ANALYZING REDDIT COMMENTS ON PALESTINE AND ISRAEL CONFLICT

I. INTRODUCTION:

Exploring the multifaceted dynamics of the Israeli-Palestinian conflict, this project delves into the intricacies of its 75-year span, marked by sporadic violence and elusive peace attempts, while also examining the profound influence of social media on shaping public perceptions and fostering dialogue amidst the ongoing turmoil.

Grasping Complexity:

- The conflict spans over 75 years, marked by sporadic violence and occasional peace attempts.
- Temporary conflict management strategies have consistently failed, merely delaying inevitable escalations.
- The absence of faith in a comprehensive peace pact underscores the challenge of reconciling deep-seated differences.
- International mechanisms established post-World War II have struggled to broker lasting peace.
- Israel's dominance, evident through West Bank settlements and Gaza control, obstructs Palestinian statehood.
- Stalemate arises from both Israeli and Palestinian reluctance to make concessions necessary for a viable two-state solution.

Influence of Social Media:

- Social media significantly shapes public perception by amplifying narratives and disseminating information.
- Both sides exploit online platforms to reinforce existing biases and further their agendas.
- However, social media can also facilitate dialogue and promote peace if leveraged constructively.

II. RESEARCH ON REDDIT FOR POLITICAL DISCUSSIONS

Reveals a dynamic landscape characterized by diverse communities representing a spectrum of ideologies and viewpoints. Within these communities, users tend to cluster

in echo chambers, reinforcing their existing beliefs and contributing to polarization. Despite efforts by
moderators to maintain civil discourse, striking a balance between fostering free speech and preventing toxicity poses significant challenges.

III. FLOW DIAGRAM: DATA COLLECTION AND PROCESSING PIPELINE

1. **Data Collection**:
   - Reddit comments were gathered from pertinent subreddits such as r/worldnews, r/IsraelPalestine, and r/AskMiddleEast.
   - Information including comment ID, score, self-text, subreddit, and creation time was collected.
2. **Preprocessing**:
   - Text cleaning involved removing special characters, URLs, and irrelevant content from the comments.
   - Tokenization was performed to break down comments into words or phrases.
   - Sentiment analysis was conducted to gauge the emotional tone of the comments.
3. **Feature Extraction**:
   - Relevant features such as sentiment scores and word frequencies were extracted.
   - A bag-of-words representation was created to capture the essence of the comments.
4. **Analysis and Visualization:**
   - Patterns were explored to identify common themes, controversial topics, and sentiment trends within the comments.
   - Comment scores were visualized over time to observe any fluctuations or patterns.
5. **Insights and Interpretation:**
   - User sentiments, biases, and engagement levels were understood through analysis.
   - The impact of specific events or news on discussions was investigated to gain further insights.

IV. DATA PREPROCESSING

Data preprocessing is crucial in the machine learning pipeline as it ensures that raw data is transformed into a clean, structured format, enabling models to learn effectively and produce accurate predictions.

The process of data preprocessing, particularly in the context of text data for analysis or machine learning tasks, involves several crucial steps aimed at enhancing the quality and usability of the dataset. Firstly, noise removal techniques are applied to eliminate irrelevant information that may distort analysis results. This includes the elimination of URLs and HTML tags using regular expressions, as well as the lowercasing of all text data to ensure consistency and the removal of special characters and punctuation marks that may not contribute to the analysis.

Furthermore, **handling missing values** is imperative to prevent bias and inaccuracies in the analysis. Techniques such as imputation or removal are employed to address missing data points effectively. Text preprocessing techniques are then applied to further refine the data for analysis. **Tokenization** involves splitting text into individual words or tokens, facilitating subsequent analysis. **Lemmatization** reduces words to their base or root form, ensuring consistency in the dataset. Additionally, **stop-word removal** eliminates common words, known as stop words, that do not carry significant meaning and may introduce noise into the analysis.

