Project Title	Climate Change Modeling
Tools	Jupyter Notebook and VS code
Technologies	Machine learning
Domain	Data Science
Project Difficulties level	Advanced

Dataset : Dataset is available in the given link. You can download it at your convenience.

Click here to download data set

About Dataset

Overview

This dataset encompasses over 500 user comments collected from high-performing posts on NASA's Facebook page dedicated to climate change (https://web.facebook.com/NASAClimateChange/). The comments, gathered from various posts between 2020 and 2023, offer a diverse range of public opinions and sentiments about climate change and NASA's related activities.

Data Science Applications

Despite not being a large dataset, it offers valuable opportunities for analysis and Natural Language Processing (NLP). Potential applications include:

- **Sentiment Analysis:** Gauge public opinion on climate change and NASA's communication strategies.
- **Trend Analysis:** Identify shifts in public sentiment over the specified period.

- **Engagement Analysis:** Understand the correlation between the content of a post and user engagement.
- **Topic Modeling:** Discover prevalent themes in public discourse about climate change.

Column Descriptors

1. **Date:** The date and time when the comment was posted.
2. **LikesCount:** The number of likes each comment received.
3. **ProfileName:** The anonymized name of the user who posted the comment.
4. **CommentsCount:** The number of responses each comment received.
5. **Text:** The actual text content of the comment.

Ethical Considerations and Data Privacy

All profile names in this dataset have been hashed using SHA-256 to ensure privacy while maintaining data usability. This approach aligns with ethical data mining practices, ensuring that individual privacy is respected without compromising the dataset's analytical value.

Acknowledgements

We extend our gratitude to NASA and their Facebook platform for facilitating open discussions on climate change. Their commitment to fostering public engagement and awareness on this critical global issue is deeply appreciated.

Note to Data Scientists

As data scientists analyze this dataset, it is crucial to approach the data impartially. Climate change is a subject with diverse viewpoints, and it is important to handle the data and any derived insights in a manner that respects these different perspectives.

Climate Change Modeling Machine Learning Project

Project Overview

The Climate Change Modeling project aims to develop a machine learning model to predict and understand various aspects of climate change. This can include predicting temperature changes, sea level rise, extreme weather events, and other related phenomena. The project involves analyzing historical climate data, identifying trends, and making future projections to help in planning and mitigation efforts.

Project Steps

1. **Understanding the Problem**
    - The goal is to predict and model various climate change indicators, such as temperature anomalies, precipitation patterns, and sea level changes, using historical climate data and machine learning techniques.
2. **Dataset Preparation**
    - **Data Sources**: Collect data from sources like NOAA (National Oceanic and Atmospheric Administration), NASA, IPCC (Intergovernmental Panel on Climate Change), and other climate research organizations.
    - **Features**: Include variables like temperature, precipitation, CO2 levels, solar radiation, sea level, and other relevant environmental factors.
    - **Labels**: Climate change indicators such as temperature anomalies, sea level rise, frequency of extreme weather events.
3. **Data Exploration and Visualization**
    - Load and explore the dataset using descriptive statistics and visualization techniques.
    - Use libraries like Pandas for data manipulation and Matplotlib/Seaborn for visualization.
    - Identify trends, patterns, and correlations in the data.
4. **Data Preprocessing**
    - Handle missing values through imputation or removal.
    - Standardize or normalize continuous features.
    - Encode categorical variables using techniques like one-hot encoding.
    - Split the dataset into training, validation, and testing sets.
5. **Feature Engineering**
    - Create new features that may be useful for prediction, such as rolling averages or lagged variables.
    - Perform feature selection to identify the most relevant features for the model.
6. **Model Selection and Training**
    - Choose appropriate machine learning algorithms based on the problem. Common choices include:
        - Linear Regression
        - Decision Trees
        - Random Forest
        - Gradient Boosting Machines (e.g., XGBoost)
        - Neural Networks
        - Long Short-Term Memory (LSTM) networks for time series data

    - Train multiple models to find the best-performing one.
7. **Model Evaluation**
    - Evaluate the models using metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared.
    - Use cross-validation to ensure the model generalizes well to unseen data.
    - Visualize model performance using plots like residual plots and predicted vs. actual plots.
8. **Future Projections**
    - Use the trained model to make future projections of climate change indicators.
    - Validate the projections using available data and compare them with scientific forecasts and models.
9. **Scenario Analysis**
    - Conduct scenario analysis to understand the impact of different factors (e.g., CO2 emission scenarios) on climate change.
    - Use the model to simulate different scenarios and assess their potential impact.
10. **Deployment (Optional)**
    - Deploy the model using a web framework like Flask or Django.
    - Create a user-friendly interface where users can input data and receive climate change predictions and scenarios.
11. **Documentation and Reporting**
    - Document the entire process, including data exploration, preprocessing, feature engineering, model training, evaluation, and projections.
    - Create a final report or presentation summarizing the project, results, and insights.

