1. LANGUAGES IDENTIFICATION

Table Of Contents

Introduction

Language Detection is one of the most important issue as we have reached to the point where the world has become a global village in order to stay connected with the world there is a need to understand the language of the text thereon.Since it is difficult for human brain to understand and learn all the languages , we needed computational systems to be build which can perform the tasks .In this following report we are going to demonstrate some of the important factors. We will then use the simple techniques for demonstrating our data .

Purpose Of The Project

The main objective of the project is to find the purpose is the detect the languages and find the most relevant predictive models

Data Cleaning And Preprocessing

- So the first step while dealing with any kind of the data is to do the preprocessing is to clean the dataset and in case of the text data the first thing is to remove the special charectors because they add no value to the data and adding onto that they induce noise to the algorithms and after that we removed the blank spaces and the no meaning words .
- After the removal of all the unnecessary information we were left out with the formatted data with 78160 rows and 3 columns .
- After getting the data which is fit for feature engineering we used count vectorizer to extract the features in order to save the coding time

After removing the unnecessary information we used vectoriser the characters instead of words because character is the smallest unit so doing that we can have more meaningful insights to extract the features and divided the dataset into test and train in order to save the coding time and as a result we extracted 97 features to represent for this database .

features: [' ', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u', 'v', 'w', 'x', 'y', 'z', '\x80', '\x81', '\x82', '\x83', '\x84', '\x85', '\x86', '\x87', '\x88', '\x89', '\x8a', '\x8b', '\x8c', '\x8d', '\x8e', '\x8f', '\x90', '\x91', '\x92', '\x93', '\x94', '\x95', '\x96', '\x97', '\x98', '\x99', '\x9a', '\x9b', '\x9c', '\x9d', '\x9e', '\x9f', '\xa0', '¡', '¢', '£', '¤', '¥', '¦', '¨', '©', 'ª',

'¬', '\xad', '®', '¯', '°', '±', '²', '³', '´', 'µ', '¶', '·', '¸', '¹', 'º', '¼', '½', '¾', '¿', 'â', 'ã', 'ä', 'å', 'è', 'î', 'ï', 'ð', 'ñ'] Len features: 97

DATASET DIVISION

So we splitted the dataset into 80/20 for training and testing

Modelling

We use sckitlearn package for data modelling and we created the model using multinomial naive bayes and multinomial naive bayes is the best modelling technique which we could have used as bayesian learning technique is common in NLP as word count for text classification.

Classification Techniques

- **Multinomial Naive Bayes-**Multinomial Naive Bayes algorithm is the probabilistic learning method which is based on the bayes theorem and predicts the tag of a text such as piece of newspaper article or email. So we created the model and divided into test and train pieces and then in order to check the accuracy of the model we performed the accuracy test and the result was 92.3 % which was fairly good but as we were looking at the best classification technique so we decide to test other methods .
- **Random Forest-**

  It is a classification method which comprises of number of decision trees with various subsets of the given dataset and the result is deduced by taking up the average to improve the predictive accuracy. Random forest does not rely on one tree but it relies on the prediction of the majority of the trees . Just as we expected the accuracy we got from the random forest classifier approach was 94.13% which was much better than the naive bayes classifier approach. But we figured out that there was still some room for improving the accuracy of the model which was done using optimising hyperparameters or what we commonly call as hyperparameter tuning. Where the hyperparameters which we used were n_estimators (which specifies the number of trees in the forest) .So once we tuned the hyperparameters , we got the optimal parameters were max_features: log2,n_estimators as 200, we performed the random forest classifier and as a result the accuracy of the model came out to be 94.37%.

- **Support Vector Machine-** SVM is a technique which works by mapping the data to a high –dimensional feature where the data points can be categorised and are not linearly separable. And the data is divided where the separatorc ould be drawn as a hyperplane.And the result of the accuracy came out to be 89.8% which was way lesser than the random forest classifier and the naive bayes classifier
- **Gaussian Naive bayes –**So the difference between the naive bayes and the gaussian naive bayes is that here each class follows gaussian distribution
- **Logistic Regression-** Logistic regression is a kind of statistical model which in its basic form takes the help of a logistic function in model and a binary dependent variable.

Model evaluation

We tried Random Forest, SVC, Gausian, Logistic regression algorithms and we saw that Random Forest give us the best accuracy In additionally, we use Grid search CV to optimize the parametters of the algorithm. It give me n_estimators=300, max_features= 'log2' is the best parametters. Which gave us the best accurate results

```
Optimal paras:  {'max_features': 'log2', 'n_estimators': 300}
```

	model_name	accuracy
0	RandomForestClassifier	0.941006
1	SVC	0.898260
2	GaussianNB	0.802968
3	LogisticRegression	0.932971

```
[0.94079284 0.9434638  0.9470957  0.9484226  0.937408  ]
[0.91432225 0.91385265 0.91613357 0.92116209 0.913344  ]
[0.81675192 0.82693784 0.82970829 0.81928713 0.81888   ]
[0.93842711 0.93789972 0.9392912  0.94176745 0.930752  ]
```

Limitations Faced While Classification

Disadvantages: **Training Time** The time for training was very fast with the classification but to find the optimal parametters for random forest algorithm, it took more time (near 30 mins). The system needs to learn online or realtime training and prediction, we deduced that Gaussian or Logistic Regression is

better because the accuracy is not much different but the time for training is very fast.

Further Improvements

We could have used other aproaches like RNN, reinforcement learning in order to learn and remember the characters, to train the model. So it is not good to use this way because the parameter model was very large, the time training would be so long and hot GPU,.. But we think that this method will be our better solution.

References

[1] A. Simões, J.J. Almeida, and S.D. Byers, Language identification: a neural network approach. (2014)
https://www.researchgate.net/publication/290102620_Language_identification_A_neural_network_approach