Iteration 4 BDAS (Steps 1 – 8)

The assignment follows a sequence of steps that is a synthesis of the Cross-Industry Standard Process for Data Mining (CRISP-DM) process (SPSS, 2007) and the KDD process (Fayyad et al., 1996).

GitHub link: https://github.com/YurajK00/722-Iteration-4.git

1. Situation Understanding

There is overwhelming scientific consensus that the Earth's climate is changing, primarily due to human activities that increase the concentration of greenhouse gases (GHGs) in the atmosphere. GHGs such as carbon dioxide (CO2), methane (CH4), and nitrous oxide (N2O) trap heat in the Earth's atmosphere, leading to a warming effect known as the greenhouse effect. Global temperature records show that the Earth's average surface temperature has been rising steadily over the past century, with the most pronounced increases observed in recent decades. The burning of fossil fuels for energy production, transportation, and industrial processes is the largest source of anthropogenic CO2 emissions, contributing significantly to global warming. Deforestation and land-use changes release stored carbon into the atmosphere and reduce the Earth's capacity to absorb CO2 through photosynthesis, exacerbating the greenhouse effect. Industrial activities such as cement production, chemical manufacturing, and waste management release CO2 and other GHGs, further intensifying global warming. In recent years, particularly in 2015 and 2016, we have experienced unprecedented global temperatures. Each month of 2016 set a new global heat record and witnessed the largest increase in atmospheric CO2 concentrations to date. Millions of people worldwide are suffering from the impacts of climate change. For instance, the drought driven by El Niño, intensified by climate change, is worsening food insecurity, highlighting a significant gap in climate adaptation and disaster preparedness. Additionally, the number of people displaced from their homes due to extreme weather events has reached unprecedented levels, representing the most severe humanitarian crisis since World War II.

1.1 Situation objectives

Climate change exacerbates habitat loss, fragmentation, and degradation, leading to shifts in species distributions, extinction risks, and loss of ecosystem services. Addressing the problem of climate change and rising global temperatures requires concerted efforts at the global, national, and local levels, with a focus on mitigation, adaptation, resilience-building, and sustainable development. To address this situation some of the objectives would include:

• Analyze global temperature records to understand long-term trends, variability, and recent changes in average surface temperatures.
• Evaluate climate model projections and scenarios to forecast future changes in temperature, precipitation, sea-level rise, and extreme weather events.

1.2 Assessment of the Situation

While it is relatively easy to understand and assess climate change, it remains crucial to take action and tackle the situation effectively. Resources The situation will be assessed by compiling data collected from various global and national climate organization repositories into a single dataset. Machine learning models like Linear Regression will play a crucial role in handling predictions and providing insights into the situation. Basic programming languages like Python will be used as a framework to implement the model and perform exploratory data analysis with visual representations as well. Requirements The basic requirement would include data with some historical trends and relevant facts that could be helpful in extrapolation and impact temperature changes over the years. All while the above is true there are some potential risks and contingencies predicting future global temperatures and climate change based on historical data comes with several risks and uncertainties

1.2.1 Data Limitations:

Using historical data up to 2013 may not capture all relevant factors influencing climate change. There may be new variables or trends emerging after 2013 that could significantly impact temperature patterns.

1.2.2 Model Uncertainty: Forecasting future temperatures relies on the assumption that historical patterns will continue. However, climate is a complex system influenced by numerous factors, and future trends may not follow past patterns.

1.2.3 External Factors: External events such as volcanic eruptions, changes in solar radiation, or policy decisions affecting greenhouse gas emissions can have unforeseen impacts on global temperatures.

1.2.4. Feedback Loops: Climate change can trigger feedback loops that amplify warming, such as melting ice leading to reduced reflectivity (albedo effect) or thawing permafrost releasing additional greenhouse gases. Predicting the timing and magnitude of these feedback loops is challenging.

1.2.5 Contingency Measures:

1. Scenario Analysis: Considering multiple scenarios with different assumptions about future trends and external factors can potentially help assess the range of possible outcomes and their associated uncertainties.

2. Sensitivity Analysis: By testing the sensitivity of the model to changes in key variables or assumptions it can help identify which factors have the greatest impact on the predictions of the model and evaluate their uncertainty.

3. Expert Consultation: Although how vague it may sound consulting with climate scientists or domain experts to gain insights into emerging trends, potential feedback mechanisms, and areas of uncertainty in climate models can help minimizing the risks and be crucial for a robust model development

4. Communication of Uncertainty: Clearly communicating the limitations and uncertainties associated with the predictions could provide stakeholders with a nuanced understanding of the potential risks and the range of possible outcomes.

5. Adaptation Strategies: Finally, developing adaptive strategies that can be implemented in response to different climate change scenarios. This may include infrastructure improvements, land-use planning, and policy interventions to mitigate the impacts of climate change.

1.3 Data Mining Goals

1. Pattern Recognition: Identifying recurring patterns or associations within the data that can provide valuable insights or predictive capabilities.

2. Anomaly Detection: Detecting unusual or unexpected patterns in the data that may indicate errors, anomalies, or other significant events.

3. Prediction and Forecasting: Developing models to predict future trends or outcomes based on historical data, such as forecasting future temperatures or climate change patterns.

1.4 Project Plan

Project Objective: To analyze historical climate data and identify trends and patterns to understand the impact of climate change.

Duration: 1 week (Friday, May 15 2024 - Friday, 24 May, 2024)

Phase 1: Project Setup and Preparation

Day 1 - Activities:

- Gather relevant information, determine situation for the project, understand the situation.
- Define project scope, objectives, and success criteria.
- Revise lab exercises and course material. Refine Python skills and Spark.
- Develop initial project timeline and milestones.

Phase 2: Data Collection and Exploration

Day 2 –

- Identify data sources for climate data (e.g., NOAA, NASA, Berkley Earth). - Collect historical climate data for analysis.
- Explore the collected data to understand its structure, format, and quality.
- Clean the data by addressing missing values, outliers, and inconsistencies.
- Perform preliminary data visualization to identify patterns and trends.

Phase 3: Data Preparation and Feature Engineering

Day 3 –

Select relevant features for analysis based on project objectives.
- Engineer new features if necessary (e.g., calculate monthly or yearly averages). - Integrate data from different sources if applicable.
- Normalize or standardize the data to ensure consistency.
- Split the data into training, validation, and test sets.

Phase 4: Model Development and Evaluation

Day 4 Activities:

- Select appropriate data mining algorithms for analysis (e.g., linear regression, time series analysis).

- Develop predictive models using the training data.
- Evaluate model performance using validation data and adjust parameters as needed.
- Fine-tune models to improve accuracy and generalization.
- Validate models using the test data set and assess their performance.

Phase 5: Results Interpretation and Documentation

Day 5 -Activities:

- Interpret the results of the data mining analysis.
- Document key findings, insights, and recommendations.
- Prepare visualizations and reports to communicate the results to stakeholders.
- Review and finalize the project documentation.

Phase 6: Project Review and Iteration

Day 6 Activities:

- Conduct a project review meeting to assess the outcomes and lessons learned.
- Identify areas for improvement and potential future iterations.
- Update project documentation and knowledge repository.
- Plan for future iterations or follow-up projects based on the findings.

2.1 Collecting Data

Initial raw data was collected from non – profit Berkely Earth with reports on reports on how land and ocean temperature vary by location (https://berkeleyearth.org/data/). The high-resolution Berkeley Earth data set has been used to construct new country, regional and local summaries. However, for this Iteration only temperatures with respect to the countries have been chosen. The Berkeley Earth Surface Temperature Study combines 1.6 billion temperature reports from 16 pre-existing archives. It is nicely packaged and allows for slicing into interesting subsets (for example by country). They publish the source data and the code for the transformations they applied. They also use methods that allow weather observations from shorter time series to be included, meaning fewer observations need to be thrown away. However, the Final Dataset used is a compilation of three Datasets of global temperatures over sea and land and temperatures by regions and cities. (https://berkeleyearth.org/temperature-region/southern-hemisphere) (https://berkeleyearth.org/temperature-region/northern-hemisphere)

2.2 Data Description

The format of the data is in csv form. It is a compiled dataset of three different datasets in a single csv.
The data contains 159695 rows with 465370 as entries with totaling of 4 columns.

```
<class 'pandas.core.frame.DataFrame'>
Index: 465370 entries, 1274 to 577461
Data columns (total 4 columns):
 #   Column                         Non-Null Count   Dtype
---  ------                         --------------   -----
 0   dt                             465370 non-null  datetime64[ns]
 1   AverageTemperature             465370 non-null  float64
 2   AverageTemperatureUncertainty  465370 non-null  float64
 3   Country                        465370 non-null  object
dtypes: datetime64[ns](1), float64(2), object(1)
memory usage: 17.8+ MB
```

Fig 1

Features and Datatype:

Features	Description	Format
Date	Range of date from 1743 to 2013	YY-MM-DD Datetime Format
LandAverageTemperature	global average land temperature in Celsius	Continuous Doubletype format
LandAverageTemperatureUncertainty	the 95% confidence interval around the average	Continuous Doubletype format
Country	List of over 243 countries from Åland to Zimbabwe	Categorical String datatype

2.3 Data Exploration:

The three datasets is visually very large to show in a tabular form. Thus, the first and thus only the top 20 rows are rows have been presented.

