AI511 HELP BOOST OUR ONLINE REACH!

Team name : cloud9

Manan Mehta IIITB (MT2021517)

Vishwajeet Deulkar IIITB (MT2021154)

PROBLEM STATEMENT AND DATASET INTRO

- The goal of this project is to, to predict which web pages would attract high, long-lasting traffic with dataset having 25 features.
- Dataset contains split train (5916 rows x 27 columns) and test (1479 rows x 26 columns)
- The problem has been solved using classification models (logistic regression and Multi-layer Perceptron classifier)
- Workflow : 1) Data Preprocessing on train-test data by merging (Dealing With Missing Values, One Hot Encoding, outlier detction) 2) NLP preprocessing on text data columns(lowering case, punctuation marks removal, hyperlink removal, stripping, stop words removal, stemming, tokenization) 3) using BOW and TF-IDF for vectorizing text data in one solution 4) using GloveVec for vectorizing text data in other solution 5)Train test split 6) Applying model and predicting output probabilities 7) Stacking of models.

DATA PRE-PROCESSING ON NON-TEXT DATA

- 1) Dealing With Missing Values: In our dataset we have missing values for alchemy_category_score columns we replaced it with mean of column as column values were continuous and mean & median was approx. Same. Replaced missing values for isNews and isFrontPageNews columns we replaced it with mode of columns respectively.
- 2) Dropping least related columns: URL column was dropped as it was least related also, webdescription column has URL data. 'Framebased' column has all 0 values hence it was dropped.
- 3) One-Hot encoding: One-Hot encoding has been done for the alchemy_category column as it was categorical data and has limited unique values.
- 4) Outlier detection and removal.
- 5) Feature scaling to normalize data, using robust scalar.

NLP DATA PRE-PROCESSING ON TEXT DATA

- 1) Converting text data to lower case, removing hyperlinks, removing punctuations.
- 2) Tokenizing, Stripping non-alphabetic characters, removing stop words.
- 3) Stemming(taking only root words).
- 4) Using TF-IDF technique to quantify words in a set of documents and computed score for each word to signify its important in the document and corpus.
- 5) Also used, GloVe Embedding for feature vectorization.

ADDITIONAL EDA & PREPROCESSING PERFORMED

- 1) Outlier detection and removal using IQR.
- 2) Used GloveVec embedding to represent text data in form of vectors.

  => Glove is a pre-trained model by Stanford.

  => It represents a word in a multi-dimensional space of dimensions such as 50,100,200 and 300.

  => It represents words such that words with similar meaning lie close to each other in the multi-dimensional space.

- 3) Stacking allows to reap the benefits of different models by combining their individual predictions. We used SVM(RBF kernel) and MLP in base models and logistic regression as final(meta) model as of now.

GLOVE AND STACKING : WALKTHROUGH OUR BEST SUBMISSION

- Basic EDA and Preprocessing :

1) Dealing with missing values :

=> Checking for NA values : Given dataset contains no NA values.

=> Checking for '?' Values : alchemy_category, alchemy_category_score, isNews, isFrontPageNews these columns contains '?' Categorical features are replaced with mode and for numerical data is replaced with mean.

=> 'framebased' column is dropped as it contains all 0 values.

2) ONE HOT Encoding :

=> one hot encoding has used for 'alchemy_category_score' this categorical column.

3) Outlier detection and removal using IQR.

GLOVE AND STACKING : WALKTHROUGH OUR BEST SUBMISSION

- NLP text data Preprocessing :
- 1) Converting text data to lower case, removing hyperlinks, removing punctuations.
- 2) Tokenizing, Stripping non-alphabetic characters, removing stop words.
- 3) Lemmitization.
- 4) Used GloVe Embedding for text data representation.

GLOVE AND STACKING : WALKTHROUGH OUR BEST SUBMISSION

- Model implementation :

1) Stacking allows to reap the benefits of different models by combining their individual predictions using higher level. We used SVMRBF and MLP in base models and logistic regression as final(meta) model as of now.

2) for MLP we have tuned hyper parameters alpha and max_iter other than default values.

3) We have got public score for kaggle submission on test data : 0.89536

MODELS USED AND KAGGLE SCORE

Preprocessing	Model	Kaggle score (if submitted)
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, BOW vectorizing	logistic regression	Public score: 0.79160
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, TF-IDF vectorizing	logistic regression	Public score: 0.87392
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, TF-IDF vectorizing	Multi-layer Perceptron classifier	Public score: 0.82188
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, TF-IDF vectorizing	SVM	Public score: 0.86601

MODELS USED AND KAGGLE SCORE

Preprocessing	Model	Kaggle score (if submitted)
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloveVec Embedding	logistic regression	Public score: 0.89222
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloveVec Embedding	SVMRBF	Public score: 0.88928
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloveVec Embedding	Multi-layer Perceptron classifier	Public score: 0.89238
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloveVec Embedding	Random Forest	Public score: 0.86086
Dealing With Missing Values, One Hot Encoding, NLP data preprocessing, GloVe Embedding	Stacking	Public score: 0.89536