1. Introduction

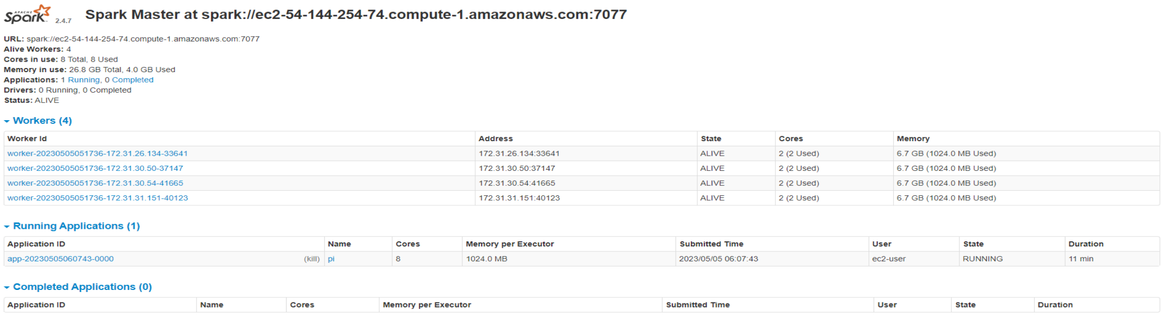
The NASA Prediction of Worldwide Energy Resources (POWER) [1] dataset provides valuable information on global energy patterns and environmental factors. However, extracting meaningful insights from the dataset is a challenging task due to its complexity and volume. The project aims to address this challenge by utilizing tools and techniques to analyze the POWER dataset. Specifically, analysis was centered around factors related to weather forecasting. To overcome the complexity of the dataset, machine learning models were used to identify patterns and trends in the data. Additionally, utilization of data visualization techniques was used to provide a deeper understanding of the data. The insights gained from the analysis of the POWER dataset can inform policy decisions related to energy production and consumption, guide investment strategies for renewable energy projects, and support research into new energy technologies. Sharing the results and insights with the research community will contribute to a better understanding of global energy patterns and support the development of sustainable energy solutions for the future.

2. Methodology

To investigate the interplay among various environmental factors, the present study will commence by establishing a PySpark[2] cluster and a Jupyter[3] working environment. Subsequently, the data will be subjected to data cleaning and dimensionality reduction techniques utilizing PySpark. The correlation between the variables will then be analyzed through the implementation of data visualization. Ultimately, a linear regression model will be generated, employing the remaining variables in the dataset, to predict the average daily temperature.

2.1. PySpark Clustering

To leverage Spark's clustering capabilities, we deployed Spark on AWS EC2 instances, capitalizing on the scalability offered by cloud computing. The clustering setup process was automated with Flintrock[4] scripts. Initially, the AWS credential configuration was written to the "*~/.aws*" directory, followed by the execution of the "$ flintrock configure" command to generate the Flintrock configuration file. Next, the Flintrock configuration was customized to include the desired number of slave nodes, AWS security group, and the SSH key-pair for the instances. In consideration of cost efficiency, we opted to begin with four slave nodes and t3.large instance type. The cluster can be scaled up using the Flintrock command "*$ flintrock add-slaves cluster --num-slaves {slave_number}*".



Spark Master at spark://ec2-54-144-254-74.compute-1.amazonaws.com:7077

URL: spark://ec2-54-144-254-74.compute-1.amazonaws.com:7077
Alive Workers: 4
Cores in use: 8 Total, 8 Used
Memory in use: 26.8 GB Total, 4.0 GB Used
Applications: 1 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers (4)

Worker Id	Address	State	Cores	Memory
worker-20230505051736-172.31.26.134-33641	172.31.26.134:33641	ALIVE	2 (2 Used)	6.7 GB (1024.0 MB Used)
worker-20230505051736-172.31.30.50-37147	172.31.30.50:37147	ALIVE	2 (2 Used)	6.7 GB (1024.0 MB Used)
worker-20230505051736-172.31.30.54-41665	172.31.30.54:41665	ALIVE	2 (2 Used)	6.7 GB (1024.0 MB Used)
worker-20230505051736-172.31.31.151-40123	172.31.31.151:40123	ALIVE	2 (2 Used)	6.7 GB (1024.0 MB Used)

Running Applications (1)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration
app-20230505060743-0000 (kill)	pi	8	1024.0 MB	2023/05/05 06:07:43	ec2-user	RUNNING	11 min

Completed Applications (0)

Application ID	Name	Cores	Memory per Executor	Submitted Time	User	State	Duration

Figure 1: Display of running EC2 instances.

