RMIT University Vietnam, School of Science and Technology

RMIT UNIVERSITY

Course code	EEET2574	
Course name	Big Data for Engineering	
Assessment name	Assessment 3: Group Project	
Lecturer	Arthur Tang	
Due date	19 Jan 2024	
Submission date	19 Jan 2024	
Group members	Huynh Van Anh	s3836320
	Nguyen Le Bao Han	s3894290
	Nguyen Thanh Sang	s3878340
	Lee Wonjun	s3773920
	Nguyen Dang Nhat	s3878292

I declare that in submitting all work for this assessment I have read, understood, and agreed to the content and expectations of the Assessment declaration.

Table of Contents

1. Overview

1.1. Problem statement & Importance

Agriculture plays a vital role in supplying food which is crucial to human life maintenance. Food security, which is people having physical and economic access to sufficient safe and nutritious food that meets needs for healthy life [1], produce wide range of positive impacts as not only improved health and poverty reduction, but also economic growth, job creation, trade opportunities, and increased global security and stability [2].

For better crop production, weather forecast has been used in agriculture. Growth of crops are highly related to weather as water availability, sunshine, and temperature. Based on weather forecast, farmers establish plannings for fertilization, disease management, and field operations. [3] They also prepare for extreme weather such as storms, hurricanes, floods, and droughts.

However, agriculture in the US is facing a huge challenge which is adopting to extreme climate change. Last year, California experienced one of the most severe long-term droughts of the past 1,200 years, and it affected cropland of rice resulting in over half the state's rice acres going unplanted. [4] High nighttime temperatures in 2010 and 2012 affected corn yields across the U.S. Corn Belt, and premature budding due to a warm winter caused $220 million in losses of Michigan cherries in 2012.[5]

Although agricultural productivity in U.S. increased, it is not more resilient to extreme weather. [4] Therefore, better predictions of weather should be done to be prepared for extreme weather as global boiling. [6]

1.2. Utilizing Big Data

Weather forecasts are already done by many national weather services and companies. For example, the national weather service of the USA operates with a budget of approximately $930 billion. [7] The reason weather service requires a lot of spendings is because it needs data collection from satellites, radars, and weather stations of various data as temperature, humidity, wind information, and atmospheric pressure. Those data are processed and predicted based on models on multiple supercomputers, and the result is analyzed to predict the weather.

The process of traditional weather forecast already uses big data for prediction. This project is done on similar methods, collecting data, and predicting based on trained models. Data of past will be collected to build the model that can predict the weather based on new input on current situation, and it will be tested by input of new data. After the model is built, it will process real time information to predict weather based on meteorology. After post-processing to improve readability for normal people, weather forecast is disseminated on various platforms for people.

This project aims to build a system that gets input data for training multiclass classification of weather type and predict weather based on live data from API based on AWS environment. Weather type indicates the status of weather as rain, cloudy, and fog. The target places of weather forecast are agricultural cities in the US, which is the world's largest producer of corn, third-largest producer of wheat, and fifth-largest producer of potatoes, [8] meaning demand for agricultural-weather forecast is big.

1.3. Roles and Contributions of group members

The following is the role and contribution of each group member.

Name	Role	Contribution
Huynh Van Anh	Product Manager	Design Pipeline, manage Kinesis usage
Nguyen Le Bao Han	Data Engineer & Analyst	Cleaning & Examining data, development of model
Nguyen Thanh Sang	Data Scientist & Analyst	Managing training dataset, development of model, building visualization dashboard
Lee Wonjun	Data Engineer	Cleaning data, manage presentation and report
Nguyen Dang Nhat	Data Scientist	Managing training dataset, development of model, building visualization dashboard

Table 1: Role and contribution of team members

2. Solution Design

2.1. Proposed solutions

The solution we propose is to create a machine-learning model that can generate weather forecast predictions based on historical weather data in cloud environments. In this project, we created a big data pipeline for efficiently extracting, loading, and transforming datasets from many sources. Then, multiple models were trained and thoroughly assessed to determine which model produced the most accurate weather forecasting predictions. Furthermore, we developed a user-friendly visualization dashboard that not only summarizes historical weather data but also displays future weather predictions, making the information more accessible and understandable to clients.

2.2. Datasets

2.2.1. Data Sources

There will be two key data sources in this project: OpenWeatherMap and Kaggle.

- Weather API from OpenWeatherMap

This API makes it simple for users to obtain essential meteorological data, short-term and long-term forecasts, and aggregated weather data. We used OneCall API 3.0. The list of parameters are shown below:

Parameters	**Description**
lat	Latitude, decimal (-90; 90). Latitude of the location
long	Longitude, decimal (-180; 180). Longitude of the location
exclude	By using this parameter, we can exclude some parts of the weather data from the API response. It should be a comma-delimited list (without spaces) Available values: • current • minutely • hourly • daily • alerts
appid	Unique API key

Table 2.1: Parameters of OpenWaeatherMap API

We utilize the Weather API to retrieve data two days ahead of when it is called. The data is recorded every hour. Fields to be ingested:

Fields	**Description**
dt	Time of the forecasted data, Unix, UTC
temp	Temperature (Kelvin)
pressure	Atmospheric pressure on the sea level, hPa
humidity	Humidity, %
Wind speed	Wind speed (m/s)
weather_description	Weather condition

Table 2.2: Fields of OpenWeatherMap API

- Kaggle

Kaggle is the world's biggest data science community, with powerful tools and resources to support data-driven exploration and learning. It also offers Kaggle datasets, a comprehensive collection with diverse fields and interests. This rich repository not only supports a wide range of dataset publication formats, but also serves as a beneficial platform for learners, researchers, and professionals to access diverse datasets.

- Historical Hourly Weather Data 2012-2017

This dataset contains five years of high temporal resolution (hourly measurements) data of various weather attributes from 2012 to 2017. The dataset was collected for 30 US and Canadian Cities, as well as 6 Israeli cities, using Weather API on the OpenWeatherMap website.

We have selected some fields to extract the weather information of some US cities from 2015 to 2017.

Fields	**Description**
datetime	Timestamp of weather observation record
cities	List of US cities
humidity	Humidity (%)
pressure	Atmospheric pressure on the sea level (hPa)
temperature	Temperature (Kelvin)
weather_description	Weather condition
wind_speed	Maximum wind speed for the date specified in the request (metre/sec)

Table 2.3: Fields of Historical Horuly Weather Data

- US Accidents (2016 - 2023)

This dataset is a national car accident dataset that covers 49 states in the United States and was collected in real time using several Traffic APIs from February 2016 to March 2023. This dataset includes not just information on car accidents, but also meteorological information at the time of the event.

We have picked the relevant fields in order to extract weather information of several US cities from 2018 to 2022.

Fields	**Description**
City	List of the US cities
Weather_Timestamp	Timestamp of weather observation record
Temperature(F)	Temperature (Fahrenheit)
Humidity(%)	Humidity (%)
Pressure(in)	Air pressure (in)
Wind_speed	Wind speed (mph)
Weather_Condition	Weather condition

Table 2.4: Fields of US Accidents dataset

2.2.2. Dataset

- Training dataset

We merged two aforementioned separate Kaggle data sources to produce a training dataset for our model. This dataset combines meteorological data from cities in the United States, including several weather-related elements.   This data spans the years 2015 to 2022, providing a thorough seven-year historical weather dataset. This training dataset will include the following fields:

Fields	**Description**
City	List of US cities
Temp	Temperature (Kelvin)
Humidity	Humidity (%)
Pressure	Atmospheric pressure on the sea level (hPa)
Wind_speed	Maximum wind speed for the date specified in the request (m/s)
Weather_group	Weather condition group

Table 2.5: Fields of training dataset

- Unseen streaming dataset

As previously mentioned about the Weather API, we will utilize it to collect hourly meteorological data two days after called. This dataset will be integrated with our model to generate weather forecast predictions. It consists the following fields:

Fields	Description
DateTime	Timestamp of weather observation record
City	List of US cities
Temperature	Temperature (Kelvin)
Pressure	Atmospheric pressure on the sea level (hPa)
Humidity	Humidity (%)
Wind Speed	Maximum wind speed for the date specified in the request (metre/sec)

Table 2.6: Fields of unseen training dataset

2.3. Technology, Infrastructure, Models

2.3.1. Solution Design:

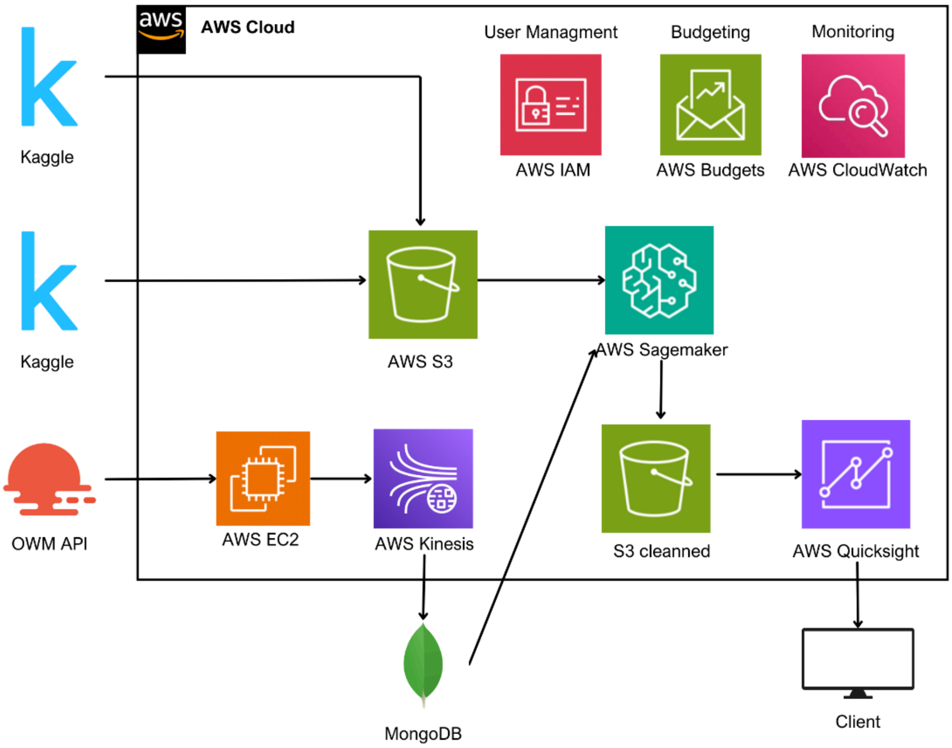


Figure 1: Solution Design for OpenWeather Predictions Big Data Project

We leverage a seamless integration of AWS services, combining the real-time capabilities of **Kinesis** and **MongoDB** with the analytical power of **SageMaker** and **QuickSight**, to deliver a comprehensive big data solution that efficiently processes both online and offline data.

Online Flow: Starting with the **OpenWeatherMap API,** we gather real-time weather predictions for the next two days. AWS Kinesis acts as our data buffer, streaming this information to an **EC2 instance**, where it's stored efficiently in MongoDB.

Offline Flow: On the offline side, we integrate two historical datasets collected from Kaggle. Both datasets converge in AWS S3, providing a scalable and centralized storage solution. Next, AWS SageMaker takes the stage for advanced analytics. It handles data cleaning, preprocessing, and supports machine learning model development.

The result is our cleaned and processed data, residing in a separate S3 bucket. For dynamic visualization and business intelligence, we turn to AWS Quick Sight. QuickSight connects to our cleaned data in S3, enabling us to create insightful dashboards and reports.