Sample Code

Here’s a basic example using Python and scikit-learn to model climate change indicators

```
# Import necessary libraries
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Load the dataset
# Example: Using a mock dataset with climate data
data = pd.read_csv('climate_data.csv')

# Explore the dataset
print(data.head())
print(data.describe())

# Preprocess the data
# Separate features and labels
X = data.drop('temperature_anomaly', axis=1)
y = data['temperature_anomaly']

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train the model
model = RandomForestRegressor(random_state=42)
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'MAE: {mae}')
print(f'MSE: {mse}')
print(f'R2: {r2}')

# Plot the results
plt.figure(figsize=(10, 6))
plt.scatter(y_test, y_pred, alpha=0.5)
plt.xlabel('Actual Temperature Anomaly')
plt.ylabel('Predicted Temperature Anomaly')
plt.title('Actual vs Predicted Temperature Anomaly')
plt.show()

# Future projections (mock example)
# Assuming we have future data for the same features
future_data = pd.read_csv('future_climate_data.csv')
future_data_scaled = scaler.transform(future_data)
future_predictions = model.predict(future_data_scaled)

print(future_predictions)
```

This code demonstrates loading a climate dataset, preprocessing the data, training a Random Forest regressor, evaluating the model, and making future projections.

Additional Tips

- Incorporate domain expertise to ensure the model's predictions are realistic and scientifically valid.
- Use advanced time series forecasting techniques like LSTM networks for more accurate long-term predictions.
- Continuously update the model with new data to improve its accuracy and relevance over time.
- Collaborate with climate scientists to validate and interpret the model's predictions.

Sample Project Report

n [4]:

```
df
```

ut[4]:

	date	likesCount	profileName	commentsCount	text
0	2022-09-07T17:12:32.000Z	2	4dca617d86b3fdce80ba7e81fb16e048c9cd9798cdfd6d...	NaN	Neat comparison I have not heard it before.\n ...
1	2022-09-08T14:51:13.000Z	0	518ab97f2d115ba5b6f03b2fba2ef2b120540c9681288b...	NaN	An excellent way to visualise the invisible! T...
2	2022-09-07T17:19:41.000Z	1	d82e8e24eb633fd625b0aef9b3cb625cfb044ceb8483e1...	3.0	Does the CO2/ghg in the troposphere affect the...
3	2022-09-08T00:51:30.000Z	4	37a509fa0b5177a2233c7e2d0e2b2d6916695fa9fba3f2...	NaN	excellent post! I defo feel the difference - o...
4	2022-09-07T19:06:20.000Z	16	e54fbbd42a729af9d04d9a5cc1f9bbfe8081a31c219ecb...	26.0	Yes, and carbon dioxide does not harm the Eart...
...	...	...	...	...	...
517	2022-12-22T17:21:37.000Z	0	9e17b1a6422032d47472f0216c73aafda7587e302eed5e...	NaN	One can only hope for a peak 😞
518	2022-12-22T17:19:51.000Z	1	48e55d898603a136aefc44771f248bffd67242583a462a...	5.0	what is the error margin for the temperature e...

In [5]:

```python
MODEL = f"cardiffnlp/twitter-roberta-base-sentiment"
tokenizer = AutoTokenizer.from_pretrained(MODEL)
model = AutoModelForSequenceClassification.from_pretrained(MODEL)
```

```
config.json: 100% 747/747 [00:00<00:00, 39.3kB/s]

vocab.json: 100% 899k/899k [00:00<00:00, 2.75MB/s]

merges.txt: 100% 456k/456k [00:00<00:00, 5.53MB/s]

special_tokens_map.json: 100% 150/150 [00:00<00:00, 9.28kB/s]

pytorch_model.bin: 100% 499M/499M [00:01<00:00, 304MB/s]
```

```
/opt/conda/lib/python3.10/site-packages/torch/_utils.py:831: UserWarning: TypedStorage is d
```

ETL

In [9]:

```python
df['sentiment'] = df['text'].apply(sentiment_analysis)
```

In [10]:

```python
df['label'] = df['sentiment'].apply(lambda x: sentimental_label(x))
```