The Python environment was configured on the master node of the cluster using pip for package installation. To ensure remote accessibility, a Jupyter Notebook server was established and connected via HTTP port 8888. Subsequently, the 'findspark' package was invoked at the start of the notebook to configure PySpark, thereby enabling the Python kernel to locate the path to the Spark installation directory By providing the option of using either the EC2 Spark cluster or Databricks, group members can choose the platform that best suits their needs and preferences.

2.2. Data Processing

The NASA POWER dataset contains a large amount of data both in the precise specification of location by longitude and latitude coordinates and in the number of parameters available that describe those locations. As such, eight parameters that are most likely related to weather forecasting were chosen to describe 50 cities from 1981 to 2021. The table below shows the parameters obtained from the dataset.

Parameter	Description
`T10M`	Temperature at 10 meters
`CLOUD_AMT`	Cloud amount
`QV10M`	Specific humidity at 10 meters
`PW`	Precipitable water
`PS`	Surface pressure
`GLOBAL_ILLUMINANCE`	Global horizontal illuminance from direct and diffuse radiation
`WS10M`	Wind speed at 10 meters
`EVLAND`	Land evaporation

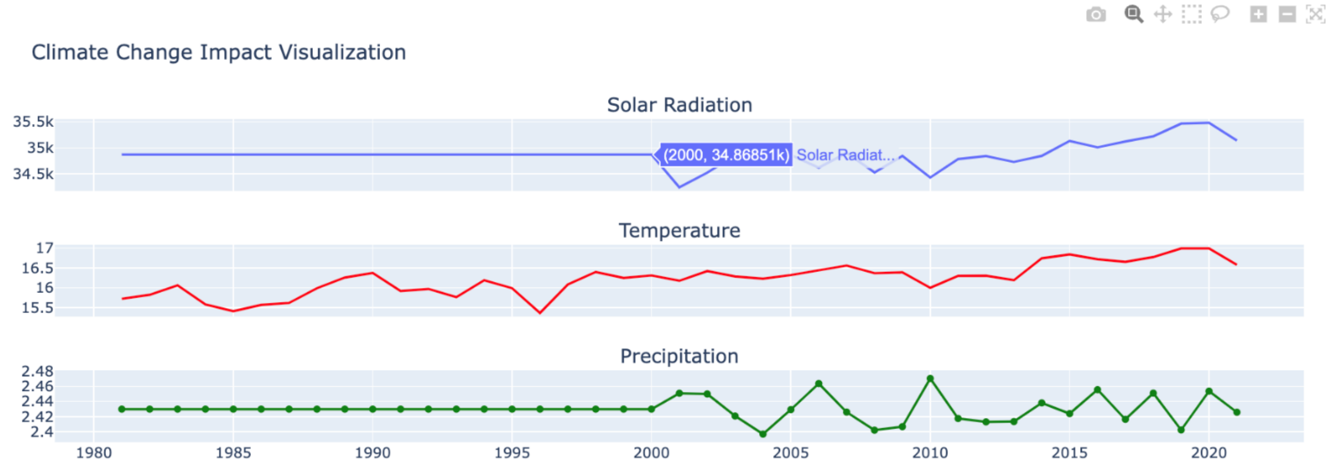
Data for each city was collected through an API request and was arranged into a PySpark data frame. Additionally, as the dataset's API only accepts longitude and latitude coordinates, the `geopy`[6] package was used to obtain geographical coordinates from city names. All PySpark data frames, which contain the parameter values for each city, were merged, creating a conclusive data frame with 748,750 rows and 13 columns. Lastly, before exporting the data to a CSV file, data cleaning was performed to sort the rows of the data frame by `YEAR` and `DOY` attributes and to replace values of -999.0 with the attribute mean.

