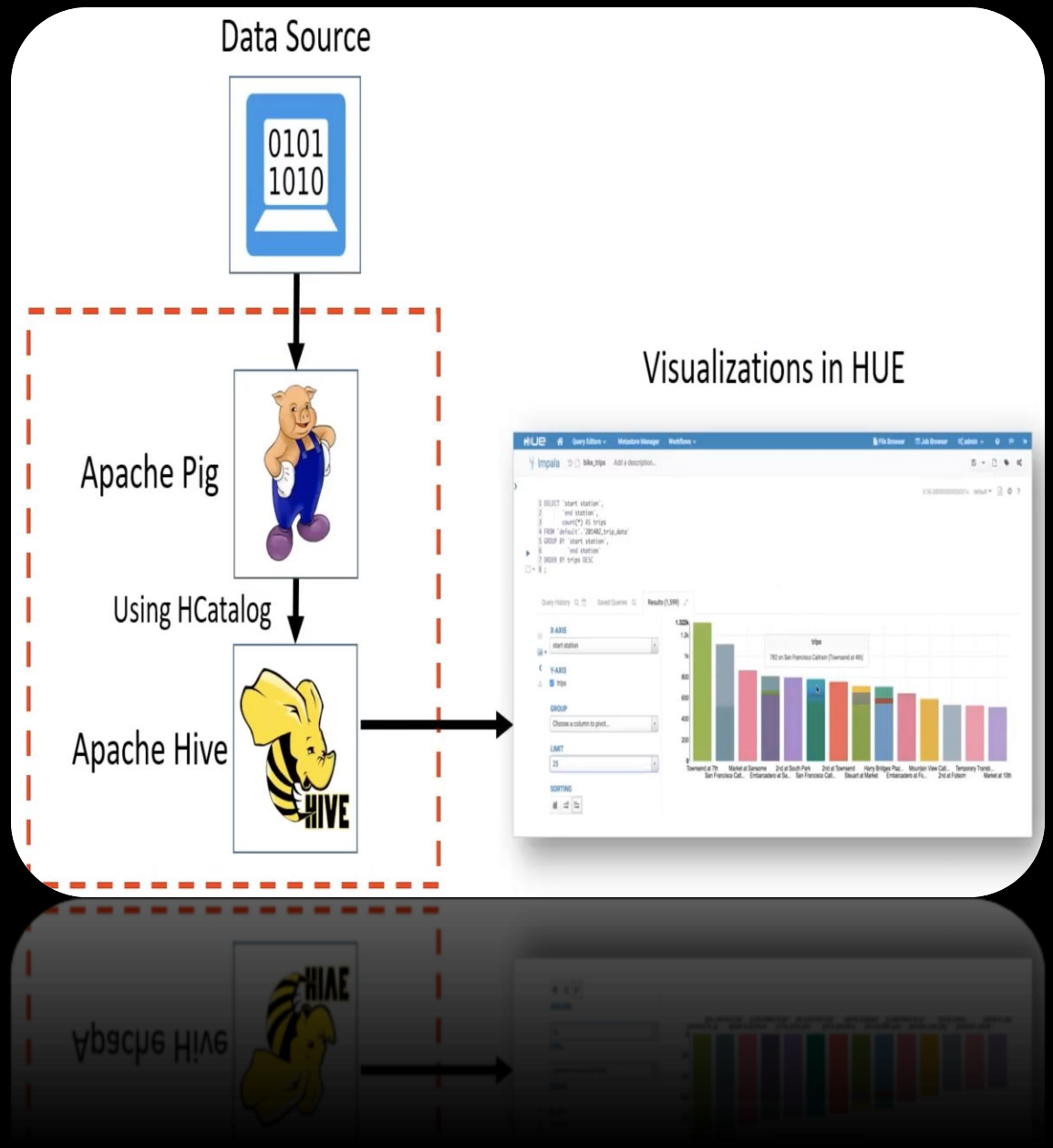
Program	Time Taken	Program type
Step1	27 seconds	Hive Script - DDL
Step2	5 minutes 24 seconds	PIG Script - Data Processing
Step3	37 seconds	Hive Script - Analysis