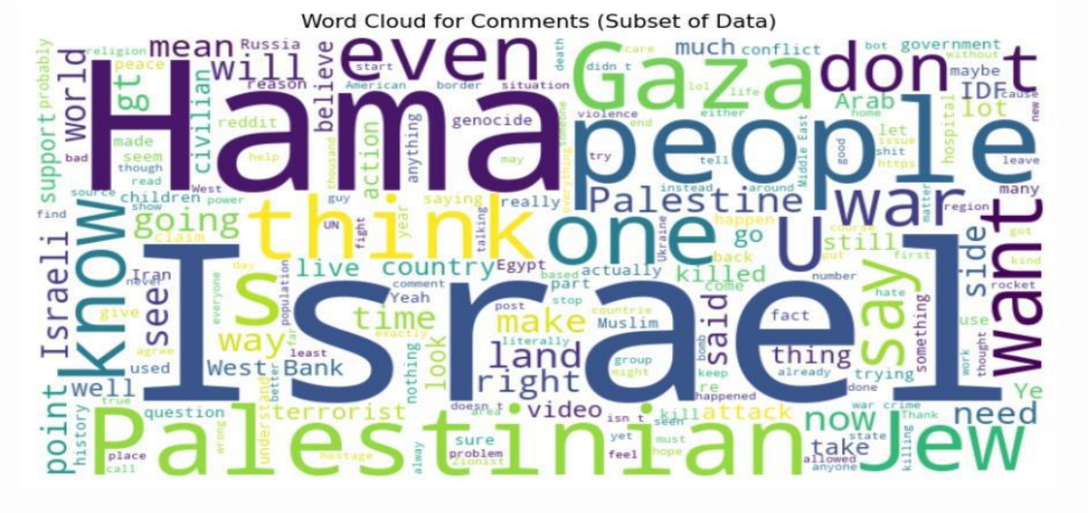
Finally, **Data quality assessment** is essential to evaluate the effectiveness of the preprocessing steps and ensure the reliability of the dataset. This assessment includes examining summary statistics, such as mean and median, for numerical features, as well as analyzing the distribution of categorical variables. Visualization techniques, such as histograms and box plots, are utilized to gain insights into the data distribution and identify any anomalies or patterns that may require further investigation. Overall, thorough data preprocessing and quality assessment are essential prerequisites for meaningful analysis and reliable results.

V. FEATURE ENGINEERING:

In the pursuit of enhancing model performance, the process of creating new features from existing data serves as a crucial strategy. One approach involves the creation of new features, such as introducing a 'vote' feature derived from the existing 'score' column to signify upvotes or downvotes, thereby enriching the dataset with additional insights. Text feature engineering is another avenue explored, encompassing techniques like TF-IDF vectorization, which transforms textual data into numerical vectors based on the frequency of terms across documents, and word embeddings, such as Word2Vec, which captures semantic relationships between words. Furthermore, domain-specific feature engineering integrates domain knowledge to devise meaningful features. For instance, incorporating features related to domain-specific metrics or indicators augments the dataset with contextually relevant information. To gain insights into feature importance, visualization techniques are employed to showcase feature importance scores derived from models or feature selection methodologies, thus elucidating the pivotal role of key features in influencing model predictions. Through these methodologies, the endeavor aims to optimize model performance by leveraging existing data resources and harnessing domain expertise.

VI. GENERATING WORD CLOUD FOR COMMENTS

In this section, we illustrate the distribution of words in the comments subset of the dataset using a word cloud visualization technique. Utilizing the WordCloud library, we transform the text data into a visual representation, wherein the size of each word corresponds to its frequency in the comments. By aggregating the text data from the 'self_text' column of the dataset, we construct a comprehensive corpus. The resulting word cloud provides a succinct overview of the prevalent themes and topics within the comments subset. Through this visualization, we gain insights into the most commonly occurring words, enabling us to identify prominent trends and patterns within the dataset. The generated word cloud serves as a valuable exploratory tool for understanding the content and context of the comments, facilitating further analysis and interpretation.



Word Cloud for Comments (Subset of Data)

VII. UTILIZING TF-IDF VECTORIZATION FOR TEXTUAL FEATURE ENGINEERING

In our effort to enhance the predictive capabilities of our model, we employed TF-IDF (Term Frequency-Inverse Document Frequency) vectorization, a fundamental technique in natural language processing. By leveraging the TfidfVectorizer module from the sklearn.feature_extraction.text library, we transformed textual data into numerical representations. With the specified parameters, including an n-gram range of (1, 1) and a maximum of 5000 features, we processed the cleaned text data extracted from our dataset. This transformation facilitated the conversion of text into a sparse matrix format, where each row corresponds to a document and each column represents a unique term. These transformed features, coupled with their respective target variable, form the basis for training our predictive model. Through the utilization of TF-IDF vectorization, we aim to capture the inherent semantic nuances within the textual data, thereby enriching the feature space and ultimately improving the performance of our predictive model.