2.3 Dimensionality Reduction (PCA)

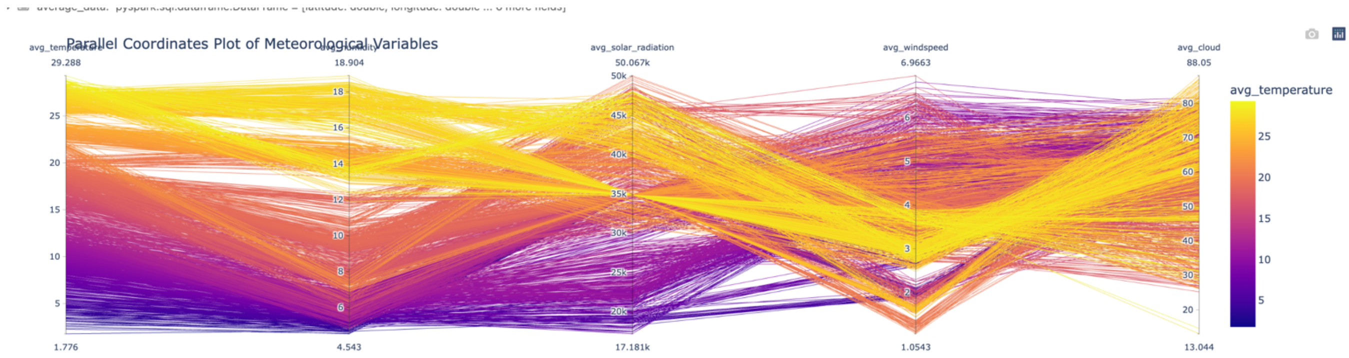
To mitigate the effects of noise and to obtain uncorrelated features that can be utilized for linear regression, it is necessary to perform dimensionality reduction on the dataset, given the high number of features present. To execute principal component analysis (PCA) within the PySpark framework, it is imperative to transform the data from a PySpark data frame into vectorized columns utilizing the VectorAssembler function. This procedure is carried out by applying the PySpark PCA function on the vector columns, which yields the PCA elements (with a specified k-value of 3). Finally, the resulting PCA is integrated back into the original data frame.

2.4. Data Visualization

The objective of the data visualization exercise was to discern significant patterns and interrelationships among variables of interest. The visualization tasks were divided into three segments, namely: (1) analyzing data from 1981 to 2021 to compare the impact of climate change on solar radiation, temperature, and precipitation; (2) constructing parallel coordinates to carefully examine the relationships between various categories; and (3) generating heatmaps to visualize data pertaining to global temperature, humidity, and solar radiation.

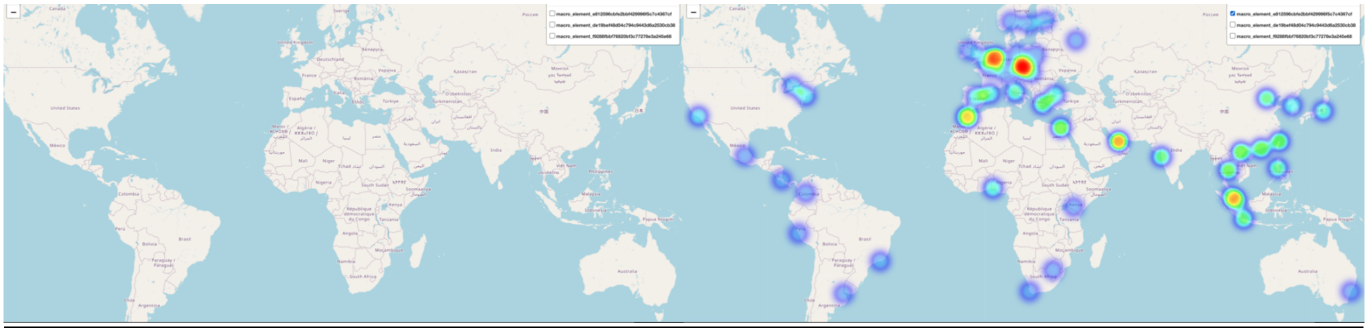


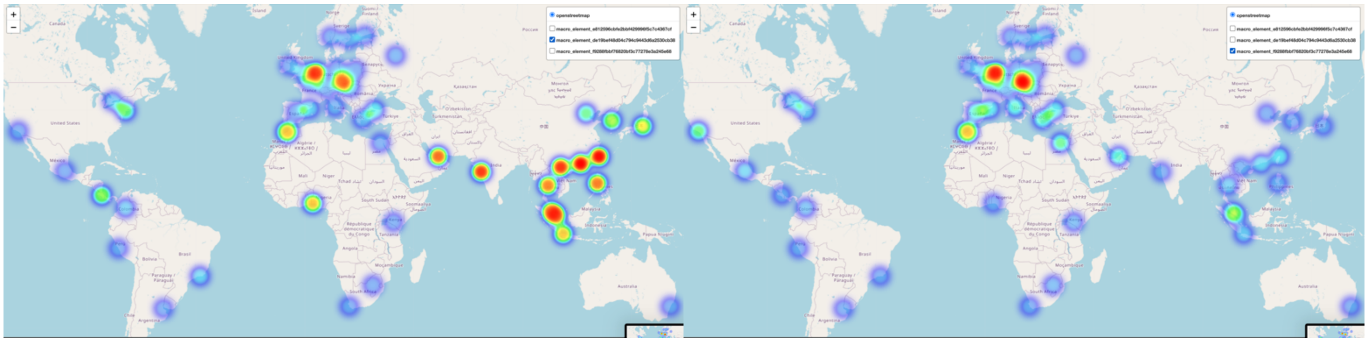
The initial image displays changes in yearly average solar radiation, temperature, and precipitation. As solar radiation and precipitation records are unavailable before 2001, the mean value of solar radiation has been calculated up to this point using column averages to replace null values. The observed increase in solar radiation and temperature, which exceed the environmental effects of a 1-degree Celsius rise, may suggest persistent climate change. Conversely, precipitation appears relatively stable with minor fluctuations, but additional research is necessary given the limited data available until 2001.



```
average_data: pyspark.sql.dataframe.DataFrame = [latitude: double, longitude: double ... 6 more fields]
```