- The first datset titled "GlobalTemperatures.csv" contains historical temperature records from various regions around the globe. Below is a detailed description of the columns in the dataset:

1. dt: This column represents the date of the temperature recording in the format YYYY-MM-DD.
2. LandAverageTemperature: This column indicates the average land temperature recorded for the given date.
3. LandAverageTemperatureUncertainty: This column represents the 95% confidence interval around the land average temperature, reflecting the uncertainty in the measurement.
4. LandMaxTemperature: This column, although containing missing values in the initial rows, is supposed to represent the maximum land temperature recorded for the given date.
5. LandMaxTemperatureUncertainty: This column indicates the uncertainty around the maximum land temperature.
6. LandMinTemperature: This column, also containing missing values in the initial rows, is intended to represent the minimum land temperature recorded for the given date.
7. LandMinTemperatureUncertainty: This column shows the uncertainty around the minimum land temperature.
8. LandAndOceanAverageTemperature: This column is supposed to represent the combined average temperature of both land and ocean areas for the given date, though it contains missing values in the initial rows.
9. LandAndOceanAverageTemperatureUncertainty: This column indicates the uncertainty around the combined land and ocean average temperature.

This dataset spans several centuries, starting from the year 1750 and there are missing values in several columns, particularly in the temperature measurements for maximum, minimum, and combined land and ocean temperatures.

The primary focus of the initial data seems to be on land average temperatures and their uncertainties.

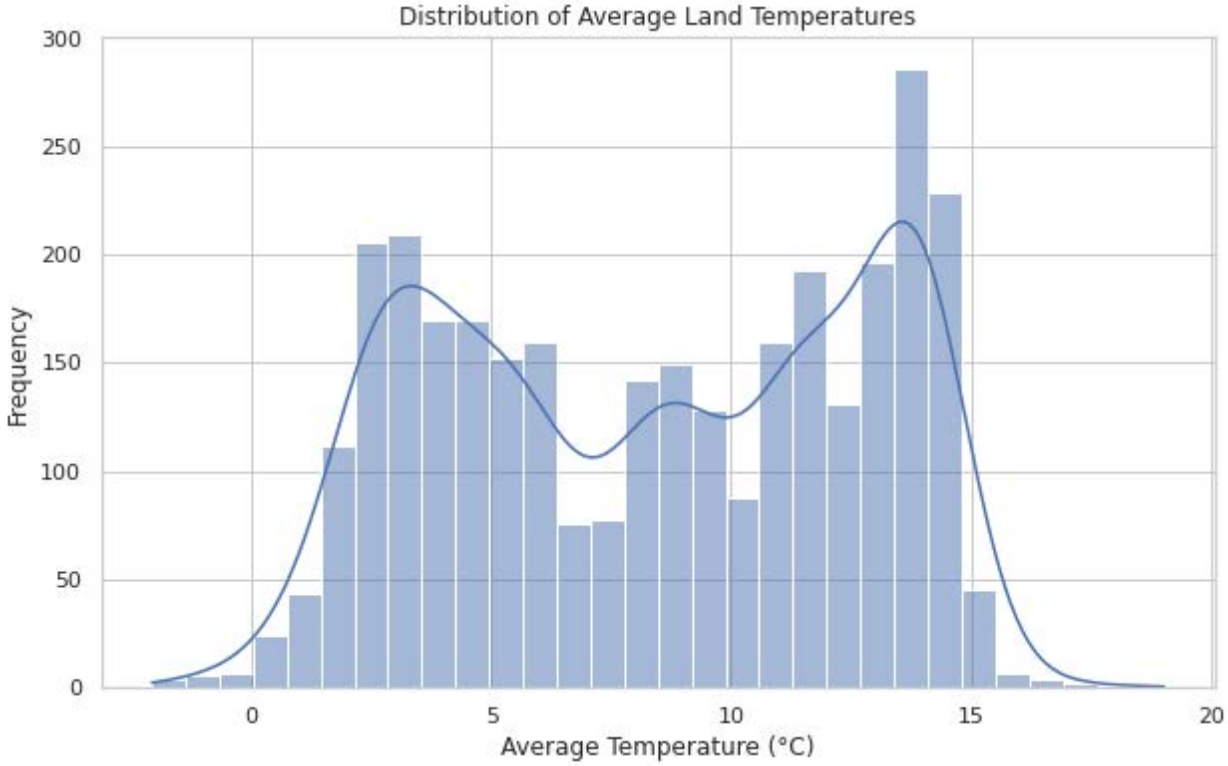


Fig 2 (a)

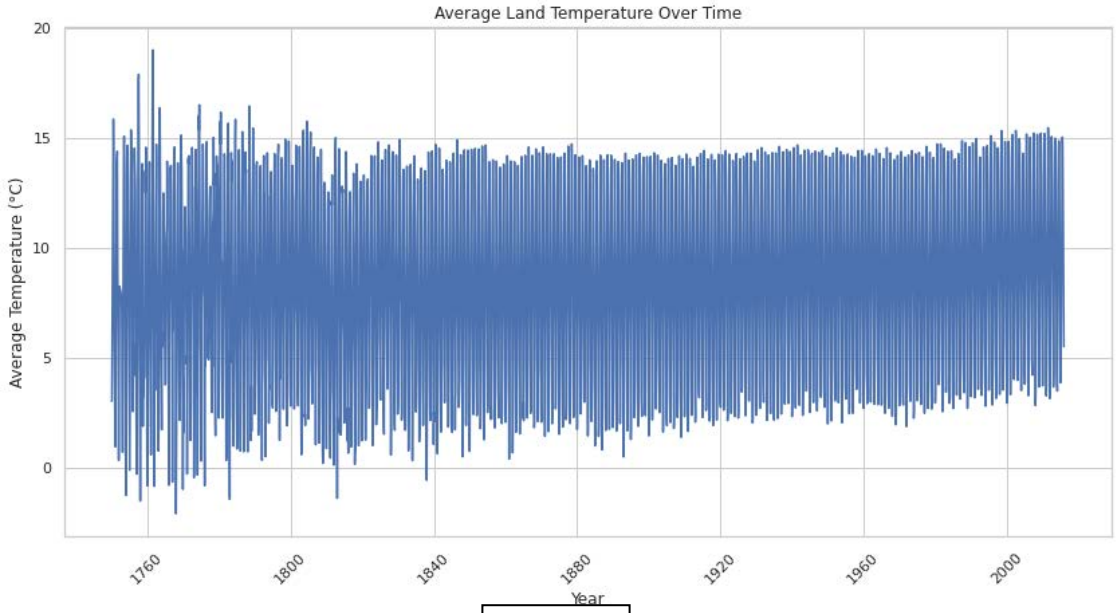


Fig 2 (b)

- The second dataset is virtually very large to upload on the jupyter notebook with size amounting to about 502 mb. It's a dataset appears to contain historical temperature records for the major cities across the globe and the country they represent to.
  Below is a detailed description of the columns in the dataset:

  1. dt: This column represents the date of the temperature recording in the format YYYY-MM-DD.
  2. AverageTemperature: This column indicates the average temperature recorded on the given date. There are some missing values in this column.
  3. AverageTemperatureUncertainty: This column represents the uncertainty in the average temperature measurement, expressed as a confidence interval.
  4. City: This column specifies the city where the temperature was recorded, which is Århus in Denmark.
  5. Country: This column indicates the country where the city is located, which is Denmark.
  6. Latitude: This column shows the latitude coordinates of the city, indicating its position north of the equator.
  7. Longitude: This column shows the longitude coordinates of the city, indicating its position east of the prime meridian.