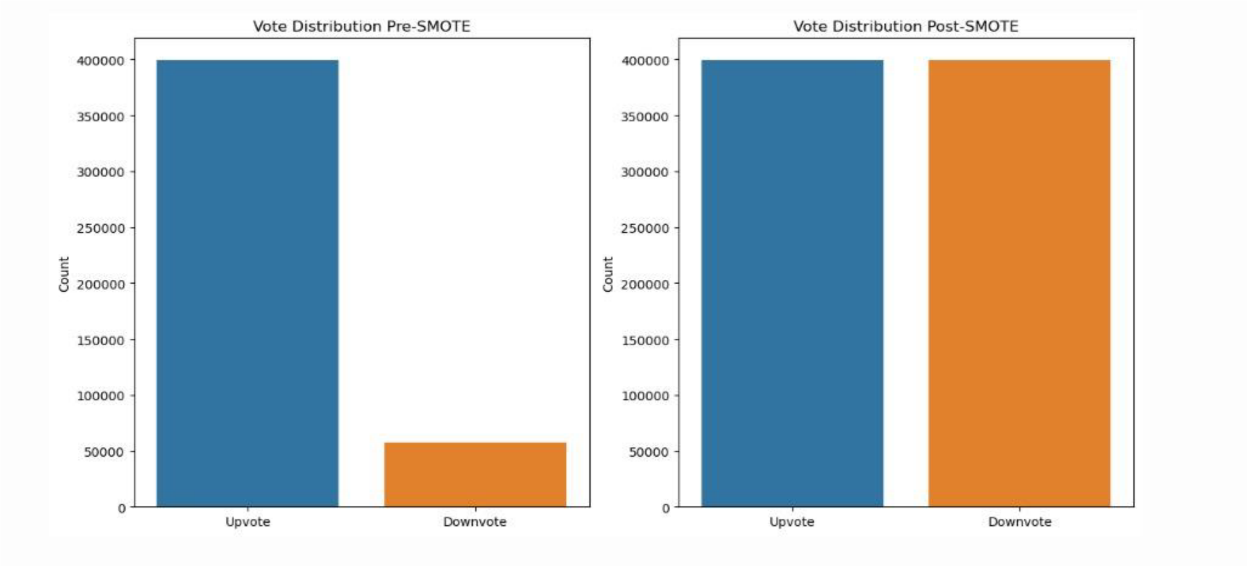
VIII. MODEL TRAINING USING RANDOM FOREST CLASSIFIER WITH SMOTE OVERSAMPLING

In our project, we employed a Random Forest Classifier as our predictive model due to its robustness and ability to handle complex datasets effectively. The Random Forest algorithm constructs multiple decision trees during training and combines their predictions through voting to generate the final output. To address class imbalance within our dataset, we utilized the Synthetic Minority Over-sampling Technique (SMOTE) to augment the minority class samples. This technique generates synthetic samples by interpolating between existing minority class instances, thereby balancing the class distribution. By applying SMOTE to our training data, we ensured that the classifier learned from a more representative dataset, reducing the risk of biased predictions towards the majority class.

Our Random Forest Classifier was configured with 50 decision trees (n_estimators=50) and additional parameters such as min_samples_split, min_samples_leaf, and random_state were tuned to optimize model performance. The n_jobs parameter was set to -1 to utilize all available CPU cores for parallel processing, expediting model training.