This parallel coordinate plot visualizes all feature variables, including temperature, humidity, solar radiation, wind speed, and cloud cover. To enhance correlations, temperature is indicated by the line color. Upon closer examination, the plot reveals a direct relationship between temperature and solar radiation, as well as a moderate association between temperature and humidity. However, no correlation exists between temperature and wind speed. Moreover, the notion that cloud cover and humidity are directly linked, which could impact precipitation levels, is refuted since cloud cover does not exhibit any correlation with other variables, contrary to expectations.





The final visualization comprises heat maps representing local temperatures, humidity levels, and solar radiation for the month of May. Technical challenges in the folium library necessitated the creation of twelve separate visualizations, one for each month, as adding a time frame to the heatmap was not feasible. The strata are arranged from top to bottom based on temperature, humidity, and solar radiation, despite the absence of layer names. The heatmap on the right-hand side confirms the strong correlation between temperature and solar radiation, while humidity appears consistent across various regions, except for certain areas such as Southeast Asia and the Middle East. Furthermore, the visualization confirms a direct relationship between temperature and humidity and highlights regional variations.

2.5. Linear Regression

Our analysis employed linear regression, a commonly used statistical modeling technique, to predict city temperature based on influential factors such as humidity, cloud cover, and day of the year. Through the utilization of linear and multi-regression analysis, our objective was to assess the models' ability to forecast temperature, utilizing both the available data and data processed through PCA functions.

To perform this analysis, we divided the data into training and testing sets, allocating 70% for training and 30% for testing. In the subsequent sections, we will provide a comprehensive explanation of each component of the regression test function, accompanied by a thorough analysis of the obtained results.

Baseline

The baseline code played a pivotal role in our analysis, serving as a fundamental point of reference for predicting city temperature. It served as a benchmark for evaluating the performance of more intricate models and assessing the efficacy of advanced approaches. Through the establishment of this baseline, we were able to gauge the improvements achieved by incorporating additional features and employing sophisticated modeling techniques, thus providing a valuable reference for comparing outcomes.

The baseline code utilized a single data point, namely 'T10M', to predict temperature. Its function was to establish a simplistic approach that could serve as a benchmark for subsequent models. Initial predictions were obtained by utilizing basic statistical measures such as the mean or median of the dependent variable. The baseline code served as a starting point for assessing the performance improvements achieved by more sophisticated models.

In summary, the baseline code played a critical role in our analysis, providing a foundation for evaluating and comparing the outcomes of our more complex models.

Linear Regression

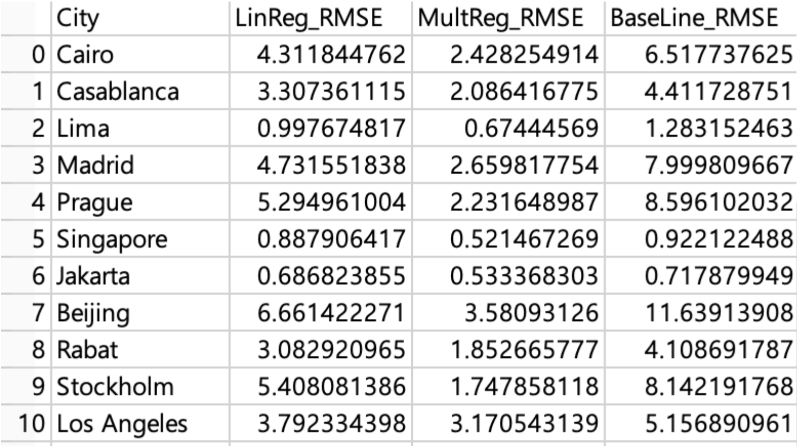
The linear regression code builds upon the baseline code by incorporating a wider range of features. Specifically, three processed data points from the previous section, namely 'PCA1,' 'PCA2,' and 'PCA3,' were utilized as independent variables. By fitting a linear regression model to this expanded dataset, our aim was to capture the underlying patterns and relationships between the features and city performance metrics. The model generated predictions based on these patterns, allowing us to evaluate its accuracy and performance.