To ensure security and efficiency, we implement AWS IAM for user management. AWS Budgets help us control spending, and AWS CloudWatch monitors resource performance, triggering alerts for anomalies. Finally, clients can access these valuable insights through Quick Sight, providing a user-friendly interface to explore and understand the data.

2.3.2. Technology:

Amazon Kinesis, a component of Amazon Web Services (AWS), is a cloud-native platform specifically designed for the instantaneous handling of extensive streaming data. The Amazon Kinesis Streams platform is crucial because it effectively manages and distributes data streams, making it well-suited for real-time analytics and big data operations.

MongoDB is a prominent NoSQL database management system renowned for its adaptability and capacity to handle large amounts of data. Contrary to conventional relational databases, MongoDB lacks a fixed schema, allowing for the storage of flexible and adaptable data structures. With its ability to effectively manage unstructured and semi-structured data, MongoDB is widely used in many sectors. It is particularly favored for applications that deal with extensive datasets and dynamic data formats.

AWS S3, short for Amazon Simple Storage Service, is a highly scalable and secure cloud storage service provided by Amazon Web Services (AWS). It is designed to store and retrieve vast amounts of data. In our assignment, we use AWS S3 to store the raw data, the data after ETL, the data after preprocessing, streaming data and the prediction data.

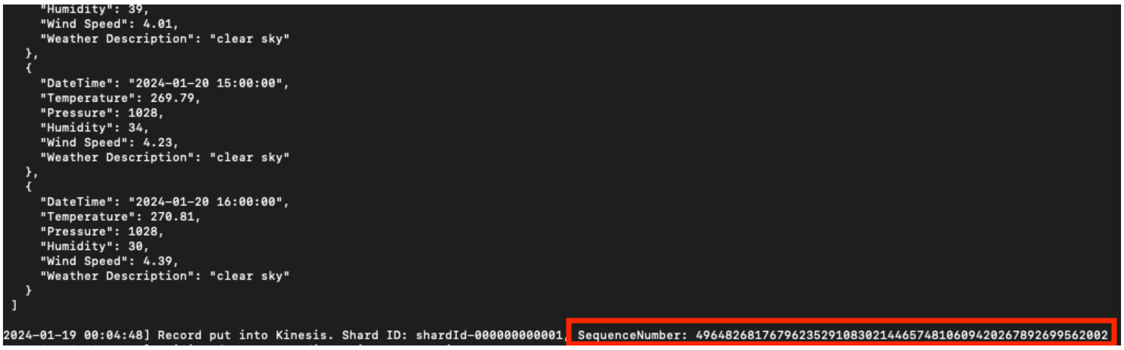
AWS SageMaker Notebook Instance is a cloud-based development environment provided by Amazon Web Services (AWS) for building, experimenting, and deploying machine learning models. It offers a fully managed Jupyter notebook experience, allowing users to write and execute code, visualize data, and collaborate on projects seamlessly. We used SageMaker Notebook Instance for ETL (extract, transform, load) process, data preprocessing, cleaning and model training process.

AWS QuickSight is a cloud-based business intelligence (BI) service provided by Amazon Web Services (AWS). It enables users to easily create interactive visualizations, dashboards, and reports. With QuickSight, our team has created an interactive dashboard with various visuals and filters to gain more insights to the datasets.

2.3.3.1 ETL

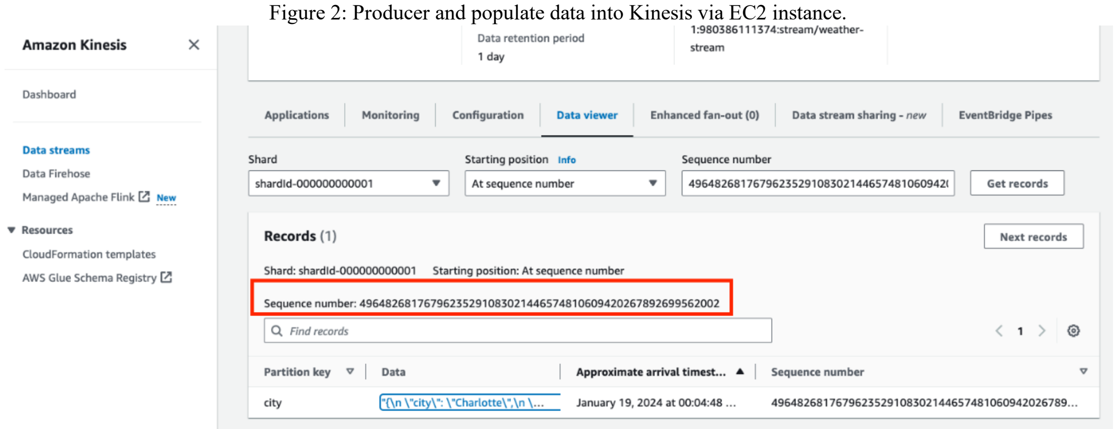
2.3.3.1.1 Streaming ETL

Within our online system, we begin the process by connecting with the OpenWeatherMap API to get accurate and up-to-date weather forecasts for the next 48 hours. The dynamic weather data is smoothly sent using AWS Kinesis, which functions as our specialized data buffer.



```
    "Humidity": 39,
    "Wind Speed": 4.01,
    "Weather Description": "clear sky"
  },
  {
    "DateTime": "2024-01-20 15:00:00",
    "Temperature": 269.79,
    "Pressure": 1028,
    "Humidity": 34,
    "Wind Speed": 4.23,
    "Weather Description": "clear sky"
  },
  {
    "DateTime": "2024-01-20 16:00:00",
    "Temperature": 270.81,
    "Pressure": 1028,
    "Humidity": 30,
    "Wind Speed": 4.39,
    "Weather Description": "clear sky"
  }
]
2024-01-19 00:04:48] Record put into Kinesis. Shard ID: shardId-000000000001, SequenceNumber: 49648268176796235291083021446574810609420267892699562002
```

Figure 2: Producer and populate data into Kinesis via EC2 instance.