	dt	AverageTe	AverageTe	City	Country	Latitude	Longitude
2	1743-11-0	6.068	1.737	Ã…rhus	Denmark	57.05N	10.33E
3	1743-12-01			Ã…rhus	Denmark	57.05N	10.33E
4	1744-01-01			Ã…rhus	Denmark	57.05N	10.33E
5	1744-02-01			Ã…rhus	Denmark	57.05N	10.33E
6	1744-03-01			Ã…rhus	Denmark	57.05N	10.33E
7	1744-04-0	5.788	3.624	Ã…rhus	Denmark	57.05N	10.33E
8	1744-05-0	10.644	1.283	Ã…rhus	Denmark	57.05N	10.33E
9	1744-06-0	14.051	1.347	Ã…rhus	Denmark	57.05N	10.33E
10	1744-07-0	16.082	1.396	Ã…rhus	Denmark	57.05N	10.33E
11	1744-08-01			Ã…rhus	Denmark	57.05N	10.33E
12	1744-09-0	12.781	1.454	Ã…rhus	Denmark	57.05N	10.33E
13	1744-10-0	7.95	1.63	Ã…rhus	Denmark	57.05N	10.33E
14	1744-11-0	4.639	1.302	Ã…rhus	Denmark	57.05N	10.33E
15	1744-12-0	0.122	1.756	Ã…rhus	Denmark	57.05N	10.33E
16	1745-01-0	-1.333	1.642	Ã…rhus	Denmark	57.05N	10.33E
17	1745-02-0	-2.732	1.358	Ã…rhus	Denmark	57.05N	10.33E
18	1745-03-0	0.129	1.088	Ã…rhus	Denmark	57.05N	10.33E
19	1745-04-0	4.042	1.138	Ã…rhus	Denmark	57.05N	10.33E
20	1745-05-01			Ã…rhus	Denmark	57.05N	10.33E

Fig 2 (c)

The third dataset titled GlobalTemperatures by Country which is the dataset in consideration for the whole iteration is a compiled dataset of the first two dataset.

3. Data Preparation

The data was combined from two distinct datasets, "GlobalTemperatures" and "GlobalTemperaturesByCity," to create a comprehensive "GlobalTemperaturesByCountry" dataset. The data preparation process involved several key steps, ensuring the data was selected, cleaned, constructed, integrated, and reformatted effectively.

3.1 Data Selection

The primary goal was to create a unified dataset that captures global temperature trends by country, incorporating detailed temperature records from individual cities and general global averages.

Both the datasets were evaluated for completeness, accuracy, and relevance. The "GlobalTemperatures" dataset provided broad global temperature trends, while the "GlobalTemperaturesByCity" dataset offered granular temperature data specific to various cities.

The were some technical constraints which were taken into considerations which included the handling of large datasets, ensuring efficient processing and storage, and compatibility with analytical tools like PySpark and Pandas.

3.2 Data Cleaning

Both datasets contained missing values and inconsistent date formats. The city dataset also had special character issues in the city names (e.g., "Ã…rhus" instead of "Århus").

Resolution of Issues:

Missing Values: Missing temperature values were filled with the mean temperature of the respective dataset to maintain continuity.

Date Formatting: Dates were converted to a consistent datetime format.

Special Characters: City names with special characters were corrected to their proper forms.

To deal with missing values a graph was plotted with ***Count of Missing values*** against ***Year*** to determine which period signified the most missing or NA values to further try reduction in unnecessary dataset. Transformation To further mitigate the process, **Interpolation method** was used which is the process of filling in missing values in a DataFrame or Series by estimating values based on the surrounding data points. The inbuilt library of python called pandas provides several methods for interpolation, which allows us to choose the most appropriate method based on the

data and requirements. For this dataset linear method was used.

```
#It seems lot of the data seems to be missing prior the year 1870. To generalise filtering out the data prior to 1850
data_filtered = df[df['dt'].dt.year >= 1850]

# Check the missing data count after filtering
data_filtered.head(), data_filtered.isnull().sum()
```

Img 1

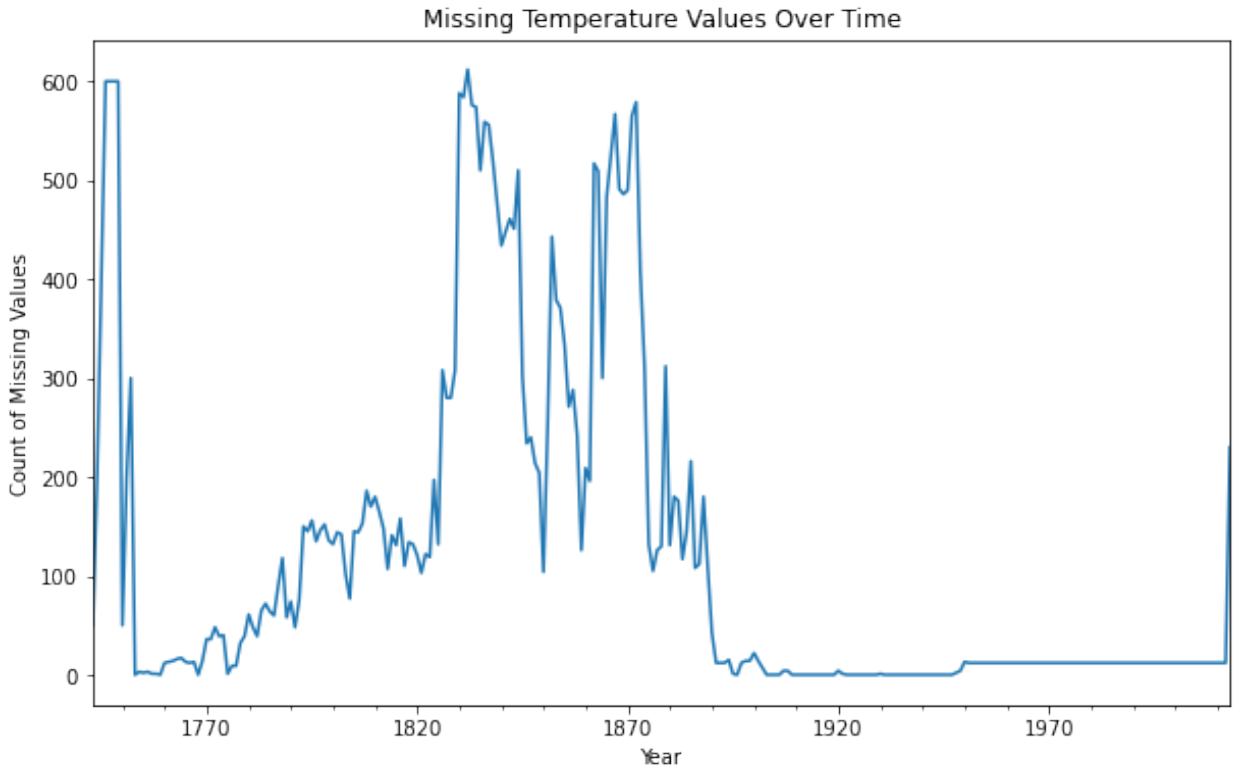


Fig 3

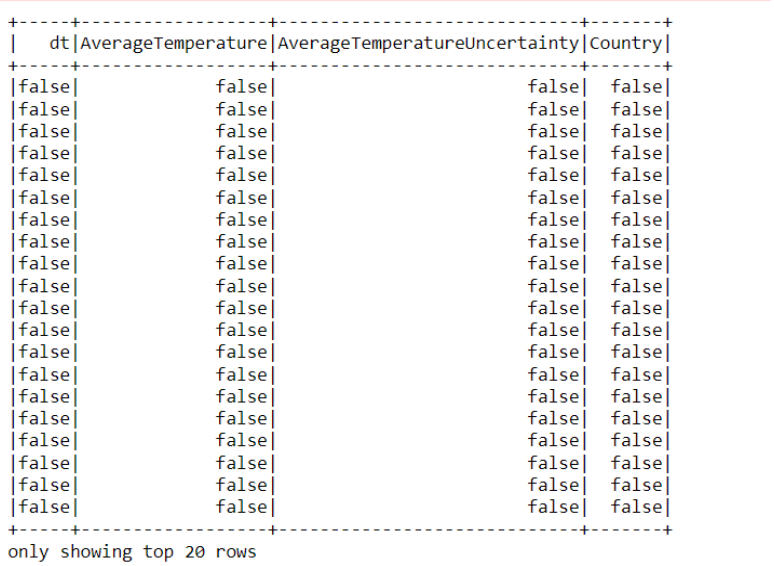
```
# Filter data for years >= 1850
filtered_df = df.filter(year('dt') >= 1870)

# Handle missing values using interpolation (Note: PySpark does not support direct interpolation, so convert to Pandas)
pandas_filtered_df = filtered_df.toPandas()
pandas_filtered_df['AverageTemperature'] = pandas_filtered_df['AverageTemperature'].interpolate(method='linear')
pandas_filtered_df['AverageTemperatureUncertainty'] = pandas_filtered_df['AverageTemperatureUncertainty'].interpolate(method='l

# Convert back to Spark DataFrame
filtered_df = spark.createDataFrame(pandas_filtered_df)

# Verify missing values are handled
filtered_df.select([col(c).isNull().alias(c) for c in filtered_df.columns]).show()
```

Img 2



dt	AverageTemperature	AverageTemperatureUncertainty	Country
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false
false	false	false	false

only showing top 20 rows

Fig 4

3.3 Data Construction

Feature Creation: New features were created to enhance the analysis. For instance, we derived country-level average temperatures from city-level data.

Data Repositories/Tables: The cleaned datasets were stored in new tables, facilitating easier access and manipulation for subsequent analysis.

3.4 Data Integration

Merging Datasets: The integration process involved merging the "GlobalTemperatures" and "GlobalTemperaturesByCity" datasets. This was achieved by matching records on the date and geographical identifiers (latitude and longitude for cities and country names).

Alignment: Ensured that city-level data were appropriately aggregated to the country level before merging with the global dataset.

Consistency: Maintained consistency in measurement units and naming conventions across both datasets.

3.5 Data Reformatting

Standardizing Formats: Reformatted all temperature values to a consistent scale (Celsius) and ensured that date values were uniformly represented.

Trimming Content: Removed extraneous columns and records that were not relevant to the country-level analysis, streamlining the dataset for focused analysis.

4. Data transformation

4.1 Data Reduction

Upon examining the **Figure 3**, it becomes evident that the highest level of inconsistencies occurs within the time span from 1770 to 1870, leading to the drawn conclusion. Additionally, missing values have been identified and documented using numerical representation, with a total of 32,651 missing values for the Average Temperature feature and 31,912 for the Average Temperature Uncertainty feature. Based on the observations from the graph, a conclusion was reached indicating that a significant number of missing values persist prior to the year 1850. Consequently, the data was filtered accordingly using the method ***interpolation***, resulting in a reduction of nearly 7,000 value counts. Subsequently, a reassessment was conducted to verify the presence of any remaining inconsistencies. The count of missing values was reduced by almost 2500.

```
#It seems lot of the data seems to be missing prior the year 1870. To generalise filtering out the data prior to 1850
data_filtered = df[df['dt'].dt.year >= 1850]

# Check the missing data count after filtering
data_filtered.head(), data_filtered.isnull().sum()
```

Img 3

```
(            dt  AverageTemperature  AverageTemperatureUncertainty Country
 1274 1850-01-01              -9.083                          1.834   Åland
 1275 1850-02-01              -2.309                          1.603   Åland
 1276 1850-03-01              -4.801                          3.033   Åland
 1277 1850-04-01               1.242                          2.008   Åland
 1278 1850-05-01               7.920                          0.881   Åland,
 dt                                0
 AverageTemperature            12912
 AverageTemperatureUncertainty    12173
 Country                           0
 dtype: int64)
```

Fig 5

4.2 Data Projection

In order to gain deeper insights into temperature trends over time, a line plot was created to compare global average temperature trends with those of New Zealand. This involved grouping the variable "global_avg_temp" by the "AverageTemperature" category, and a new variable "nz_avg_temperature" was similarly grouped by the category "New Zealand." To facilitate side-by side comparison of the results, a tight layout was generated.

```
# Group by year and calculate the global average temperature
global_avg_temp = filtered_df.groupBy(year('dt').alias('Year')).avg('AverageTemperature')

# Filter data for New Zealand and calculate yearly averages
nz_avg_temp = filtered_df.filter(col('Country') == 'New Zealand').groupBy(year('dt').alias('Year')).avg('AverageTemperature')

# Convert to Pandas for plotting
global_avg_temp_pd = global_avg_temp.toPandas()
nz_avg_temp_pd = nz_avg_temp.toPandas()

# Plotting global and New Zealand temperature trends
fig, ax = plt.subplots(1, 2, figsize=(16, 6))

ax[0].plot(global_avg_temp_pd['Year'], global_avg_temp_pd['avg(AverageTemperature)'], label='Global Average Temperature', color='
ax[0].set_title('Global Average Temperature Over Time')
ax[0].set_xlabel('Year')
ax[0].set_ylabel('Average Temperature (°C)')
ax[0].legend()
ax[0].grid(True)

ax[1].plot(nz_avg_temp_pd['Year'], nz_avg_temp_pd['avg(AverageTemperature)'], label='New Zealand', color='blue')
ax[1].set_title('Temperature Trends in New Zealand')
ax[1].set_xlabel('Year')
ax[1].set_ylabel('Average Temperature (°C)')
ax[1].legend()
ax[1].grid(True)

plt.tight_layout()
plt.show()
```

Img 4

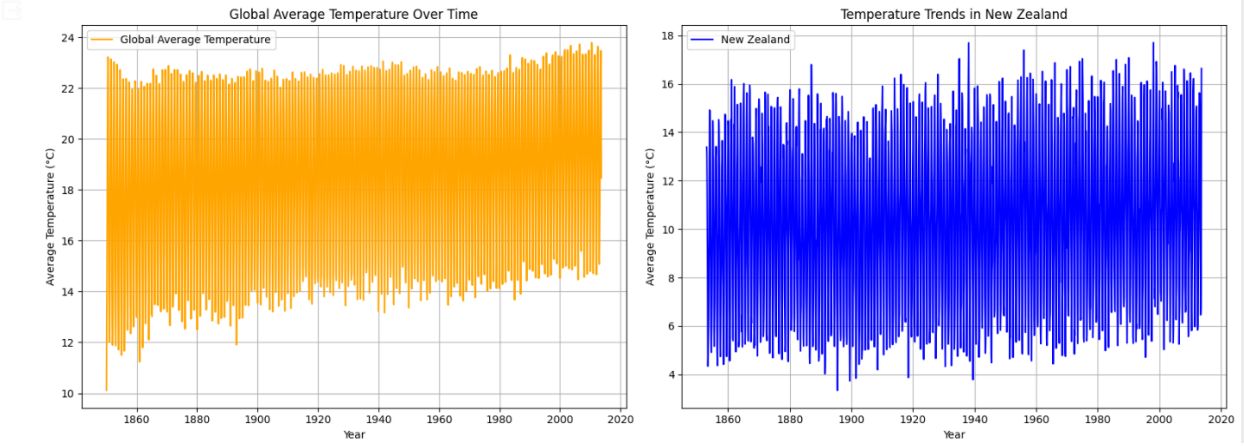


Fig 6

5. Data – Mining Method

In this analysis, we utilized several data mining (DM) methods to achieve specific data mining objectives. The primary objectives included data cleaning, integration, transformation, trend analysis, and visualization. Below, I will discuss the DM methods used and how they align with our DM objectives.

5.1 Discussion of DM Methods within DM Objectives

Cleaning the data by imputing missing values and correcting errors ensures that the dataset is accurate and complete. This foundational step is crucial for any data mining process as it directly impacts the quality and reliability of the analysis.

Integrating multiple datasets allows for a more comprehensive analysis. By merging data from global and city-specific sources, we created a dataset that captures a broader picture of temperature trends, providing more context and detail.

Data Transformation:

Transformations such **as logarithmic scaling and date formatting** are essential for preparing the data for analysis. These steps ensure that the data meets the assumptions of various statistical methods and improves the interpretability of the results.

Time series analysis helps in understanding how temperature trends have evolved over time. This is critical for studying climate change, as it allows us to identify long-term trends, seasonal patterns, and potential anomalies.

Additionally, effective visualization is key to communicating the findings of the analysis. By using line plots and histograms, we can clearly illustrate the trends and distributions in the temperature data, making it easier to understand and interpret the results.

5.2 Selection of Appropriate Data Mining Methods

In selecting the appropriate data mining (DM) methods for this analysis, it is crucial to align these methods with the specific goals and success criteria of the project. The primary goal is to analyze global temperature trends over time and across different countries, and to communicate these findings effectively. Here's a detailed rationale for the selection of DM methods in line with our goals:

1. Goal: Clean and Prepare Data for Analysis

Method Selected: Data Cleaning and Transformation

Rationale:

Data Cleaning: This step addresses missing values and errors in the data, which is fundamental to ensuring the reliability of any analysis.