Multi-regression

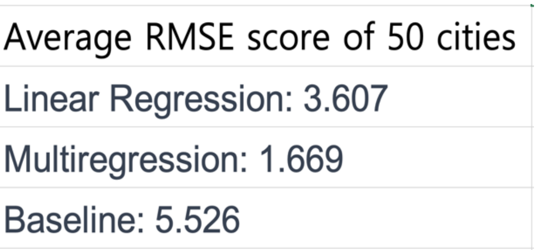
The multi-regression code built upon the linear regression code by incorporating a comprehensive set of variables. In addition to the 'PCA1,' 'PCA2,' and 'PCA3' features, we included 'DOY,' 'CLOUD_AMT,' 'QV10M,' 'PW,' 'PS,' 'GLOBAL_ILLUMINANCE,' 'WS10M,' and 'EVLAND' as independent variables. This more extensive dataset allowed us to capture a wider range of factors that could potentially influence city performance metrics. By employing multi-regression analysis, our aim was to explore the combined effects of these variables and their impact on prediction accuracy.

Results

We utilized RMSE (Root Mean Squared Error) as a metric to evaluate their accuracy.



	City	LinReg_RMSE	MultReg_RMSE	BaseLine_RMSE
0	Cairo	4.311844762	2.428254914	6.517737625
1	Casablanca	3.307361115	2.086416775	4.411728751
2	Lima	0.997674817	0.67444569	1.283152463
3	Madrid	4.731551838	2.659817754	7.999809667
4	Prague	5.294961004	2.231648987	8.596102032
5	Singapore	0.887906417	0.521467269	0.922122488
6	Jakarta	0.686823855	0.533368303	0.717879949
7	Beijing	6.661422271	3.58093126	11.63913908
8	Rabat	3.082920965	1.852665777	4.108691787
9	Stockholm	5.408081386	1.747858118	8.142191768
10	Los Angeles	3.792334398	3.170543139	5.156890961



Average RMSE score of 50 cities

Linear Regression: 3.607

Multiregression: 1.669

Baseline: 5.526

The table presented above provides a clear indication of the insights gained from our analysis of the linear regression, multi-regression, and baseline models for predicting city temperature. Our findings indicate that the multi-regression approach was the most effective model, achieving an impressively low RMSE of 1.669. This superior performance can be attributed to the inclusion of additional variables, such as PCA components, day of the year, cloud cover, and other relevant factors. The incorporation of these variables contributed to improved temperature predictions. These results underscore the importance of considering multiple factors when constructing models and forecasting city temperature. By utilizing the multi-regression technique and incorporating a diverse range of variables, we successfully enhanced the accuracy of temperature predictions.

3. Conclusion

This project analyzed global energy patterns and environmental factors using the NASA POWER dataset. PySpark cluster on AWS EC2 instances, machine learning models, data visualization, and statistical modeling were used to extract insights. The analysis identified patterns and trends among environmental factors such as solar radiation, temperature, precipitation, humidity, cloud cover, wind speed, and day of the year. Machine learning models predicted city temperature. Overall, this project demonstrated the power of PySpark, machine learning, data visualization, and statistical modeling for complex datasets.

References

[1] S. Stackhouse Jr., D. R. Westberg, and T. Zhang, "NASA's Prediction of Worldwide Energy Resource Web Services," IEEE Transactions on Geoscience and Remote Sensing, vol. 51, no. 3, pp. 1257-1265, Mar. 2013.

[2] M. Zaharia, et al., "Apache Spark: A Unified Engine for Big Data Processing," in Proceedings of the 1st International Conference on Big Data Computing and Communications, Beijing, China, 2015, pp. 1-6.

[3] A. Smith, "Data Analysis with Python," Jupyter Notebook, GitHub, 2022. [Online]. Available: https://github.com/asmith123/data-analysis-notebook. [Accessed: May 21, 2023].

[4] N. Chammas, "Flintrock: A Command-Line Tool for Apache Spark Clusters," GitHub, Apr. 2021. [Online]. Available: https://github.com/nchammas/flintrock. [Accessed: May 21, 2023].

[5] J. R. Kugelman, "FindSpark: Simplifying Apache Spark Cluster Deployment," GitHub, Jun. 2016. [Online]. Available: https://github.com/minrk/findspark. [Accessed: May 21, 2023].

[6] GeoPy contributors, "GeoPy Documentation," GeoPy, n.d. [Online]. Available: https://geopy.readthedocs.io/en/stable/. [Accessed: May 21, 2023].