Config 2 - 1 Primary and 3 core Nodes cluster

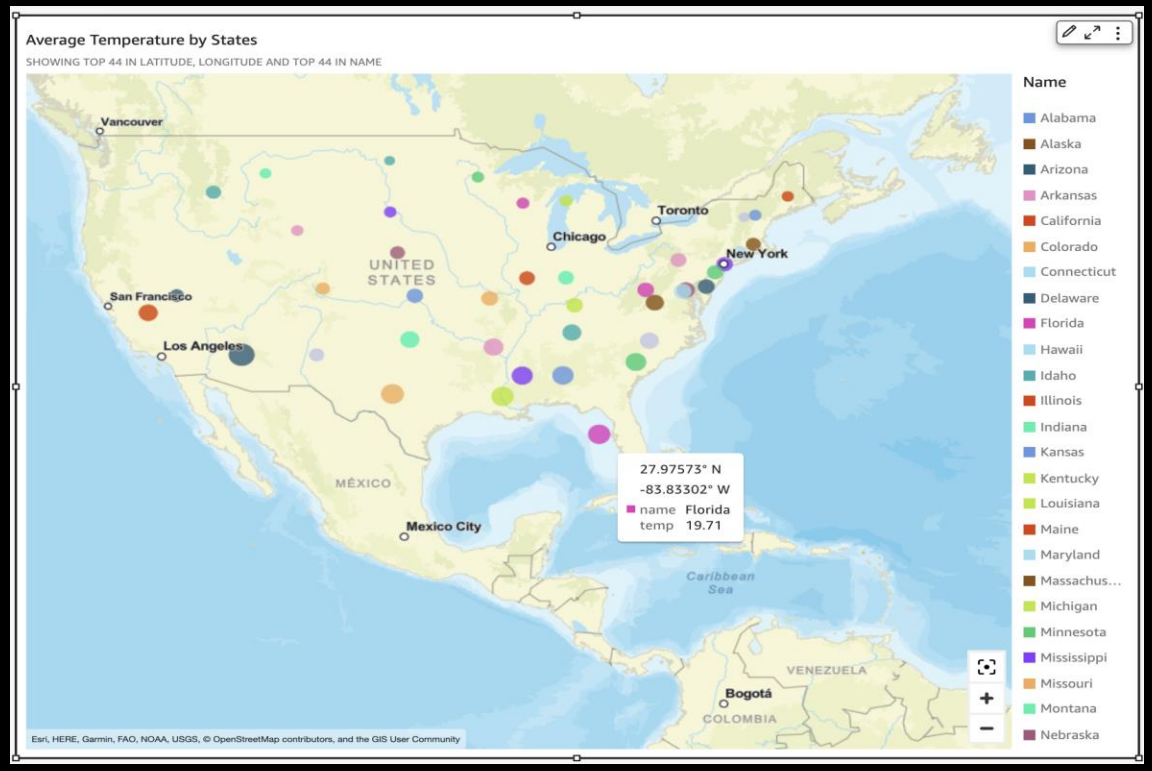
Challenges

- Used HCatStorer() in pig script to store intermediate output in Hive metastore as Hive script can directly read from it instead of writing to S3