Data retention period
1 day

1:980386111374:stream/weather-stream

Shard: shardId-000000000001

Starting position: At sequence number

Sequence number: 49648268176796235291083021446574810609420(

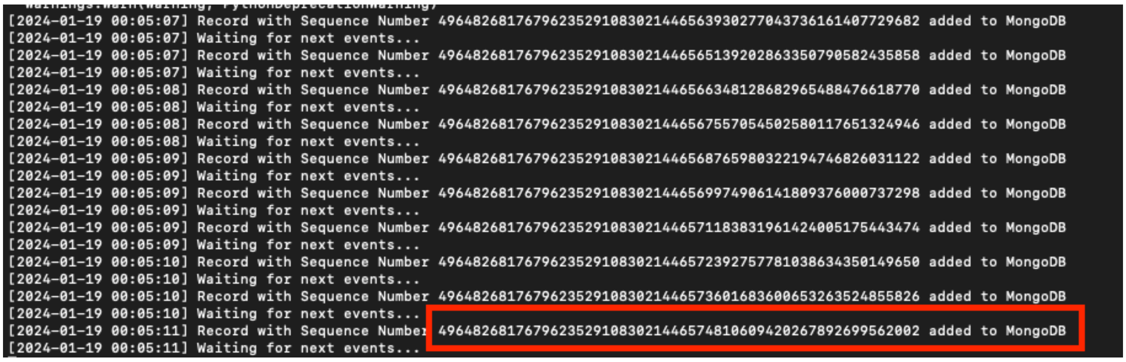
Records (1)

Shard: shardId-000000000001 Starting position: At sequence number

Sequence number: 49648268176796235291083021446574810609420267892699562002

Partition key	Data	Approximate arrival timest...	Sequence number
city	"{\n \"city\": \"Charlotte\",\n \...	January 19, 2024 at 00:04:48 ...	4964826817679623529108302144657481060942026789...

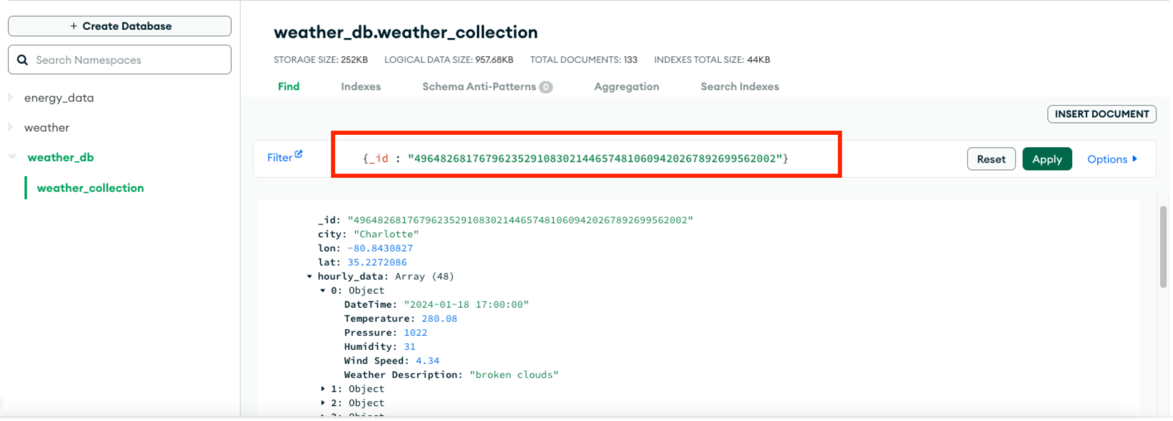
Figure 3: Check out the event having the latest sequenceNumber on Kinesis UI.



```
  warnings.warn(warning, PythonDeprecationWarning)
[2024-01-19 00:05:07] Record with Sequence Number 49648268176796235291083021446563930277043736161407729682 added to MongoDB
[2024-01-19 00:05:07] Waiting for next events...
[2024-01-19 00:05:07] Record with Sequence Number 49648268176796235291083021446565139202863350790582435858 added to MongoDB
[2024-01-19 00:05:07] Waiting for next events...
[2024-01-19 00:05:08] Record with Sequence Number 49648268176796235291083021446566348128682965488476618770 added to MongoDB
[2024-01-19 00:05:08] Waiting for next events...
[2024-01-19 00:05:08] Record with Sequence Number 49648268176796235291083021446567557054502580117651324946 added to MongoDB
[2024-01-19 00:05:08] Waiting for next events...
[2024-01-19 00:05:09] Record with Sequence Number 49648268176796235291083021446568765980322194746826031122 added to MongoDB
[2024-01-19 00:05:09] Waiting for next events...
[2024-01-19 00:05:09] Record with Sequence Number 49648268176796235291083021446569974906141809376000737298 added to MongoDB
[2024-01-19 00:05:09] Waiting for next events...
[2024-01-19 00:05:09] Record with Sequence Number 49648268176796235291083021446571183831961424005175443474 added to MongoDB
[2024-01-19 00:05:09] Waiting for next events...
[2024-01-19 00:05:10] Record with Sequence Number 49648268176796235291083021446572392757781038634350149650 added to MongoDB
[2024-01-19 00:05:10] Waiting for next events...
[2024-01-19 00:05:10] Record with Sequence Number 49648268176796235291083021446573601683600653263524855826 added to MongoDB
[2024-01-19 00:05:10] Waiting for next events...
[2024-01-19 00:05:11] Record with Sequence Number 49648268176796235291083021446574810609420267892699562002 added to MongoDB
[2024-01-19 00:05:11] Waiting for next events...
```

Figure 4: Consumer runs to consume the event stored on Kinesis and added to MongoDB

AWS Kinesis effectively handles the uninterrupted flow of data, guaranteeing its seamless transfer to an EC2 instance. Afterwards, the real-time weather data is saved in MongoDB, a NoSQL database renowned for its flexibility in handling changing and dynamic information. The EC2 instance acts as a mediator, enabling the efficient and streamlined storing of weather forecasts in MongoDB.



weather_db.weather_collection

STORAGE SIZE: 252KB LOGICAL DATA SIZE: 957.68KB TOTAL DOCUMENTS: 133 INDEXES TOTAL SIZE: 44KB

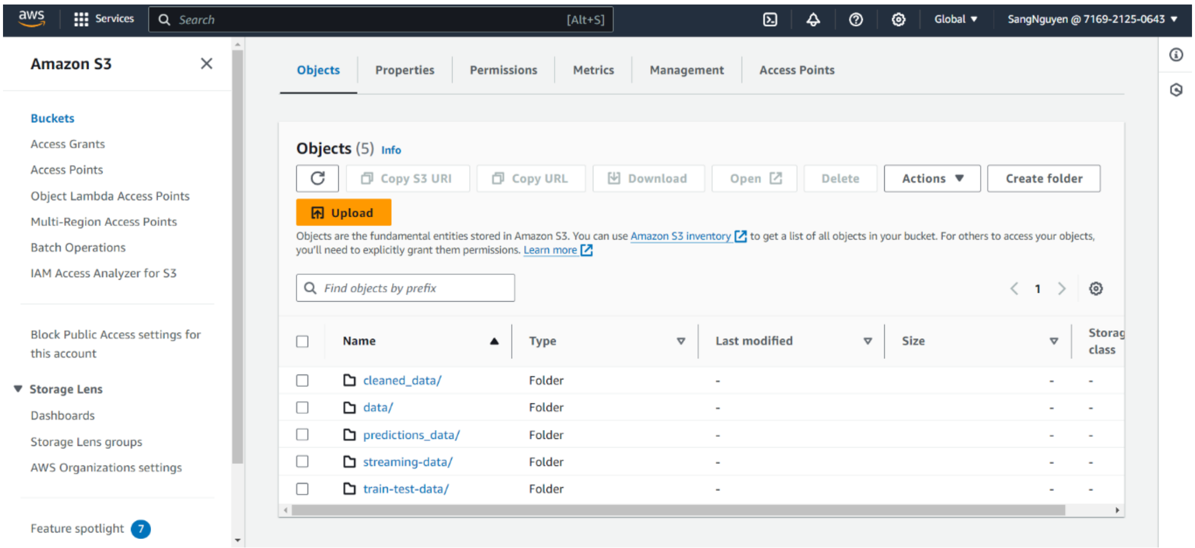
```
{_id : "49648268176796235291083021446574810609420267892699562002"}
```

```
_id: "49648268176796235291083021446574810609420267892699562002"
city: "Charlotte"
lon: -80.8430827
lat: 35.2272086
hourly_data: Array (48)
  0: Object
    DateTime: "2024-01-18 17:00:00"
    Temperature: 280.08
    Pressure: 1022
    Humidity: 31
    Wind Speed: 4.34
    Weather Description: "broken clouds"
  1: Object
  2: Object
```

Figure 5: Searching the event having "_id" similar to the latest sequenceNumber on Kinesis

This fast procedure guarantees the rapid collection of real-time weather information from the OpenWeatherMap API. The data is then buffered, processed, and saved in MongoDB for future analysis and use in our full big data solution.

2.3.3.1.2 Offline ETL



Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

	Name	Type	Last modified	Size	Storage class
□	cleaned_data/	Folder	-	-	-
□	data/	Folder	-	-	-
□	predictions_data/	Folder	-	-	-
□	streaming-data/	Folder	-	-	-
□	train-test-data/	Folder	-	-	-

Figure 6: AWS S3 Buckets

As mentioned earlier in our offline flow, we focused on converging the two historical datasets from Kaggle into AWS S3, specifically into the "data/" bucket. However, to prepare and create a training dataset for model training process in AWS Sagemaker, we built an ETL process that can effectively clean and transform these two datasets to make sure they are concatenated into a large and cohesive dataset. Then, this dataset will be loaded back to AWS S3 and stored in “cleanned_data/” bucket. Some of the key steps in this ETL process include:

Dataset	Step	Description
Historical Hourly Weather Data 2012-2017	1	Importing all necessary data files from Kaggle data source from “data/” AWS S3 bucket
	2	Filtering the data according to a specified list of US cities.
	3	Reshaping each data file and then concatenating them into one single data frame
	4	Dropping unnecessary columns and renaming the remaining ones
US Accidents (2016 - 2023)	5	Importing all necessary data files from Kaggle data source from “data/” AWS S3 bucket
	6	Extracting necessary meteorological columns and renaming them to match with the first one.
	7	Converting time column data type to datetime and doing filtering to extract data later than 2018
	8	Filtering the data according to a specified list of US cities.
	9	Converting units to ensure consistency across the two datasets
	10	Drop unnecessary columns
Training dataset	11	Concatenating two datasets to create training dataset
	12	Filtering out minority values that fall below a predefined threshold.
	13	Categorizing weather conditions into different groups of weather based on specific data descriptions
	14	Selecting the data ranging from the years 2015 to 2022
	15	Loading the data back to AWS S3 “cleaned_data/” bucket for model training in AWS SageMaker

Table 3: Offline Flow ETL Process

2.3.3.2 Preprocessing + Cleaning:

In order to be well-prepared for data modelling tasks, data processing and data cleaning are the necessary steps in order to handle all the data errors. Initially, there are some rows with missing values and our solution is to eliminate them as it just accounts for an insignificant fraction of the whole dataset, which is 3.2%. Then, it is compulsory to check duplicated rows as it may be a potential reason leading to data skewing. Therefore, all of them were dropped and the number of samples decreased to nearly 530,00 compared to 868,841 rows at the beginning. Although a great amount of data was lost, it prevented our model from biased predictions in further tasks. In addition, it is recognized that the target variable, which is “Weather_Group” feature, consists of some relatively identical values and should be grouped into one to simplify the classification problem. In this case, “cloudy” and “overcast” can be taken as vivid examples. Despite their cloud coverage being different, they still imply cloudy weather in general. As a result, they will be integrated into one weather condition named “Cloudy”. After performing the grouping task, the number of values in “Weather_Group” column dropped from six to four in total.

After that, as there are some categorical features that have not been converted into numerical format yet, Label Encoder will be applied to handle this issue. Its approach is to assign a distinct integer to each category of a feature. One important point to mention is to create new features related to month, date and hour. As this is a time-series dataset, extracting time features can be extremely important to capture the underlying seasonality and long-term trends in the data. Eventually, with the aim to select the most informative variables for training models, our decision is to use feature importance ranking based on Random Forest algorithm. However, it reveals that there is no feature that is highly important compared to others. Specifically, the scores fairly range from 0.075 to 0.175, which are quite poor. Therefore, all of the features are chosen to use in data modelling. Moreover, as our dataset is imbalanced, SMOTE (Synthetic Minority Oversampling Technique) is the solution to create new instances and help to improve the performance.

2.3.3.3. Model chosen

For the models we choose to train the datasets, we have chosen 2 tree-based models:

- Random Forest Regression
- Extra Tree Regression

Unlike linear Machine Learning models such as SVM (Support Vector Machines or Logistic Regression), tree-based models can handle datasets with missing values and outliers. They are less sensitive to outliers compared to some other models because the tree structure allows them to create separate branches to accommodate data points. Additionally, tree-based doesn’t require the data to be normalized, as they based on comparisons of feature values at each split point. In our training data, there are many skewed columns and outliers and by using tree-based models will help us save time and computational resources.

2.4. Estimated Running Cost

We will list out the resources that we intend to use:

Services	Resources	Time	Cost	Monthly cost
AWS S3	S3 standard storage	2 weeks	0$	0.02$
AWS SageMaker	Tier instance: ml.m5.xlarge Time: 140hrs	140hrs	0$	32.20$
AWS Kinesis Data Streams	Shards per month: 2	2 weeks	0$	21.90$
AWS EC2	Tier: t2.micro	2 weeks	0$	4.23$
AWS QuickSight	QuickSight standard account	30 days	0$	5$
Sum				**63.35$**

Table 4.1: Services and Resources used for development

Actual cost

Services	Actual cost
AWS S3	0.01$
AWS SageMaker	53.30$
AWS EC2	**3.20$**
AWS Kinesis	**4.30$**
AWS QuickSight	**0$**
Sum	**60.81$**

Table 4.2: Actual cost for development

F2.5. Preliminary Results

To clearly understand the results of our models, the metrics used to evaluate them will be introduced initially. As our problem is multi-class classification, f1 score was chosen to compare and analyze model's performance. It is known that f1 score is the harmonic of recall and precision, hence the concepts of those must be presented. To be more specific, precision is a measure presenting the proportion of true positives to the total positive predictions from the model. In our case, taking rainy predictions as an example, precision score shows the number of correct predictions out of all the rainy ones. Its formula is described as follow [9]:

$$precision = \frac{true\,positives}{true\,positives + false\,positives}$$

In term of recall, it is used to calculate how many true positives our model successfully predicted. Additionally, it is also known as true positive rate, and it is calculated like below [9]:

$$recall = \frac{true\,positives}{true\,positives + false\,negatives}$$

It is known that there is always a trade-off between precision and recall. Therefore, f1 score is used to measure how good the model makes that trade-off. One remarkable point to mention is that this metric significantly penalizes extreme negative values of its components as if one of them is equal to 0, f1 score also becomes 0.

$$F1 = \frac{2 \times precision \times recall}{precision + recall}$$

In our case, the specific measurement is f1 weighted as our dataset is imbalanced. Particularly, it represents a weighted average of class-wise F1 score, whose weights are calculated by the number of samples in that class. [10]. By observing the formulas, it is visible that this metric is likely to favor the majority classes.

$$Weighted\,F1 = \sum_{i=1}^{N} w_i \times F1\,score_i \quad \text{where, } w_i = \frac{No.\,of\,samples\,in\,class\,i}{Total\,number\,of\,samples}$$

With the aim to forecast weather conditions in the near future, there are four models built, namely default Random Forest, Random Forest with hyperparameter tuning, default Extra Trees and Extra Trees with hyperparameter tuning. Also, for handling the imbalanced dataset, SMOTE (Synthetic Minority Oversampling Technique) is also applied to create new instances. The table below will show the outcomes of each model:

Model	F1 score
Random Forest	0.68
Random Forest with hyperparameter tuning	0.68

Extra Trees	0.7
Extra Trees with hyperparameter tuning	0.7

Table 5: F1 score for trained model

As indicated from the above table, the scores of all models range fairly from 0.68 to 0.7, which are relatively moderate results. The highest score is achieved by Extra Trees and Extra Trees with hyperparameter tuning at 0.7 compared to the lowest score at 0.68 of Random Forest. In terms of time complexity, it is recognized that Extra Tree’s running time is faster than Random Forest since node splits are random. Therefore, Extra Trees with hyperparameter tuning is decided to be our final model as with specific parameters, it is less prone to overfitting.

2.6. Visualization dashboard

Please follow the instructions below to access the dashboard. Please note that our dashboard may become inaccessible after 12 February because it is the end of the free trial period of AWS Quicksight of the root account. If you have trouble accessing the dashboard, please contact us.

Instructions:

1. Log in IAM account with this link: https://716921250643.signin.aws.amazon.com/console
2. Input the following field:

IAM username: AthurTang
Password: Rmit12345

3. Open a new tab: copy and run this link: https://us-east-1.quicksight.aws.amazon.com/sn/dashboards/9f0b2e54-0d4f-4cf2-a5f7-194d0ca5160e/views/c2634a51-f896-4ac4-9517-9409e7d5252c?directory_alias=asm3-bigdata

Note: it won’t charge you. We set IAM account for you to interact with the dashboard

AWS QuickSight is a cloud-based business intelligence (BI) and data visualization tool offered by Amazon Web Services (AWS). It enables organizations to easily analyze, visualize, and derive insights from their data, empowering data-driven decision-making. Users can connect to various data sources, including AWS services like Amazon Redshift, Amazon S3, Amazon Athena, as well as on-premises databases, and third-party sources.

Our dashboard has 2 main parts: historical weather summary and weather forecast data.

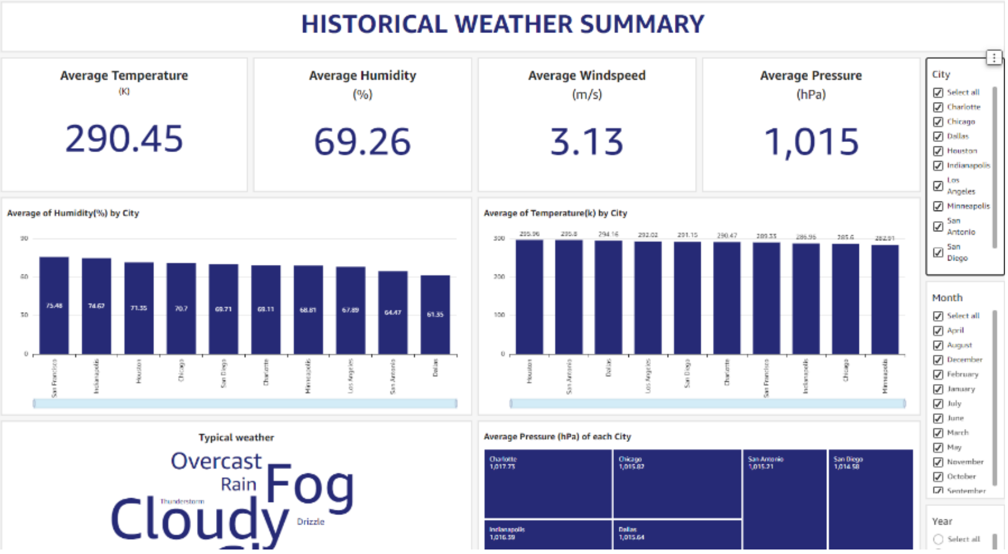
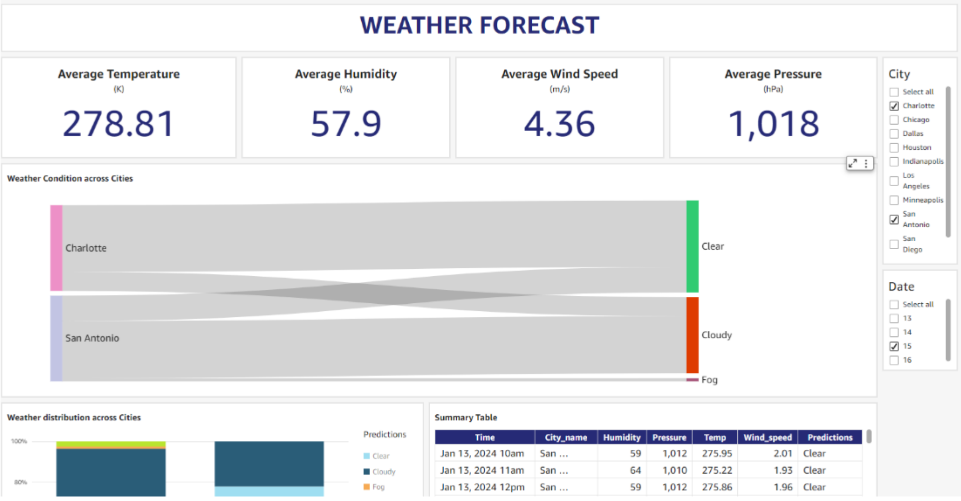


Figure 7: Historical weather summary



WEATHER FORECAST

Average Temperature (K): 278.81

Average Humidity (%): 57.9

Average Wind Speed (m/s): 4.36

Average Pressure (hPa): 1,018

City
☐ Select all
☑ Charlotte
☐ Chicago
☐ Dallas
☐ Houston
☐ Indianapolis
☐ Los Angeles
☐ Minneapolis
☑ San Antonio
☐ San Diego

Date
☐ Select all
☐ 13
☐ 14
☑ 15
☐ 16

Summary Table

Time	City_name	Humidity	Pressure	Temp	Wind_speed	Predictions
Jan 13, 2024 10am	San ...	59	1,012	275.95	2.01	Clear
Jan 13, 2024 11am	San ...	64	1,010	275.22	1.93	Clear
Jan 13, 2024 12pm	San ...	59	1,012	275.86	1.96	Clear

Figure 8: Weather forecast data

3.1. Project Summary

The project focused on training multilabel classification weather forecast model, building a pipeline to forecast weather of live data from API, and creating visualization dashboard to view information. It achieved commendable outcome of all components working without errors and acceptable accuracy. The following are achievements for each component.

1. Multilabel classification weather forecast model:
   The model gets input of past weather data from two cleaned data sources and predicts categories of weather such as rain, fog, cloud status. The weighted F1 score of the selected model was 0.71, which is a decent performance considering input data.
2. Pipeline implementation
   The pipeline has three parts: training model based on data saved on S3 and retrieving API data for prediction of weather. Training model is mainly done on AWS cloud system, which uses data stored on S3 and worked on SageMaker and saves result on cleaned S3. Retrieving API data uses API and use AWS EC2, Kinesis, and MongoDB to save data and send it to SageMaker to process testing. There were some conflicts and errors at the start of development, but the problems were all solved and succeeded until the last step, visualization dashboard implementation.
3. Visualization dashboard
   Visualization dashboard is divided into two parts: Historical weather data and prediction weather data. Both can view weather data based on filter from viewer as year and city. Readability and providing useful information were considered most along with user friendly design.

3.2. Limitation of the Solution

There are some limitations to our project. The first one is the limited number of features available. While advanced weather forecasting systems collect data from numerous sources with diverse weather features such as satellite data, aircraft observation, weather balloons, and more, our project model was trained using a smaller set of features, including temperature, humidity, pressure, and wind information. This limitation is due to free and public sources that we used, which don't offer as much diversity as some listed options. Despite the limitations, the performance was still acceptable.

Our second limitation is the lack of resources. Initially, our dataset exceeds more than one million records. Due to resource constraints, we had to scale it down to more than 800000 rows. Moreover, we attempted to utilize the AWS cloud environments, which provided just enough resources. However, considering real-life weather forecast systems that are developed on supercomputers [g] and handle significantly larger datasets than ours, it's clear that our resources were quite limited

3.3. Future Work

There can be a few points development can be made in the future. First, a better dataset can be used. The combination of data with good quality can be used for training data. If there is more information over basic weather information, better model performance is expected. Second, models can predict more weather information. Currently, models only do classification the status of weather. However, additional predictions can be made such as precipitation rate, UV index, and visibility range. Extra information related to agriculture can be provided for farmers, the main target users of the forecast. Third, more cities may be benefited of forecast. The current model only provides forecasts for some cities in the US, mainly California. Data of more cities and countries can be added to provide service for more users.

4. Reference

[1] The World Bank, "What is food security?," World Bank, https://www.worldbank.org/en/topic/agriculture/brief/food-security-update/what-is-food-security (accessed Jan. 19, 2024).

[2] USDA, "Global Food Security," Nation Institute of Food and Agriculture, https://www.nifa.usda.gov/topics/global-food-security (accessed Jan. 19, 2024).

[3] M. Eilts, "The role of weather-and weather forecasting-in agriculture," DTN, https://www.dtn.com/the-role-of-weather-and-weather-forecasting-in-agriculture/ (accessed Jan. 19, 2024).

[4] E. Weise, "Weird weather hit cattle ranchers and citrus growers in 2022. why it likely will get worse.," Yahoo! News, https://news.yahoo.com/beef-shortage-looms-florida-citrus-100040103.html (accessed Jan. 19, 2024).

[5] EPA, "US EPA," Climate Impacts on Agriculture and Food Supply | Climate Change Impacts | US EPA, https://climatechange.chicago.gov/climate-impacts/climate-impacts-agriculture-and-food-supply (accessed Jan. 19, 2024).

[6] V. Bisset, "The U.N. warns 'an era of global boiling' has started. What does that mean?," The Washington Post, https://www.washingtonpost.com/climate-environment/2023/07/29/un-what-is-global-boiling/ (accessed Jan. 19, 2024).

[7] NOAA, "National Weather Service," NOAA, https://www.weather.gov/media/mkx/general/noaa-nws-who-we-are.pdf (accessed Jan. 19, 2024).

[8] P. Bajpai, "WHO produces the world's food," Investopedia, https://www.investopedia.com/articles/investing/043015/who-produces-worlds-food.asp#:~:text=United%20States&text=And%20while%20agriculture%20contributes%20only,twelfth%2Dlargest%20producer%20of%20rice. (accessed Jan. 19, 2024).

[9] T. Tigerschiold, "What is Accuracy, Precision, Recall and F1 Score?," Labelf.ai, Nov. 17, 2022. https://www.labelf.ai/blog/what-is-accuracy-precision-recall-and-f1-score (accessed Jan. 18, 2024).

[10] R. Kundu, "F1 Score in Machine Learning: Intro & Calculation," V7labs.com, 2023. https://www.v7labs.com/blog/f1-score-guide (accessed Jan. 18, 2024).

[11] N. US Department of Commerce, "About models," National Weather Service, https://www.weather.gov/about/models#:~:text=Modern%20models%20are%20able%20to,stations%20can%20all%20be%20ingested. (accessed Jan. 19, 2024).

[12] N. US Department of Commerce, "About supercomputers," National Weather Service, https://www.weather.gov/about/supercomputers#:~:text=Currently%2C%20the%20combined%20processing%20power,Virginia%2C%20and%20Orlando%2C%20Florida. (accessed Jan. 19, 2024).

5. Appendix

Declaration of Project Team Member

Team 2

Ful Name	Student ID	Contribution to the project (%)	Contributions of the project
Huynh Van Anh	s3836320	20%	Design Pipeline, manage Kinesis usage
Nguyen Le Bao Han	s3894290	20%	Cleaning & Examining data, development of model
Nguyen Thanh Sang	s3878340	20%	Managing training dataset, development of model, building visualization dashboard
Lee Wonjun	s3773920	20%	Cleaning data, manage presentation and report
Nguyen Dang Nhat	s3878292	20%	Managing training dataset, development of model, building visualization dashboard, ETL process, AWS management

I/we state that all information provided in this form is true and correct.

Group member’s name & signature	Date
Huynh Van Anh Anh	19 Jan 2024
Nguyen Le Bao Han Han	19 Jan 2024
Nguyen Thanh Sang Sang	19 Jan 2024
Lee Wonjun	19 Jan 2024
Nguyen Dang Nhat Nhat	19 Jan 2024