Python Version: Python 3.6.4 :: Anaconda, Inc.

Used Modules: Scikit-learn, Keras, pandas, NumPy

Command - line: python train.py <*file\_name\_train*>

File train.py:

==========

Description:

1. Capable of fitting model on the given data set(\*.csv)
2. Prediction Pipeline Includes:
   1. Missing Value Treatment
   2. Drop Black Listed Candidates - Dates, Currency, Telephone, VAT Number
   3. Compute Pattern Transformers:
      1. Number of Digits
      2. Number of Separators
      3. Is\_StrictlyDigit
      4. Is\_endWithSpecialCharacter
      5. Length of the Candidate
   4. Compute Context Transformers:
      1. Number of Characters Text
      2. Number of Digits Text
      3. Number of Separators Text
      4. Is\_Email\_exists
      5. Number of Spaces
      6. POS tagged Trasformers:
         1. Number of Adverb/Adjective
         2. Number of Nouns
         3. Number of Verbs
         4. Bag of words feature vectors
         5. Saves the Vocabulary to “vocab.txt “
   5. Compute Location Transformers:
      1. Threshold Line\_nr
      2. Threshold page\_nr
3. Trains a NN classifier model:
   1. Outputs 5-Fold CV Accuracy score
   2. Awareness : Brier Score
   3. Results(\*.csv) file containing predictions and confidence levels
   4. Save the model with weights (\*.h5)

Command - line: python run.py <*file\_name\_test*>

File run.py:

=========

Description:

1. Capable of leveraging the model on the corresponding test set.
2. Inputs - Model without “gold” column
3. Computes all the features designed in the train.py. Feature Vector Computation includes
   1. Missing Value Treatment
   2. Pattern Transformers
   3. Context Transformers
   4. Location Transformers
   5. Load Vocabulary from the train.py
   6. BoW feature Vectors
4. Loads saved model.
5. Perform Predictions
6. Populates Index, Predictions, Confidence Level to a csv file

Description of the Next Steps:

========================

1. I observed my Pattern Transformers incorrectly classify few positive class samples as irrelevant. (Less than 100 Samples). Improve robustness in the pattern transformer.
2. Understand what are custom features and Implement the Optional flag to trigger custom features functionality.
3. Shed More light on Natural Language Processing techniques. Such as, Named Entity Recognition, TF-IDF, word2Vec Algorithms.
4. Frequency Distribution Analysis on the Negative class Texts.
5. Try to capture the contextual meaning of the word, which will help in building an optimal vocabulary set.
6. Build Generic Location transformers
7. A fixed vocabulary set which can be reused across matching domains.

Note: I implemented Under Sampling and SMOTE techniques to deal with the imbalance data set problem. Both the techniques resulted in bad performances on higher feature dimensions.

Results:

=======

| **Data Frame** | **Dummy Classifier Accuracy** | **CV Accuracy Score** | **Brier Loss(Pos\_label=1)** |
| --- | --- | --- | --- |
| Contract Number | 92.00 | 94.46 | 0.042 |
| Claim Number | 94.67 | 96.53 | 0.024 |