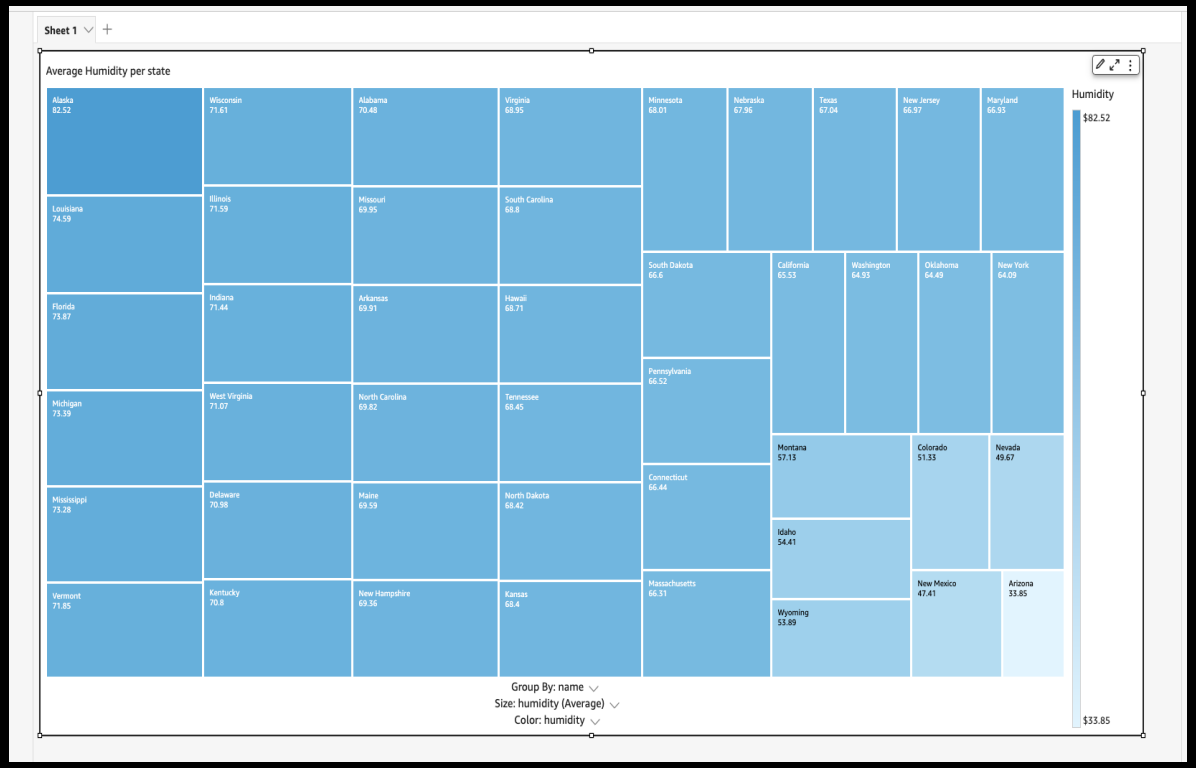
- Issue faced while testing PIG-HIVE data pipeline indicating datatype mismatch. Example output in pig char-array should correspond to the string datatype in Hive