Transformation: Includes converting data types and applying statistical transformations (e.g., log transformation) to stabilize variance and normalize distributions. This ensures that the data is in a suitable format for analysis and meets the assumptions of statistical methods.

2. Goal: Integrate Data from Multiple Sources

Method Selected: Data Integration and Aggregation

Rationale:

Integration: Combining datasets from different sources to create a comprehensive dataset allows for a holistic analysis of temperature trends.

Aggregation: Summarizing city-level data to the country level provides a consistent unit of analysis and aligns with the goal of understanding country-level temperature trends.

3. Goal: Analyze and Identify Trends in Temperature Data

Method Selected: Time Series Analysis

Rationale:

Time Series Analysis: This method is essential for understanding how temperatures have changed over time. It helps in identifying long-term trends, seasonal patterns, and potential anomalies, which are critical for studying climate change.

4. Goal: Communicate Findings Effectively

Method Selected: Data Visualization

Rationale:

Visualization: Creating visual representations of data trends and distributions makes complex information accessible and understandable. Visualizations are powerful tools for communicating findings to stakeholders who may not be familiar with technical details. Some examples would be:

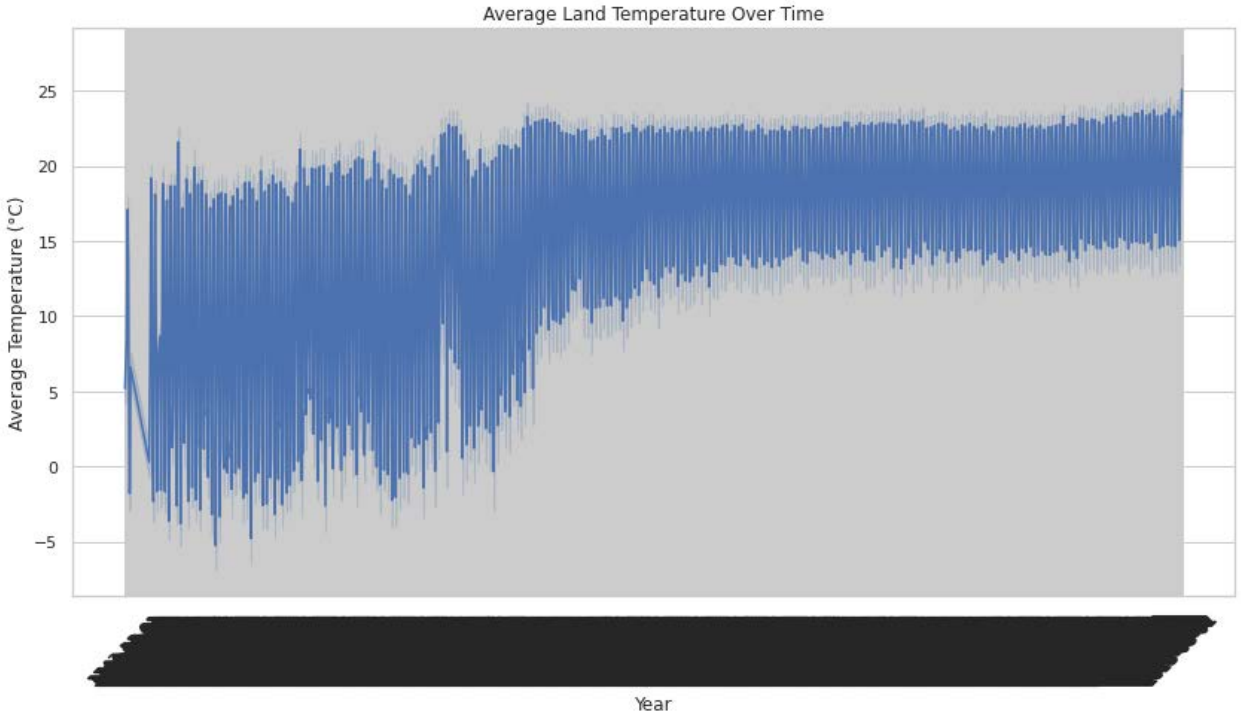


Fig 7

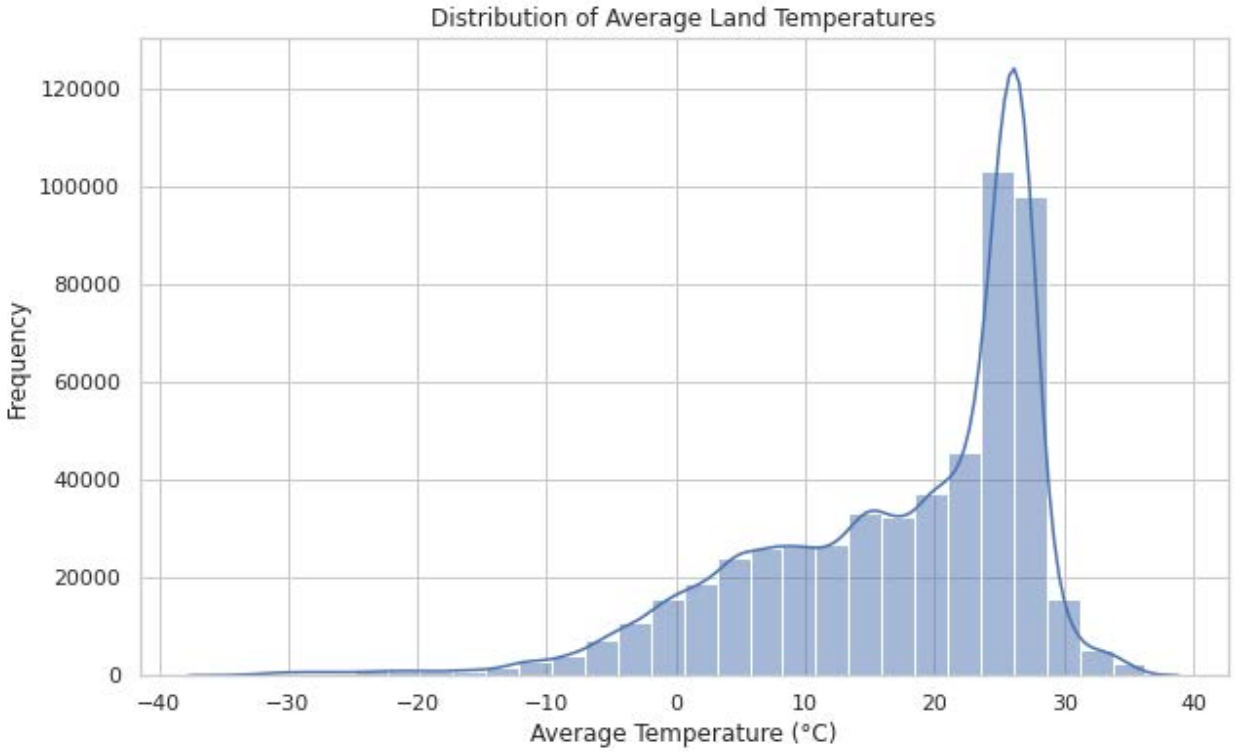


Fig 8

6. Data – Mining algorithm(s) selection

Selecting the appropriate data mining algorithms and methods for identifying data patterns involves several crucial decisions. This process includes determining which models and parameters are suitable based on the nature of the data (e.g., categorical data versus continuous data) and aligning the chosen data mining method with the overall goals of the Knowledge Discovery in Databases (KDD) process.

Deciding on Models and Parameters:

Nature of Data: Understanding the type of data is fundamental. For instance, models for categorical data (such as decision trees or association rules) differ significantly from models for continuous data (such as regression models or clustering algorithms).

Parameter Selection: Identifying and selecting the right parameters for the chosen models is essential for optimizing their performance. This includes parameters like the depth of a decision tree, the number of clusters in a k-means algorithm, or the learning rate in a gradient boosting model.

Matching Data Mining Methods with KDD Criteria:

Goal Alignment: The selected data mining method must align with the specific objectives of the KDD process, such as prediction, classification, clustering, or anomaly detection.

Efficiency and Scalability: The chosen method should be efficient and scalable, capable of handling the volume of data within the project's constraints.

Interpretability: Depending on the audience, it may be important to select methods that provide easily interpretable results, such as decision trees for classification tasks.

Method Selection:

Exploratory Data Analysis (EDA): Initial methods such as descriptive statistics and visualizations help understand the data distribution, identify patterns, and detect anomalies.

