Categorization of News Articles based on Industry

Objective: Classify the News articles based on Industry: Business, Sports, Tech, Entertainment, and Politics

Data Source: BBC

Tool used: Python

Solution:

1. Using glob library of Python, I created a data set from the text documents.
   The dataset has 2225 rows and 2 columns: ["Doc Text", "Topic"]
2. **Text Preprocessing:**
   a. Removed all punctuations and new line characters: "\n\n" using regular expressions "[^\w\s]" and "\n\n" respectively.
   b. Removed single characters resulting from the removal of punctuations using the expression "\s+[a-zA-Z]\s+".
   c. Replaced single characters from the beginning of the document with a single space using "\^[a-zA-Z]\s+".
   d. Replaced one or more spaces with a single space using regular expression "\s+".
   e. Converted all the words to lowercase and removed all the stop words using nltk library
   f. Stemmed all the words using PorterStemmer () of nltk library
3. **Tf-Idf Matrix:**
   Converted the dataset into a tf-idf matrix using TfidfVectorizer () 0f sklearn.feature_extraction.text
4. Split the dataset into training, validation, and test datasets.
5. **Data Modeling**:
   I. Built general classifiers, such as Naïve Bayes and Decision tree, with all variables. But they performed badly. I got mean F1 test scores of 0.87 and 0.83 respectively
   II. Feature Engineering:
      a. Used Feature selection techniques, such as SelectKBest, RFECV, feature_importances_, etc. But they took a lot of time to run and classifiers built on the output of some techniques performed badly.
   III. Feature Extraction:
      a. Performed PCA (n_component =3) on the dataset and developed Naïve Bayes and Decision Tree classifiers. The mean test F1 scores are 0.80 and 0.90.
      b. Performed Linear Discriminant Analysis (LDA) to find best discriminants and Random Forest Classifier based on the variables given by LDA has mean "train" F1 score of 0.47.
   IV. Ensemble Methods with Hyperparameter Tuning and PCA:
      a. Built Random Forest classifier with all variables of the training data set and obtained an accuracy of 0.96 and mean test F1 score of 0.96.
      b. Using Bayesian optimization to tune hyperparameters of both PCA and KNN, I observed an accuracy of 0.95 and mean F1 test score of 0.95.
      c. Did Bayesian optimization to tune hyperparameters of both PCA and Random Forest and got an accuracy of 0.97 and mean F1 test score of 0.96.
      d. Performed PCA (n_component =3) on the dataset and used Random search to tune the hyper parameters of Light Gradient Boosting which resulted in the accuracy of 0.93 and mean test F1 score of 0.93
      e. Performed Bayesian optimization to tune hyperparameters of both PCA and Light Gradient Boosting and obtained an accuracy of 0.95 and mean test F1 score of 0.95.

Results:

Classifier	Mean test F1 Score	Accuracy
Naive Bayes with all variables	0.87	0.88
Decision Tree with all variables	0.84	0.85
Naive Bayes with SelectKBest (k=20)	0.67	0.69
Decision Tree with SelectKBest (k=20)	0.72	0.72
Random Forest with LDA and Bayesian Optimization	Got low training value	
Naïve Bayes with PCA (n_components = 3)	0.80	0.80
Decision Tree with PCA (n_components = 3)	0.90	0.90

KNN with Bayesian Optimization (on PCA as well)	0.95	0.95
Decision Tree with Bayesian Optimization (on PCA as well)	0.93	0.93
Random Forest with all variables	0.96	0.96
Random Forest with Bayesian Optimization (on PCA as well)	0.96	0.97
Light Gradient Boosting with PCA and Random Search	0.93	0.93
Light Gradient Boosting with Bayesian Optimization (on PCA as well)	0.95	0.95

Conclusion:

a. I obtained the best accuracy of 0.97 and F1 score of 0.96 through Bayesian Optimization on both PCA and Random classifier.

Observations:

a. Performing hyperparameter tuning on both PCA and Ensemble methods/Standard classifiers gave me higher accuracy and F1 score, but took more time, than just using hyperparameter optimization on Ensemble methods/standard Classifiers.
b. When the variables are collinear, it’s better to use PCA rather than LDA because with LDA gives only maximum of min (n_classes, n_features-1) variables.
c. Classifiers, when tuned with Bayesian Optimization, don’t give same results on every instance
d. Sometimes, the standard classifiers, such as Decision Tree, Naïve Bayes, KNN, and SVM, would give best results when tuned using Bayesian Optimization
e. When the dataset is large, it’s better to use Light GB if time is the constraint or else you can use Random Forest or XGBoost which give better results
f. When the dataset has high dimensionality, it’s better to use feature extraction techniques rather than feature selection techniques. You can use feature selection techniques like wrapper methods (RFECV), but RFECV take a lot of time.