QuickSight Output

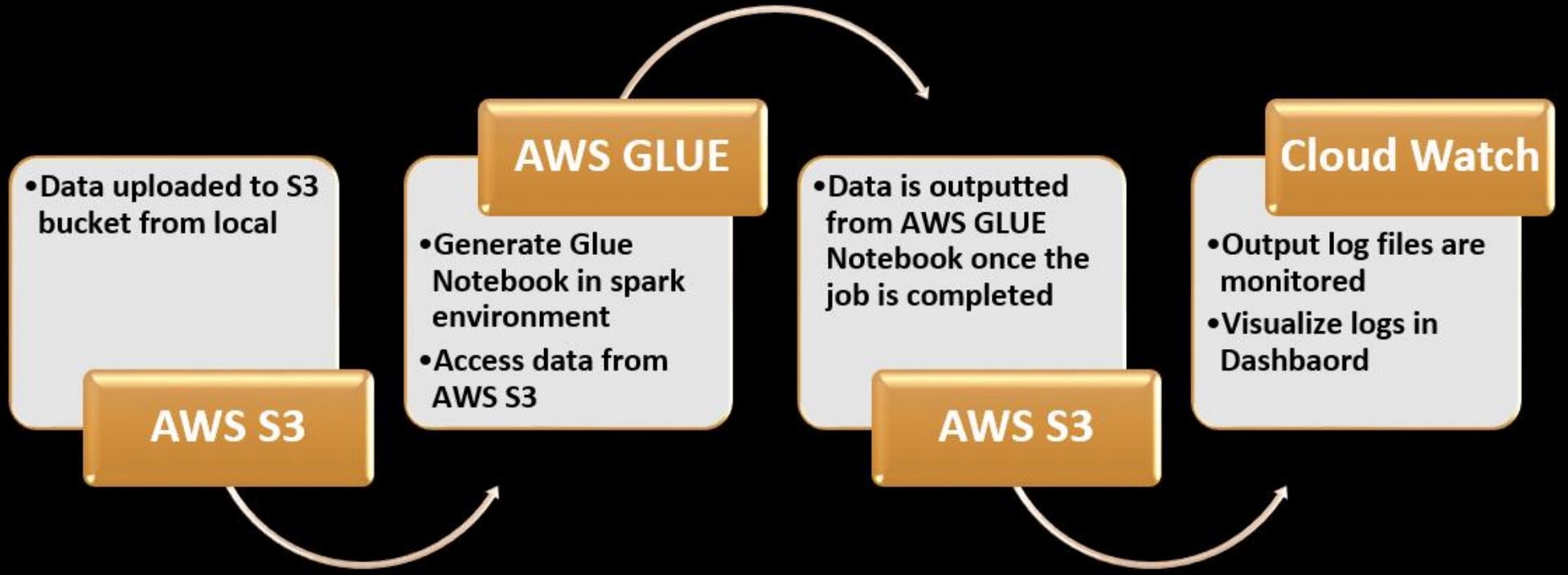


Plot 1: Average Temperature per state



Plot 2: Average Precipitation per state

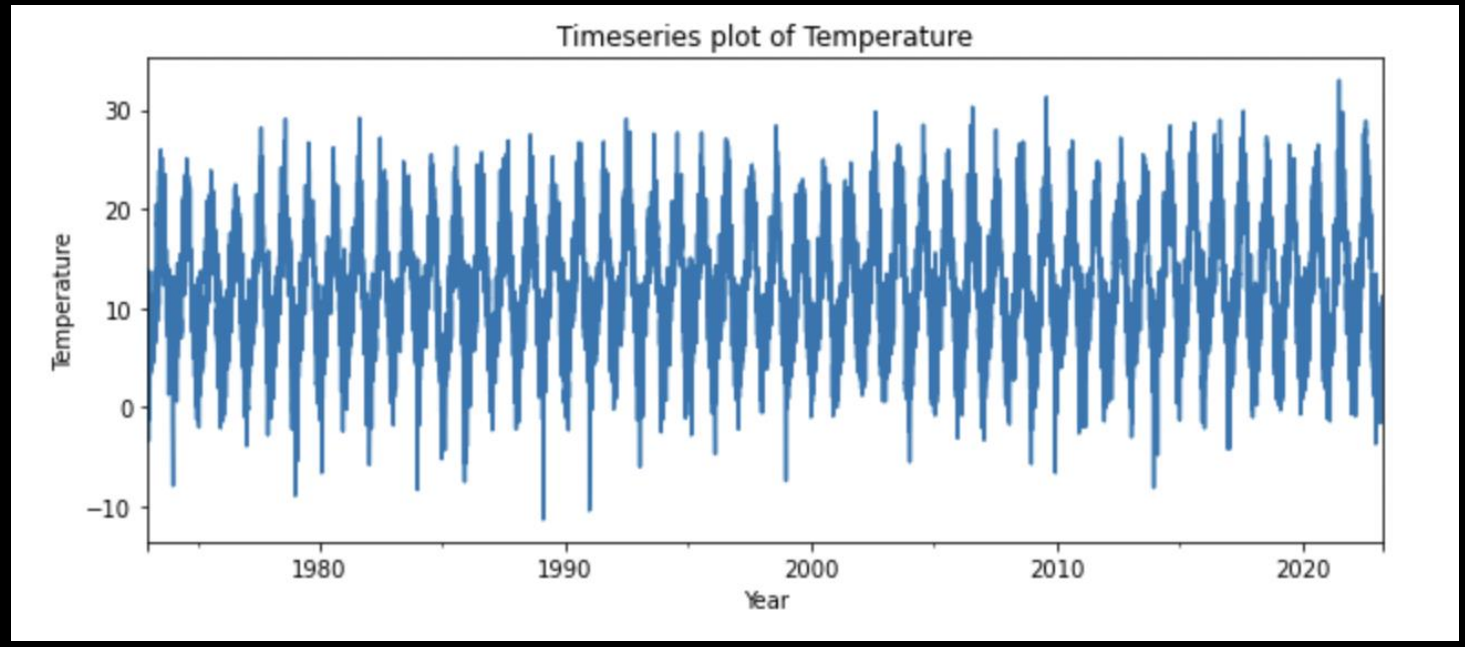
Major Task 2 -Workflow



Analysis Task

Before modelling a few pre-processing processes must be satisfied.