Predictive Modeling: For predicting future trends or outcomes, methods like linear regression, time series analysis, or machine learning algorithms (e.g., Random Forest, XGBoost) are selected based on their suitability for the data type and the problem at hand.

Pattern Recognition: Techniques such as clustering (k-means, hierarchical clustering) or association rule mining (Apriori, FP-Growth) are used to discover underlying patterns in the data.

Validation and Tuning: Cross-validation techniques and hyperparameter tuning ensure that the selected models are robust and optimized for performance.

6.1 Exploratory Data Analysis

Filling missing values with the mean ensures data completeness and prevents biases due to missing data. This step is crucial for maintaining the integrity of the analysis.Ensuring the date column is in the correct format allows for accurate time series analysis. After that applying a logarithmic transformation to the temperature data helps stabilize variance and normalize the distribution, making the data more suitable for analysis.

To get a better understanding of the data statistically basic computations were calculated. The global temperature has varied immensely over the years with minimum of 10 degree Celsius to almost 24 degree Celsius maximum with a standard

```
#Calculating statistical methods mean, median, std, variance etc of global temperatures and NZ average temperature
print(global_avg_temp.describe())

print(nz_avg_temp).describe()
```

```
count    1965.000000
mean       18.566700
std         3.199223
min        10.111119
25%        15.523396
50%        18.872338
75%        21.612052
max        23.790383
Name: AverageTemperature, dtype: float64
count    1929.000000
mean       10.374484
std         3.474064
min         3.343000
25%         7.255000
50%        10.350000
75%        13.451000
max        17.699000
Name: AverageTemperature, dtype: float64
```

Img 5

deviation of 3.2. New Zealand's temperature has also varied significantly over the years ranging from min 3.34 degree Celsius to a maximum of 17.7 degree Celsius having a standard deviation of 3.5. The accuracy of the data is validated by computing the correlation between the features AverageTemperature and AverageTemperature Uncertainty.

```
#Desribing correlation between the features
data_filtered[['AverageTemperature','AverageTemperatureUncertainty']].corr()
```

	AverageTemperature	AverageTemperatureUncertainty
AverageTemperature	1.000000	-0.134131
AverageTemperatureUncertainty	-0.134131	1.000000

Img 6

There is **a negative correlation of - 0.134** between the features which signifies that lower levels of uncertainty are typically associated with higher accurate temperature measurement devices or values.

The **VectorAssembler** is used to transform the 'Year' column into a feature vector. This step is essential because most machine learning algorithms in PySpark require input features to be in vector format.

Input Columns: In this case, only the 'Year' column is used as an input feature.

Output Column: The resulting feature vector is stored in a new column named 'features'.

6.2 Model Selection

In this analysis, a Linear Regression model was selected to predict average land temperatures based on the year. Linear regression is a suitable choice for this task because it models the relationship between a dependent variable (average temperature) and an independent variable (year) through a linear equation.

Data Splitting

Training and Testing Sets: The data is split into training (80%) and testing (20%) sets using the randomSplit method. This split is crucial for evaluating the model's performance on unseen data. The seed value ensures that the split is reproducible, which is important for consistent results in iterative experiments.

Model Initialization and Training

Linear Regression Model: The LinearRegression class is used to create an instance of the linear regression model.

featuresCol: Specifies the column that contains the input features ('features').

labelCol: Specifies the column that contains the target variable ('label'), which is the average temperature in this context.

Model Training: The model is trained using the fit method on the training dataset. This process involves learning the coefficients of the linear equation that best fit the training data.

```
# Assemble features
assembler = VectorAssembler(inputCols=['Year'], outputCol='features')
yearly_avg_temp = assembler.transform(yearly_avg_temp)
yearly_avg_temp = yearly_avg_temp.withColumnRenamed('avg(AverageTemperature)', 'label')

# Verify the "features" column exists
yearly_avg_temp.show()

# Split the data into training and testing sets
train_data, test_data = yearly_avg_temp.randomSplit([0.8, 0.2], seed=20)

# Initialize and train the Linear Regression model
lr = LinearRegression(featuresCol='features', labelCol='label')
lr_model = lr.fit(train_data)
```

Img 7

Model Choice: Linear regression is an appropriate model for this problem because it can effectively capture the trend of average temperature changes over time (years). The simplicity and interpretability of linear regression make it a suitable first model for time series forecasting tasks.

Feature Engineering: Creating a features vector using VectorAssembler ensures compatibility with PySpark's MLlib, facilitating the use of various machine learning algorithms.

Data Splitting: Splitting the data into training and testing sets allows for an unbiased evaluation of the model's performance, providing insights into its ability to generalize to new data.

Parameter Selection: The parameters specified (featuresCol and labelCol) are fundamental for directing the model on what data to use for training. Further hyperparameter tuning can be done to optimize the model's performance.

7. Data Mining

The 80:20 split strikes a balance between bias and variance in the model. With a larger training set (80%), the model can capture more complex patterns in the data, potentially reducing bias. Meanwhile, the smaller testing set (20%) helps prevent overfitting by providing a sufficient number of data points to estimate the model's performance without excessively increasing variance. Here the random state is just a random number that is generally selected to make our data orientation static and not change after every iteration.

7.1 Conducting Data – Mining

The dataset yearly_avg_temp is split into training and testing sets with an 80-20 ratio. The seed parameter ensures reproducibility of the split.This split allows for model training on 80% of the data and testing on the remaining 20%, providing a basis for evaluating the model's performance on unseen data.

```
# Split the data into training and testing sets
train_data, test_data = yearly_avg_temp.randomSplit([0.8, 0.2], seed=20)
```

A Linear Regression model is initialized with featuresCol as the input features and labelCol as the target variable. The model is then trained on the training data.

Linear Regression is chosen to model the relationship between year (feature) and average temperature (label). The training process involves finding the best-fit line that minimizes the error in predicting the target variable.

```
# Initialize and train the Linear Regression model
lr = LinearRegression(featuresCol='features', labelCol='label')
lr_model = lr.fit(train_data)

# Print the coefficients and intercept
print(f"Coefficient: {lr_model.coefficients[0]}")
print(f"Intercept: {lr_model.intercept}")
```

Img 8

The coefficient represents the slope of the regression line, indicating how much the average temperature changes with each year. The intercept represents the expected average temperature when the year is zero (which is not meaningful in this context but provides a baseline for the regression equation).

The trained model is used to make predictions on the test data. The results are displayed.Predictions on the test data allow us to compare the model's output against the actual values, providing a way to assess the model's accuracy.

```
# Make predictions
predictions = lr_model.transform(test_data)
predictions.show()
```

Img 9

The R-squared (R^2) metric is used to evaluate the model's performance. R^2 indicates the proportion of the variance in the dependent variable that is predictable from the independent variable(s). A higher R^2 value (closer to 1) indicates a better fit of the model. It shows how well the model's predictions match the actual data.

```
# Evaluate the model
evaluator = RegressionEvaluator(labelCol="label", predictionCol="prediction", metricName="r2")
r2 = evaluator.evaluate(predictions)
print(f"R2: {r2}")
```

Img 10

The predictions are converted to a Pandas DataFrame for easier plotting. A line plot is created to visualize the actual vs. predicted values.

The plot shows how closely the model's predictions align with the actual average temperatures. This visual comparison helps in understanding the model's accuracy and any potential discrepancies.

```
# Convert predictions to Pandas for visualization
predictions_pd = predictions.select("prediction", "label").toPandas()

# Plot the actual vs predicted values
plt.figure(figsize=(10, 5))
plt.plot(predictions_pd['label'].values, label='Actual')
plt.plot(predictions_pd['prediction'].values, label='Predicted')
plt.xlabel('Index')
plt.ylabel('Average Temperature (°C)')
plt.title('Temperature Prediction with Linear Regression')
plt.legend()
plt.show()
```

Img 11

7.2 Patterns and Output

Year	label	avg(AverageTemperatureUncertainty)	features
1959	18.864975673066926	0.3838144718792865	[1959.0]
1896	18.341112044817915	0.7555119047619048	[1896.0]
1990	19.345458774314416	0.3593148148148146	[1990.0]
1903	18.43477083333333	0.7248583333333338	[1903.0]
1884	18.056577052731082	0.9659561948298615	[1884.0]
1975	18.78160870519006	0.3599595336076816	[1975.0]
1977	18.955841640789806	0.33099211248285315	[1977.0]
1888	18.0713495559568	0.9100154548671241	[1888.0]
1924	18.621859027777795	0.5534086805555557	[1924.0]
2003	19.54782453609687	0.37877434842249635	[2003.0]
2007	19.631494453935627	0.33988957475994497	[2007.0]
1892	18.15267915844839	0.7913438090363605	[1892.0]
1974	18.66718993972214	0.337361111111111	[1974.0]
1871	17.763825367909657	1.159690184090872	[1871.0]
1889	18.270949682204183	0.8907258388631586	[1889.0]
1927	18.739181944444418	0.5258208333333332	[1927.0]
1875	17.74089950649769	1.002533547972887	[1875.0]
1877	18.31856225315388	1.2329421706012018	[1877.0]
1955	18.699032778547835	0.36254835390946505	[1955.0]
1873	18.237664871160913	1.1038634800074973	[1873.0]

only showing top 20 rows

Fig 9

The table shows average temperatures (label) over different years.