In [11]:

```python
df['keywords'] = df['text'].apply(extract_keywords)
```

Graphs ¶

In [12]:

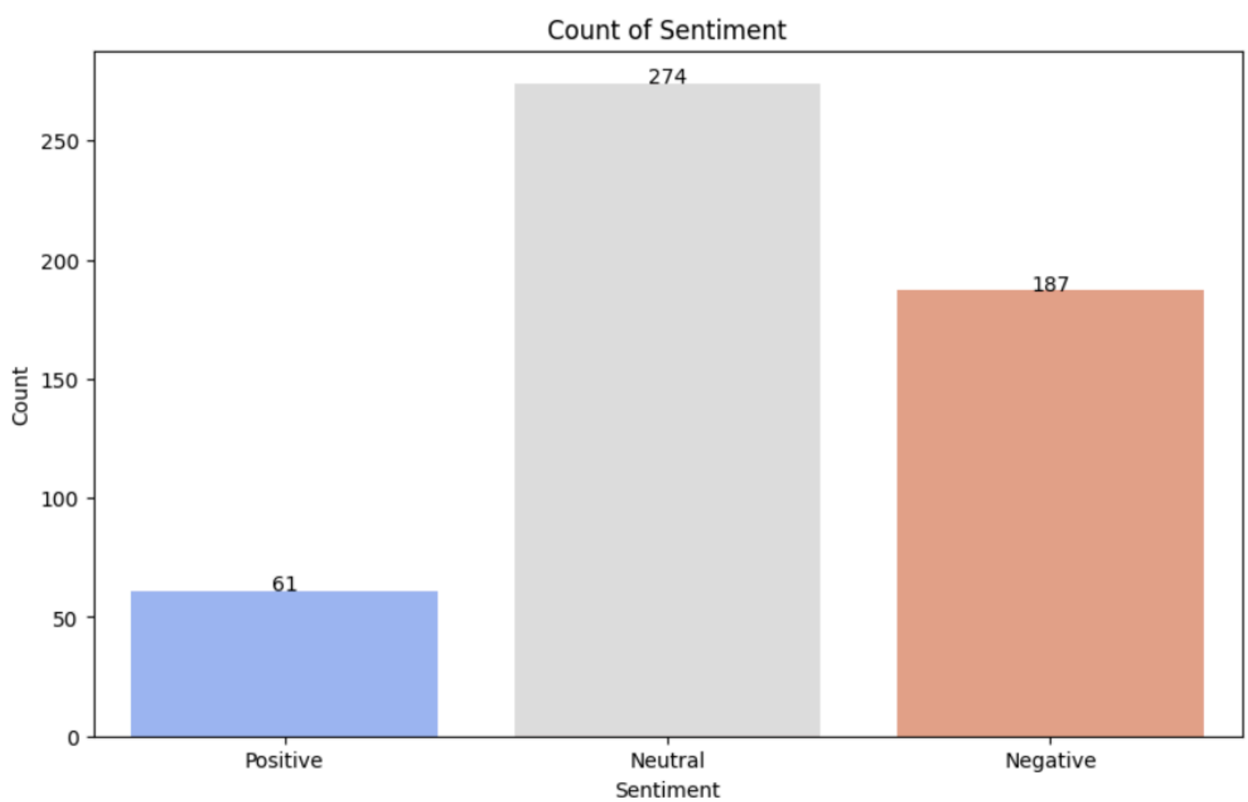
```
positive_count = df[df['label'] == 'positive'].count()[0]
neutral_count = df[df['label'] == 'neutral'].count()[0]
negative_count = df[df['label'] == 'negative'].count()[0]

labels = ['Positive', 'Neutral', 'Negative']
counts = [positive_count, neutral_count, negative_count]

plt.figure(figsize=(10, 6))
barplot = sns.barplot(x=labels, y=counts, palette='coolwarm')

for index, value in enumerate(counts):
    plt.text(index, value, f'{value}', color='black', ha="center")

plt.title('Count of Sentiment')
plt.xlabel('Sentiment')
plt.ylabel('Count')
plt.show()
```