- Timeseries must be stationary
- Stationary of the timeseries data was proved by conducting ADF Test resulting in p-value <0.05



Performance Analysis

Model	State	RMSE	MSE
Arima Model	Oregon	2.769	7.67
LSTM Model	Oregon	2.101	4.41

Table 1: ARIMA and LSTM Model output

Model	Workers	Time Taken
Arima Model	5	4 minutes 15 seconds
Arima Model	10	4 minutes

Table 2: ARIMA Model AWS GLUE job output

- Model ARIMA and LSTM are the models selected for Weather Forecast and chose ARIMA Model for deployment
- Utilized AWS GLUE spark environment to run job for ARIMA Model as show in table 2

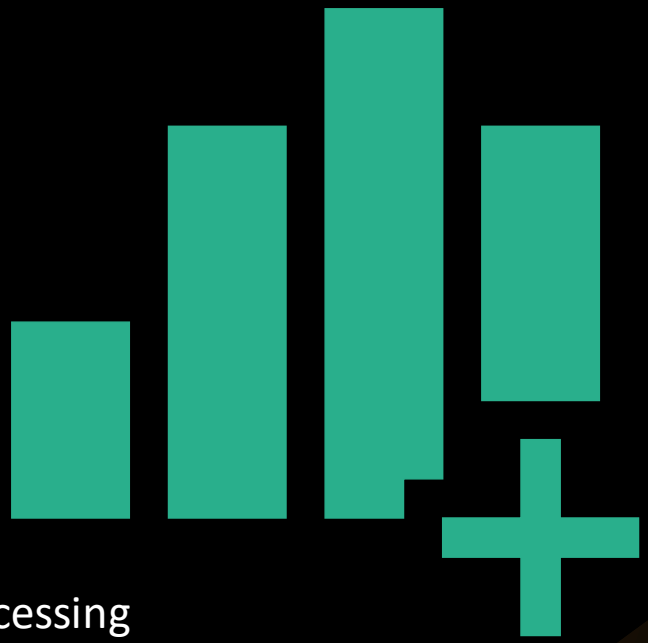
Challenges

Although seasonality appeared in the time series plot, the ADF test showed no seasonality. The SARIMAX model was employed to confirm the presence of seasonality as the ADF test may sometime fail to detect in presence of white noise then later confirmed there was no seasonality

When compared to setting up a spark environment locally, using AWS GLUE was more convenient and favorable in terms of scalability and cost

Future Work

- Improve the model's performance by investigating additional specialized models created by weather forecasting businesses, which will assist to boost prediction accuracy
- Access real time data to forecast data
- Add autoscaling services like AWS load balancer to existing pipeline



Conclusion

- To conclude, we used Serverless GLUE as well as EMR to build scalable Big data processing cloud pipelines
- Overall, we succeeded in analyzing performance of tasks running in parallel processing environments for a real-world big data use case

References

- https://www.visualcrossing.com/weather/weather-data-services/Washington/metric/2023-03-01/2023-03-27
- https://github.com/nachi-hebbar/ARIMA-Temperature_Forecasting/blob/master/Temperature_Forecast_ARIMA.ipynb
- https://www.freecodecamp.org/news/how-to-combine-multiple-csv-files-with-8-lines-of-code-265183e0854/
- https://www.machinelearningplus.com/time-series/arima-model-time-series-forecasting-python/
- https://www.kaggle.com/code/prashant111/complete-guide-on-time-series-analysis-in-python/notebook

THE END
Thank you !!