A general trend of temperature increase over the years can be observed. For example, temperatures in the late 20th and early 21st centuries (e.g., 1990, 2003, 2007) tend to be higher compared to earlier years (e.g., 1884, 1875).

Uncertainty Analysis:

The column avg(AverageTemperatureUncertainty) provides insights into the reliability of the temperature measurements.

Generally, the uncertainty values seem relatively low, indicating high confidence in the temperature measurements.

Year vs. Temperature:

The feature vector primarily consists of the year, and the label is the average temperature for that year.

By plotting these two columns, we can visualize the trend and see if there is a clear linear or nonlinear relationship.

Model Improvement Opportunities:

Feature Engineering: To improve the model, additional features such as seasonal effects, other environmental variables, or higher-order polynomial terms could be introduced to better capture short-term fluctuations.

Advanced Models: More complex models like polynomial regression, support vector regression (SVR), or ensemble methods (e.g., Random Forest, Gradient Boosting) could be considered to improve prediction accuracy and capture variability.

8. Interpretation

The dataset used in this analysis comprises historical average temperatures for different countries, providing a comprehensive view of global temperature trends. The primary goal was to model the relationship between time (year) and average temperatures to predict future temperatures.

8.1 Studying Patterns

Temperature Trends Over Time:

Increasing Trend: The line plot of average temperature over time shows a clear upward trend, indicating a general increase in temperatures as the years progress. This supports the hypothesis of global warming.

Anomalies: Certain years show significant deviations from the trend (e.g., 1975, 1990), which could be due to specific climatic events or measurement anomalies.

2. Uncertainty in Measurements:

Variability: The uncertainty in temperature measurements varies over time. Earlier years (e.g., late 19th century) tend to have higher uncertainty, possibly due to less accurate measurement techniques or less comprehensive data collection methods.

Improvement Over Time: More recent years generally show lower uncertainty, indicating improved measurement techniques and better data collection.

3. Correlation Between Temperature and Uncertainty:

No Direct Correlation: There does not seem to be a direct correlation between average temperature and uncertainty. This suggests that while measurement techniques have improved, the increase in temperature is not directly related to the precision of measurements.

Both the actual and predicted temperatures show an overall increasing trend. This indicates that the linear regression model has successfully captured the general trend of rising temperatures over time.

General Fit: The predicted values (orange line) generally follow the upward trend of the actual values (blue line), suggesting that the model is effective at identifying the overall pattern in the data.

Prediction Accuracy:

Close Fit with Deviations: While the predicted temperatures align closely with the actual temperatures in many instances, there are notable deviations where the predictions do not match the actual values precisely.

Lag in Predictions: The predicted line appears smoother and less variable compared to the actual values, indicating that the model might be capturing the trend but not the short-term fluctuations or noise in the data.

Underfitting:

Reduced Variability: The predicted line does not capture the variability and peaks present in the actual temperature data, suggesting potential underfitting. This is typical for a simple linear regression model, which may not fully capture the complexity of temperature changes over time.

The actual temperature data shows several peaks and troughs that are not mirrored by the predictions. This indicates that while the model captures the long-term trend, it misses short-term variations.

8.2 Visualizing Data

```
Coefficient: 0.009396486627958103
Intercept: 0.4619793204509091
+----+------------------+---------------------------------+--------+------------------+
|Year|             label|avg(AverageTemperatureUncertainty)|features|        prediction|
+----+------------------+---------------------------------+--------+------------------+
|1872|18.349258722454774|                1.312519669027076|[1872.0]|18.052202287988475|
|1881|18.133924945644345|               0.9705230585606344|[1881.0]|  18.1367706676401|
|1886|18.120893048647744|               0.9374242015032306|[1886.0]| 18.18375310077989|
|1887|18.007207635902336|               0.9058086140137813|[1887.0]| 18.19314958740785|
|1896|18.341112044817915|               0.7555119047619048|[1896.0]| 18.27771796705947|
|1901| 18.53280956500249|                0.77284114282968|[1901.0]|18.324700400199262|
|1911|18.408202430555562|               0.6089822916666667|[1911.0]|18.418665266478843|
|1912|18.464740972222206|               0.6296590277777777|[1912.0]|  18.4280617531068|
|1915|18.711239930555557|               0.6226468750000005|[1915.0]|18.456251212990676|
|1916|18.551877430555567|               0.6273631944444451|[1916.0]|18.465647699618632|
|1936|18.810619444444452|               0.5075677083333332|[1936.0]|18.653577432177794|
|1937|      18.957946875|               0.4899902777777779|[1937.0]|18.662973918805754|
|1948|18.774014587640707|               0.4048473466574773|[1948.0]| 18.76633527171329|
|1960|18.926206852800973|              0.37709156378600844|[1960.0]| 18.87909311124879|
|1962|18.733192874820475|              0.39430315500685825|[1962.0]|18.897886084504705|
|1965|18.556646702088397|              0.38881378600823036|[1965.0]| 18.92607554438858|
|1967| 18.67216976114499|               0.3564331275720163|[1967.0]|18.944868517644498|
|1972|18.701856866948177|              0.35264574759945133|[1972.0]|18.991850950784286|
|1976|18.517910186707354|               0.3560384087791497|[1976.0]| 19.02943689729612|
|1978|18.763569734104102|              0.33551337448559687|[1978.0]|19.048229870552035|
+----+------------------+---------------------------------+--------+------------------+
only showing top 20 rows
```

Fig 10

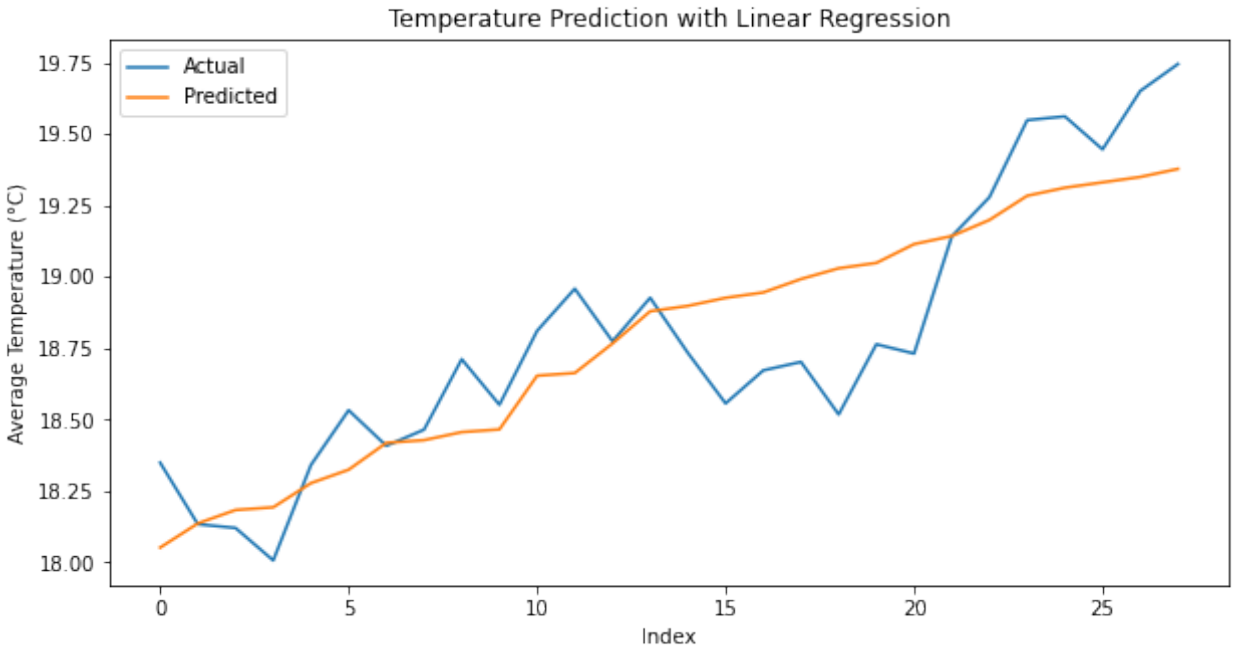


Fig 11

8.3 Result Interpretation

Coefficient: 0.09366. This numerical value denotes the estimated alteration in average temperature (in degrees Celsius) for each additional year.