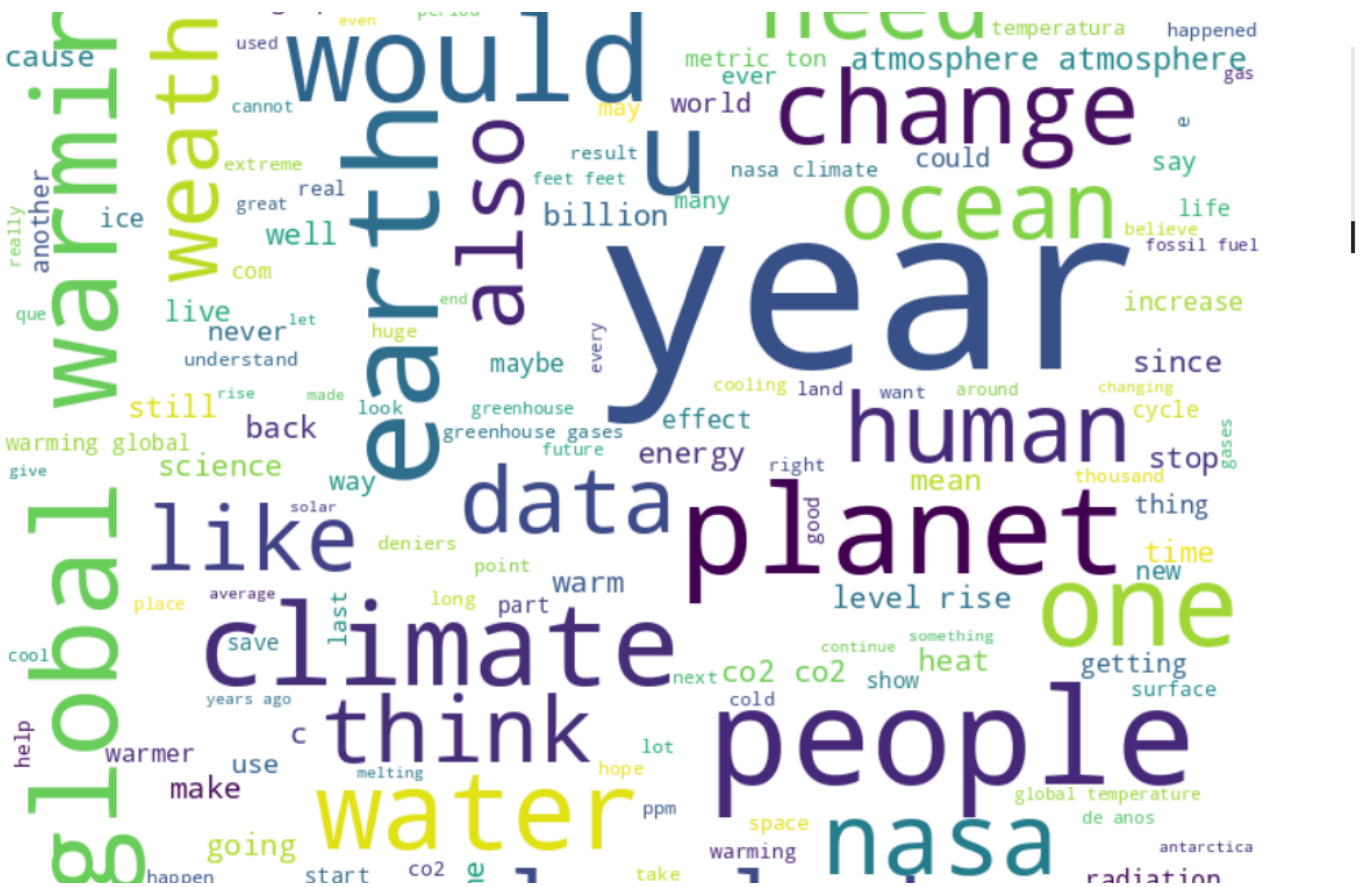
In [13]:

```
all_keywords = ' '.join([' '.join(keywords) for keywords in df['keywords'] if isinstance(keywo
rds, list)])

wordcloud = WordCloud(width = 800, height = 800,
                background_color ='white',
                stopwords = set(),
                min_font_size = 10).generate(all_keywords)

plt.figure(figsize = (8, 8), facecolor = None)
plt.imshow(wordcloud)
plt.axis("off")
plt.tight_layout(pad = 0)

plt.show()
```



In [14]:

```
all_keywords = list(itertools.chain.from_iterable(df['keywords']))

keyword_counts = Counter(all_keywords)

most_common_keywords = keyword_counts.most_common(10)
words, counts = zip(*most_common_keywords)

plt.figure(figsize=(12, 6))
plt.bar(words, counts)
plt.xlabel('Key words')
plt.ylabel('Frequency')
plt.title('Top 10 Most Frequent Keywords')
plt.xticks(rotation=45)
plt.show()
```