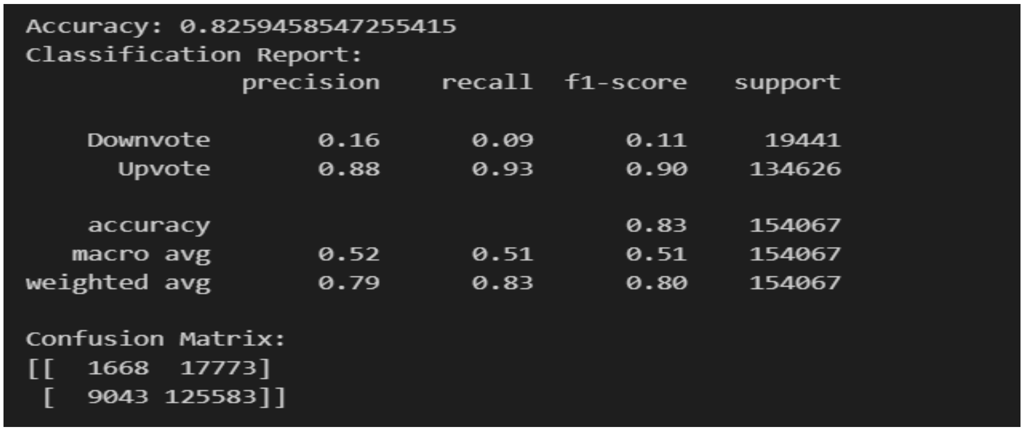
After training the Random Forest Classifier on the augmented training data, we evaluated its performance on the unseen test data. The classifier made predictions on the test set, and the resulting predictions (**y_pred**) were compared against the actual labels (**y_test**). Through this process, we assessed the model's accuracy, precision, recall, and other performance metrics to determine its effectiveness in making predictions on our dataset.

Overall, the Random Forest Classifier trained with SMOTE oversampling represents a key component of our project's predictive modeling pipeline, offering a reliable approach for addressing class imbalance and generating accurate predictions.



IX. MODEL PERFORMANCE AND EVALUATION

- **Accuracy** :quantifies the ratio of correctly classified instances to the total instances, offering a holistic evaluation of the model's predictive capability.
- **Classification Report**: This report furnishes a detailed overview of performance metrics like precision, recall, F1-score, and support for individual classes, providing insights into the model's effectiveness across diverse categories.
- **Confusion Matrix**: Depicting true positives, false positives, true negatives, and false negatives, confusion matrix facilitates a visual examination of prediction errors and misclassifications, aiding in thorough analysis of model performance.



```
Accuracy: 0.8259458547255415
Classification Report:
              precision    recall  f1-score   support

    Downvote       0.16      0.09      0.11     19441
      Upvote       0.88      0.93      0.90    134626

    accuracy                           0.83    154067
   macro avg       0.52      0.51      0.51    154067
weighted avg       0.79      0.83      0.80    154067

Confusion Matrix:
[[  1668  17773]
 [  9043 125583]]
```

MODEL FUNCTIONING :

1.*Accuracy*: The model's overall correctness is around 83.42%. It correctly predicts the class of approximately 83.42% of the samples.

2. *Precision*:

- Precision for class 0: 0.19. This means that when the model predicts an instance as class 0, it's correct about 19% of the time.

- Precision for class 1: 0.88. When the model predicts an instance as class 1, it's correct about 88% of the time.

3. *Recall*:

- Recall for class 0: 0.09. Out of all actual instances of class 0, the model correctly identifies 9%.

- Recall for class 1: 0.94. The model correctly identifies 94% of the instances of class 1.

4. *F1-score*:

- F1-score is the harmonic mean of precision and recall. It provides a balance between precision and recall.

- F1-score for class 0: 0.13.

- F1-score for class 1: 0.91.

5. *Support*: The number of actual occurrences of each class in the specified dataset.

- Class 0 has a support of 24,307.

- Class 1 has a support of 166,713**.**

6. *Confusion Matrix*:

- It shows the counts of true positive, true negative, false positive, and false negative predictions.

- In your case, out of 24,307 instances of class 0, the model correctly predicted 2,290 as class 0 and 22,017 as class 1.

- Out of 166,713 instances of class 1, the model correctly predicted 157,064 as class 1 and misclassified 9,649 as class 0.