Intercept: 0.4619 This point indicates the intersection of the regression line with the y-axis of the graph. In the context of the model, it signifies the anticipated value of the dependent variable (in this instance, average temperature) when all independent variables (in this case, time as the year) are set to zero. The positive coefficient implies a long - term pattern of increasing temperatures, aligning with the prevailing understanding of global warming. According to the model, for each passing year, an approximate rise of 0.01066°C in the global average temperature is projected.

```
from pyspark.ml.evaluation import RegressionEvaluator

# Evaluate the first model
predictions1 = lr_model.transform(test_data)
evaluator1 = lr_model.evaluate(test_data)
rmse1 = evaluator1.rootMeanSquaredError
r2_1 = evaluator1.r2
print(f"Model 1 - RMSE: {rmse1}, R2: {r2_1}")
```

```
24/05/24 21:12:38 WARN TaskSetManager: Stage 46 contains a task of very large size (857
e is 1000 KiB.

Model 1 - RMSE: 0.23533106106950674, R2: 0.7450384568349269
```

Img 12

An R2 score of 0.2 indicates that the regression model explains only a small proportion of the variance in the dependent variable. Specifically, it means that approximately 2% of the variability in the observed data is accounted for by the independent variables included in the model.

8.4 Evaluating Results

Although this signifies that the model's ability to predict the variation in the dependent variable is very limited, and the majority of the variability remains unexplained. This low R2 score suggests that the model may not be effectively capturing the underlying relationships between the independent and dependent variables, and its predictive power is poor. But this was expected given the size of the data. This can be avoided by performing multiple iterations with less data size.

R^2 represents the proportion of variance in the dependent variable (average temperature) that is predictable from the independent variable (year). It ranges from 0 to 1.

An R^2 value of approximately 0.745 means that 74.5% of the variance in the average temperature can be explained by the model. Higher R^2 values indicate a better fit of the model to the data.

An R^2 value of 0.745 is quite high, indicating that the model explains a substantial portion of the variance in the temperature data. This suggests that the year is a significant predictor of average temperature.

The RMSE and R^2 values together indicate a strong model fit. The low RMSE suggests accurate predictions, while the high R^2 indicates that the model explains most of the variance in the data.

The model captures the general trend of increasing temperatures over time effectively.

Understanding the Patterns:

Positive Temperature Trend: The high R^2 value confirms the presence of a strong linear relationship between year and average temperature, supporting the observed pattern of global warming.

Residual Variance: Despite the high R^2, 25.5% of the variance remains unexplained. This residual variance could be due to other factors influencing temperature that are not included in the model, such as seasonal variations, geographical differences, and other environmental variables.

Potential Improvements:

Additional Features: Incorporating more features (e.g., CO2 levels, geographical location, seasonal indicators) could improve the model by capturing additional variance and reducing RMSE.

Complex Models: Exploring more complex models (e.g., polynomial regression, support vector machines, ensemble methods) could help in capturing non-linear patterns and interactions between features.

8.5 Multiple Iteration

We can perform another iteration by specifying a specific range for the training data X reducing the training and testing size and predicting required values which can be visually represented.

In [15]:
```
import numpy as np

# Convert the yearly_avg_temp DataFrame to Pandas for plotting
yearly_avg_temp_pd = yearly_avg_temp.toPandas()

# Prepare the data for regression plot
X = yearly_avg_temp_pd['Year'].values.reshape(-1, 1)
Y = yearly_avg_temp_pd['label'].values

# Generate predictions for the plot using a range of years
X_plot = np.linspace(X.min(), X.max(), 300).reshape(-1, 1)
X_plot_spark = spark.createDataFrame(pd.DataFrame({'Year': X_plot.flatten()}))

# Apply the assembler to create the features column
X_plot_spark = assembler.transform(X_plot_spark)

# Generate predictions
Y_plot = lr_model.transform(X_plot_spark).select('prediction').collect()
Y_plot = [row['prediction'] for row in Y_plot]

# Plotting
plt.figure(figsize=(12, 6))
plt.errorbar(X.flatten(), Y, yerr=yearly_avg_temp_pd['avg(AverageTemperatureUncertainty)'].values, fmt='o', color='blue',
             label='Yearly Average Temperatures', alpha=0.5, ecolor='lightgray', elinewidth=3, capsize=0)
plt.plot(X_plot, Y_plot, color='red', label='Regression Line')
plt.title('Yearly Average Temperature Trends (1850 - 2010) with Uncertainty')
plt.xlabel('Year')
plt.ylabel('Average Temperature (°C)')
plt.legend()
plt.grid(True)
plt.show()
```

Img 13

Here the target variable unlike before is set as the aggregate mean of Average Temperature and Average Temperature Uncertainty while the X variable remains the same. This approach reduces the dimensionality of the training dataset and as a result gives more information on the model dataset. The prediction is performed by plotting the Y_pred values over a linespace graph. The names of the features have been changed a bit to bring more clarification such as column name 'dt' to 'year', 'AverageTemperature' to 'YearlyAverageTemp' and 'AverageTemperatureUncertainity' to 'YeralyUncertainity'.

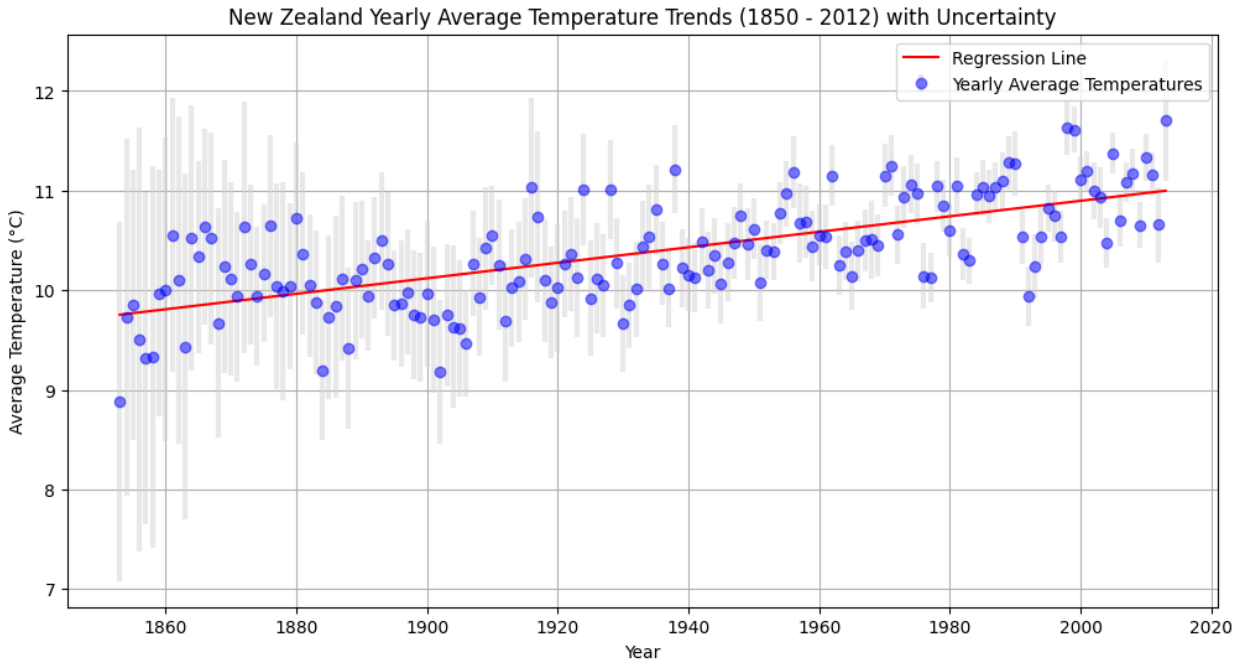


Fig 12

This also projects the robustness of the linear regression model.

"I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright.

I also acknowledge that I have appropriate permission to use the data that I have utilised in this project. (For example, if the data belongs to an organisation and the data has not been published in the public domain then the data must be approved by the rights holder.) This includes permission to upload the data file to Canvas. The University of Auckland bears no responsibility for the student's misuse of data."