Results and Conclusions

1.Implement a user-friendly search functionality using a provided sample dataset of 5000 user queries.

- *I've built a tool with a dataset of 5000 user queries, each assigned a score. However, I decided to focus on returning queries only, as almost 99.9% of the scores are zeros. This ensures a simpler and more streamlined output for users.*
- *The tool helps find queries similar to what a user searches for. It uses pre-trained models. When a user searches, the tool looks through the dataset and gives back the top three similar queries based on scores.*

Approch

1.Analyzing the User's Query : *The user's query is preprocessed by converting it to lowercase. This ensures that the query is treated consistently, regardless of the case.*

2.Retrieving Relevant Results from the Dataset:

- *Using TF-IDF vectorizer is employed to transform the processed text from the dataset into a numerical representation.*
- *The search function takes a user's query, preprocesses it similarly, handles misspelled words using SpellChecker, and transforms the query using the same TF-IDF vectorizer.*

3.Ranking Results Based on Relevance:

- *Cosine similarity is used to calculate the similarity between the user's query vector and the vectors of the dataset.*
- *The results are then ranked based on these similarity scores.*

4.Displaying the Top 3 Relevant Results:

- *return top 3 relevant results are extracted from Queries dataset otherwise print "No results found."*

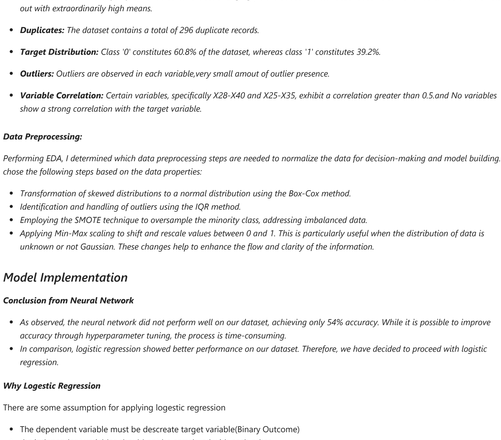
5. User-friendly search model deploye on web application using streamlit turns data scripts into shareable web apps.

About Streamlit

- *Streamlit is a Python-based web application framework and it is a free and open-source library.*
- *Streamlit is an open-source Python library that is designed to make it easy to create web applications for data science and machine learning.*
- *It allows you to turn data scripts into shareable web apps with minimal effort. Streamlit simplifies the process of creating interactive and visually appealing data applications, making it accessible to both experienced developers and those with less programming experience.*

- *To run a Streamlit app from the command line, you typically use the streamlit run command followed by the filename of your Python script containing the Streamlit app. Here's the basic Step:*
  - *step 1: First change to the "Documents" directory in cmd.(provide the path to the directory you want to change to)*
  - *step 2: write command >>streamlit run app.py*

As you can see in the sample below



02. Binary Classification Model Building

Summary of Dataset Analysis:

the dataset reveals several key insights are belows

- ***Data Skewness:*** *The data is skewed, with approximetly above 90% of zero values present in each variable.*
- ***Variable Types:*** *All variables are numerical, including the target variable.*
- ***Variable Means:*** *the majority of variables have means ranging from 0.006054 to 6.048653. observe some variable, X56 and X57 stand out with extraordinarily high means.*
- ***Duplicates:*** *The dataset contains a total of 296 duplicate records.*
- ***Target Distribution:*** *Class '0' constitutes 60.8% of the dataset, whereas class '1' constitutes 39.2%.*
- ***Outliers:*** *Outliers are observed in each variable,very small amout of outlier presence.*
- ***Variable Correlation:*** *Certain variables, specifically X28-X40 and X25-X35, exhibit a correlation greater than 0.5.and No variables show a strong correlation with the target variable.*

Data Preprocessing:

Performing EDA, I determined which data preprocessing steps are needed to normalize the data for decision-making and model building. I chose the following steps based on the data properties:

- *Transformation of skewed distributions to a normal distribution using the Box-Cox method.*
- *Identification and handling of outliers using the IQR method.*
- *Employing the SMOTE technique to oversample the minority class, addressing imbalanced data.*
- *Applying Min-Max scaling to shift and rescale values between 0 and 1. This is particularly useful when the distribution of data is unknown or not Gaussian. These changes help to enhance the flow and clarity of the information.*

Model Implementation

Conclusion from Neural Network

- *As observed, the neural network did not perform well on our dataset, achieving only 54% accuracy. While it is possible to improve accuracy through hyperparameter tuning, the process is time-consuming.*
- *In comparison, logistic regression showed better performance on our dataset. Therefore, we have decided to proceed with logistic regression.*

Why Logestic Regression

There are some assumption for applying logestic regression

- The dependent variable must be descreate target variable(Binary Outcome)
- the independent variables should not be correlated with each other
- requires quite large sample sizes

And Using data pre-processing steps, our set optimal variables are satisfied by all these assumptions.