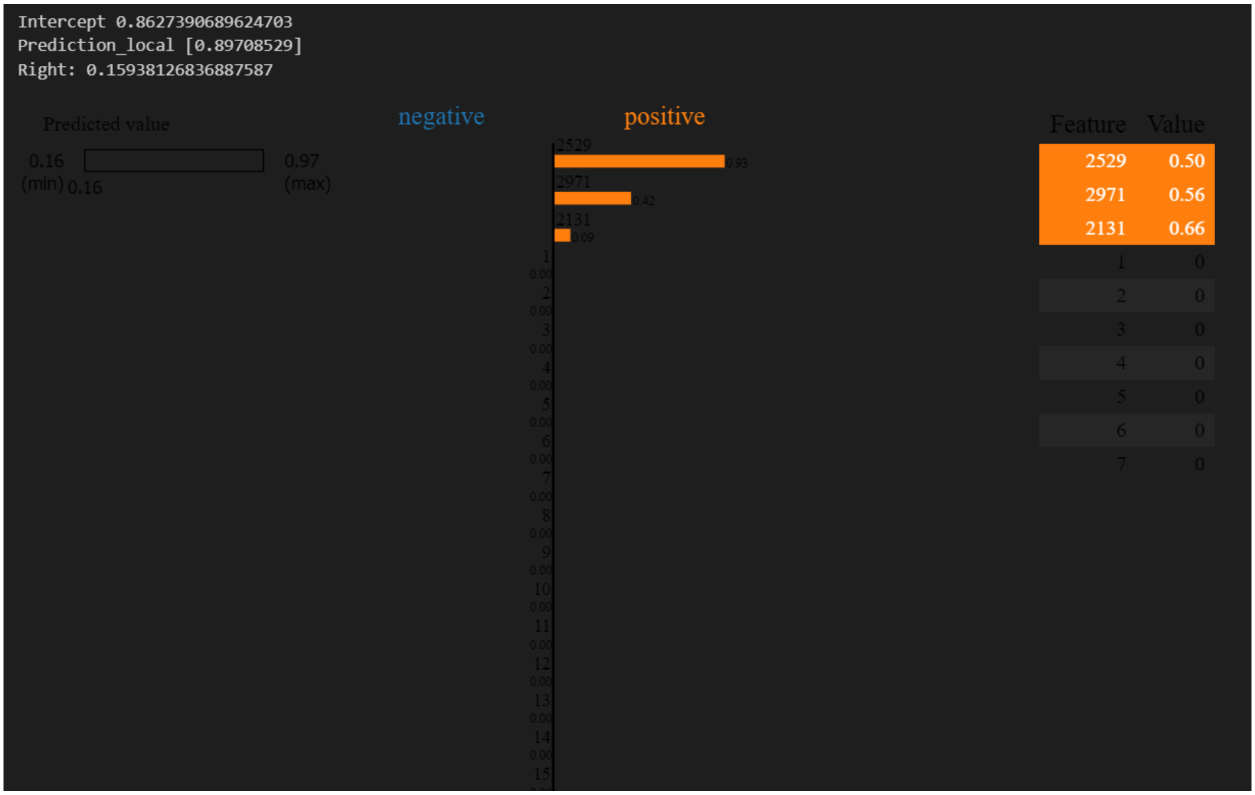
X. MODEL INTERPRETABILITY

The report highlights the significance of LIME (Local Interpretable Model-agnostic Explanations) as a technique for enhancing the interpretability of machine learning models. LIME serves as a valuable tool in understanding the predictions made by complex machine learning models, particularly when dealing with intricate decision-making processes.

LIME operates by generating local approximations of the model's behavior around individual predictions. This involves perturbing input features and observing the resulting changes in predictions, thus providing insights into the rationale behind the model's decisions.

A key concept employed by LIME is the use of "interpretable proxies" to approximate the behavior of complex models. These proxies are simplified, interpretable models that serve as substitutes for the original model within local contexts, aiding in the understanding of its decision-making process.

The insights gained from LIME explanations are invaluable, as they reveal the key features influencing the model's predictions for specific instances. This understanding facilitates model tuning and validation by identifying influential features for refinement and validation strategies.

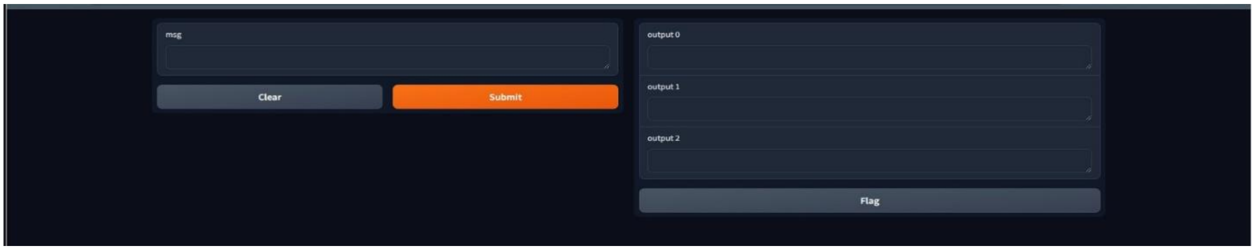


```
Intercept 0.8627390689624703
Prediction_local [0.89708529]
Right: 0.15938126836887587
```

Feature	Value
2529	0.50
2971	0.56
2131	0.66
1	0
2	0
3	0
4	0
5	0
6	0
7	0

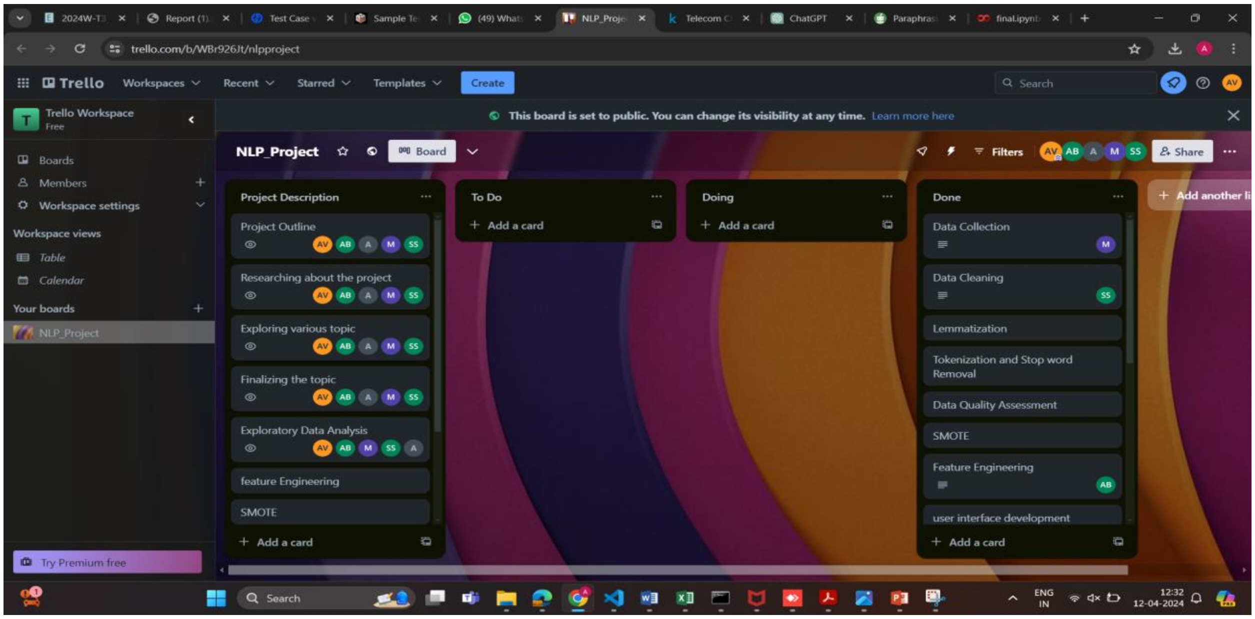
XI. USER INTERFACE :

The Gradio interface is designed to provide users with an intuitive and interactive platform for predicting the likelihood of upvotes or downvotes for Reddit posts based on their content. Upon entering a message into the provided text input field, users can initiate the prediction process. The interface then displays the predicted classification of the post (either 'Upvote' or 'Downvote') along with the corresponding probabilities for each outcome. These probabilities offer insights into the confidence level of the prediction, enabling users to gauge the potential reception of their posts. With its user-friendly design and real-time feedback, the Gradio interface empowers users to make informed decisions regarding their content creation strategies, ultimately enhancing their engagement and visibility on the platform.



XII. TRELLO BOARD

The Trello board for this project serves as an organizational hub, facilitating efficient task management and progress tracking. It is divided into several key sections.



This board is set to public. You can change its visibility at any time. Learn more here

NLP_Project

Project Description
- Project Outline
- Researching about the project
- Exploring various topic
- Finalizing the topic
- Exploratory Data Analysis
- feature Engineering
- SMOTE
- Add a card

To Do
- Add a card

Doing
- Add a card

Done
- Data Collection
- Data Cleaning
- Lemmatization
- Tokenization and Stop word Removal
- Data Quality Assessment
- SMOTE
- Feature Engineering
- user interface development
- Add a card

https://trello.com/b/WBr926Jt/nlpproject

GITHUB LINK - https://github.com/sharansara/Reddit-comment-on-israeli---palasteine