Conclusion from Logestic Regression

- *There were 3910 records in the dataset, out of which 80% of the data was given for training the model and 20% of the validation data, i.e., 951 records, were given for testing the model. And out of 951 records, 121 records were misclassified.*

- *Considering these results, a decision can now be made based on the insights derived from the dataset analysis and logistic regression.*
- *As we can see that model results on validation data , its shows*
  - ***Accuracy:*** *the accuracy show that 0.87 ,which indicates that the accuracy of the model is 87%.*
  - ***Precision:*** *the precision show that 0.85 ,which indicates that the precision of the model is 86%.*
  - ***Recall:*** *the Recall show that 0.89 ,which indicates that the Recall of the model is 89%.*
  - ***F1-score:*** *the f1-score show that 0.87 ,which indicates that the f1-score of the model is 87%.*
  - ***AUC score:*** *model is performing well and its closest to 1 (93%) which is indicates that the model is performing well in distinguishing between the two classes (positive and negative).*

Predication On Testing Data

- *we can also testing on given test data that our model prediction results for the test dataset, which consisted of 691 records. Our model classified 90.9% of the samples as class 1 (Postive) and 9.1% as class 0(Negative).*

Note : *when we apply the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance in a dataset, the original shape of the data can change. SMOTE creates synthetic examples of the minority class by interpolating between existing minority class instances. This process introduces new instances that weren't in the original dataset.*

03. Decision Making Based On given Data

- *Considering these results, a decision can now be made based on the insights derived from the dataset analysis and logistic regression.*
- *The model looks good for making decisions in our situation. It's accurate and balances well between being precise and capturing all relevant cases. We can trust it to predict outcomes related to our goal.However, we need to be aware of the possibility of getting some predictions wrong. It's crucial to understand the impact of both false positives and false negatives based on the specific problem we're dealing with.*
- *In conclusion, the model is a useful tool for our predictions.*

04.Suggestion On Dataset

- *In reviewing the dataset, I noticed that the variable names (X1, X2, etc.) are not very informative. To make better decisions for the organization,to store information on assigen specific name of variables that provide meaningful for decisions.*

- *for Example ,in a scenario like predicting customer behavior, like DPD, age, number of loans, income, and tenure, the organization can enhance decision-making and better predict outcomes related to the target variable. . Using this variables we make decision for minimise credit risk*

05.Output Files Description

The following is the name of the folder within the folder contains files with descriptions that contain the "Codes_And_Results" folder:

These folder contain all output files and codes files that provide a comprehensive overview of the dataset analysis, model building, and outcomes, making it easier for to navigate each files

1. Codes:

- ***app.py:*** *This Python file uses the Streamlit library to execute search queries on the dataset.*
- ***Task_Global_AI_Solution.ipynb:*** *This file, written in Python using Jupyter Notebook, covers the steps involved in building a binary classification model. It includes tasks like Exploratory Data Analysis (EDA), Data Preprocessing, Feature Selection, and Model Building.*

2. Features_Importance:

- ***overview_of_each_var:*** *An HTML file providing a summary of each variable's characteristics, offering insights into the quality of the data.*
- ***Variables_For_Predication:*** *This file outlines the final set of optimal variables selected for prediction, detailing the process of variable exclusion and highlighting their importance scores.*
- ***Variables_Stat_summary:*** *Contains descriptive statistics for each variable, including data dispersion, data types, and volume of data.*

3. Outputs:

- ***test_data_with_predication:*** *This file includes the predicted labels (Class 1 or Class 0) for the test data.*
- ***updated_submission:*** *Presents the target class labels predicted for the test data after the model-building process.*

4. HTML_Files: *its includes all codes HTML files*

In [ ]:

Thank You

Suraj I Meshram

-Contact No.: +91-8888675122

LinkedIn | GitHub | Hackerrank | HackerEarth

In [ ]: