kanhyanamnaresh · 2 days ago · 3 min read

Text Classifier Using Naive Bayes Classifier.

Updated: a few seconds ago

Text Classification
Tokenization
Word2Vec
K-nearest Neighbor
Naive Bayes
Classifier
Deep Learning
Conditional Random Field (CRF)
Support Vector Machine (SVM)
Contextualized Word Representations
A Survey
RMDL
HDLTex
FastText
Logistic Regression
Deep Belief Network (DBN)
Random Forest
Recurrent Neural Networks (RNN)
Recurrent Convolutional Neural Networks (RCNN)
Term Frequency-Inverse Document Frequency
Hierarchical Deep Learning for Text
Convolutional Neural Networks (CNN)

Image Reference: https://medium.com/text-classification-algorithms/text-classification-algorithms-a-survey-a215b7ab7e2d
This blog post talks about the Naive Bayes Classifier. It is applied mostly for text classification. This is simple classifier which classifies based on the probabilities of events.

Naive Bayes Algorithm

Naive bayes classifiers are collection of classification that are based on Bayes Theorem. Naive Bayes is more technically referred to as posterior probability. The classification model can be used to handle binary as well as multiple classifications. It works well in text classification problems such as spam filtering and classification reviews as positive or negative.
Bayes' Theorem is used to find the probability of an event occurring given the probability of another event that has already occurred.
Bayes theorem mathematically represented as

$$P(A|B)=P(A)xP(B|A)/P(B)$$

Image Source:https://www.freecodecamp.org/news/bayes-rule-explained/

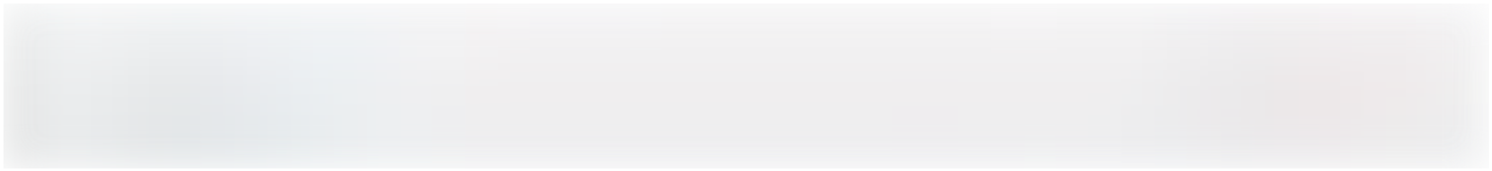
In this blog we perform Text classification using Naive Bayes classifier. The dataset we used here is Ford Senetence Classification dataset available on kaggle. The dataset consists of various types labelled as Responsibility, Requirement, Skill, SoftSkill, Experience and Education based on the appropriate description provided.

Importing libraries:

The below provided libraries are imported which are useful for reading input files , vocabulary building and plotting the graphs.

Loading Dataset:

First we read the "train_data.csv" dataset using pandas library into a data frame by providing the column names we want to read in to data frame.
The below code is used to read data into data frame from "train_data.csv".



Output:

Cleaning The Dataset:

As the data contains puntuations, we need to clean the data and remove punctuation in the "New_Sentence" column of the dataframe.
The below code is used to remove the punctuations from the data.

Output:

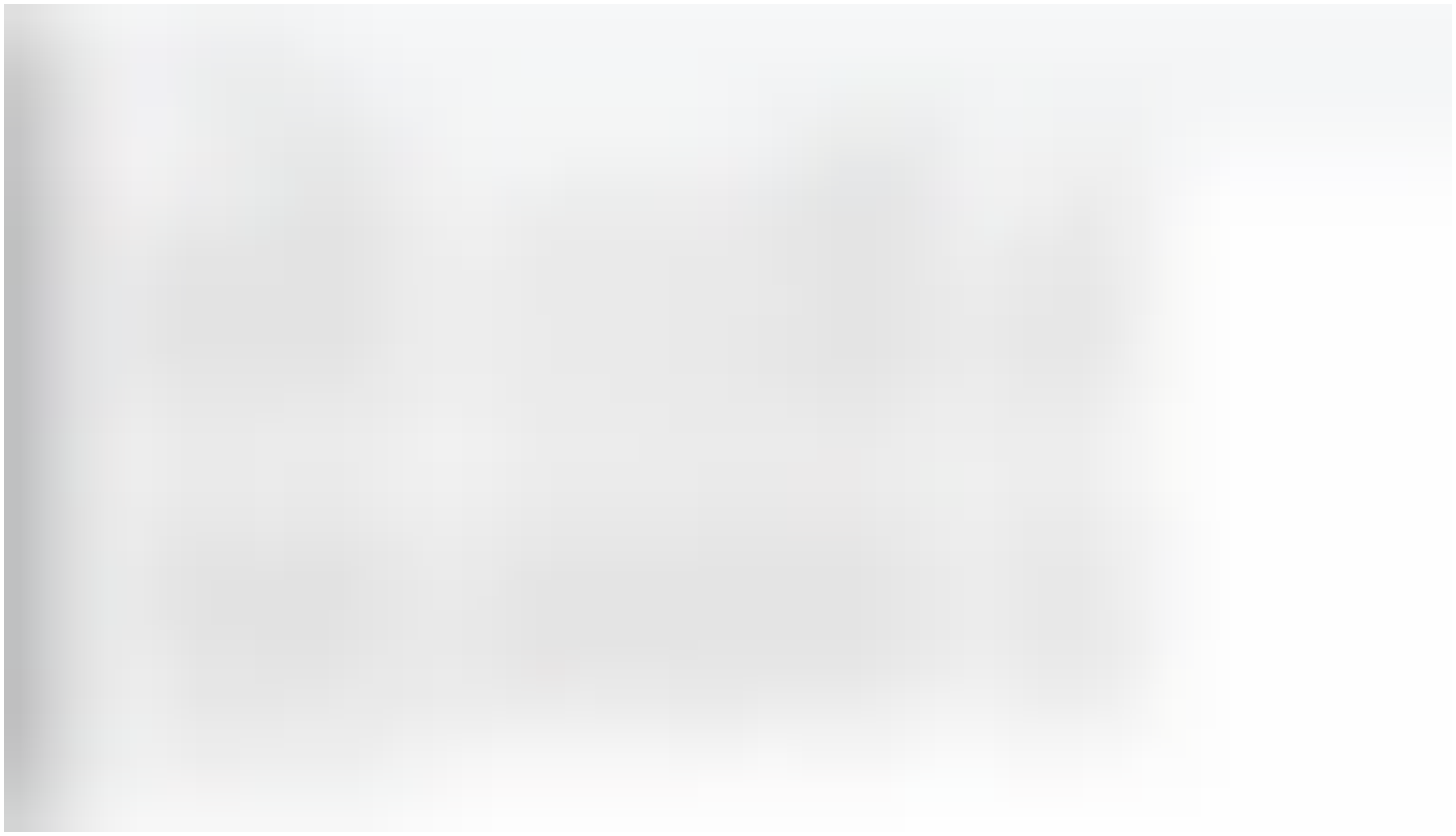
The below code provides the information of the data frame such as number of columns, count of items in each column and type of the object of each column.

Split Data:

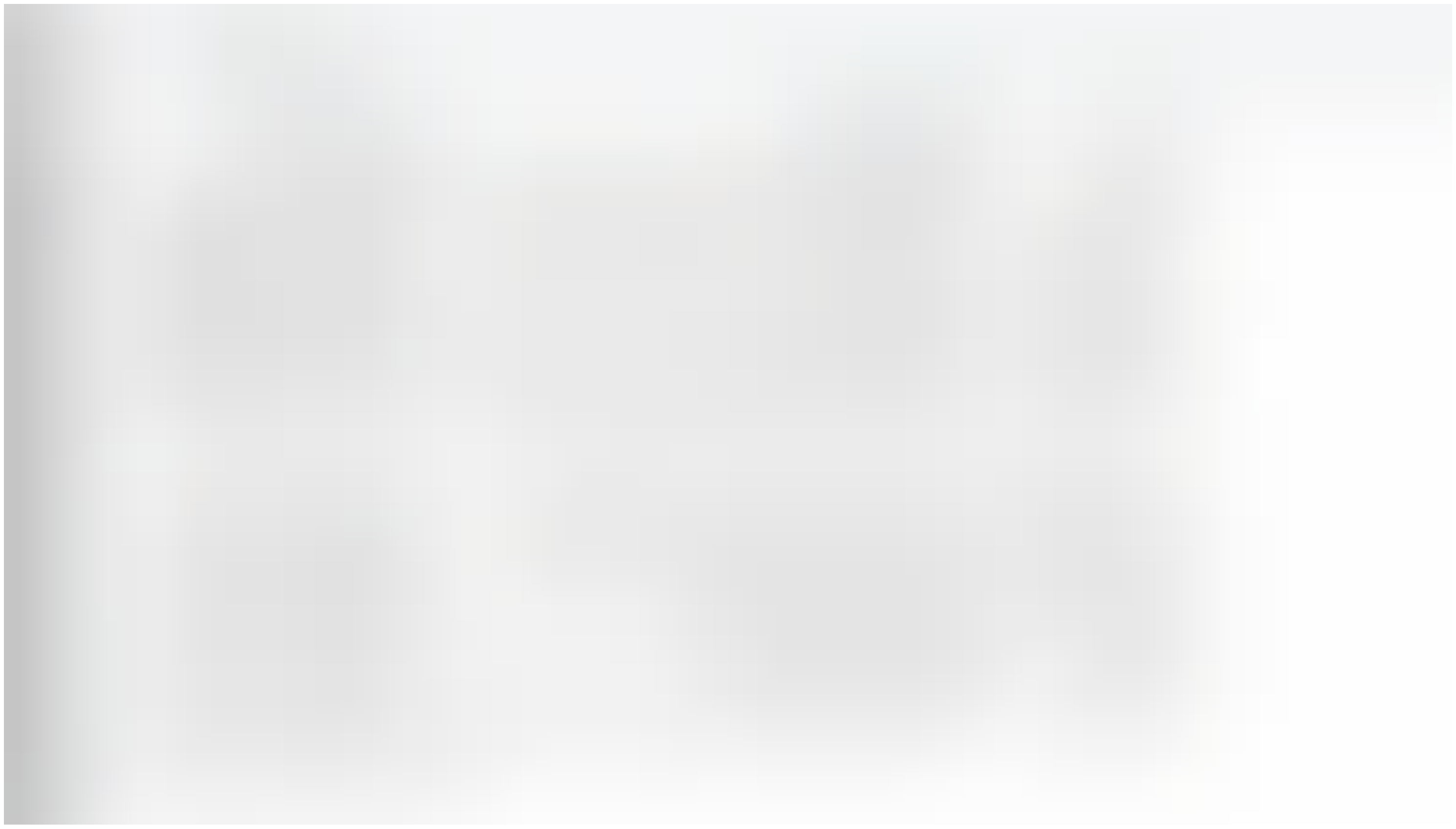
After cleaning the data, we split the data into train, dev and test sets. He we split 80% as train set , 10% as dev set and 10% as test set. Below code is used to split the dataset.

The below snippet show the count of train data after split.

The below snippet show the count of dev data after split.



the below snippet show the count of test data after split.



Building the word cloud using the train dataset.

Output:

Building Vocabulary:

We consider the count of the words which are occurring frequently and later we omit the words which are occurring less than 5 times. We perform this omitting because these words are not important in the prediction of the text class.

Below code calculates the probability of word

Conditional probability

Output:

Accuracy on Dev dataset:

Once we build the dataset we calculate the accuracy of the dev dataset.

Accyracy of the model is :50%

Effect of Smoothing:

What is smoothing?(Referenced from source)

Laplace smoothing is smoothing technique that helps tackle the problem of zero probabilityin the Naive Bayes algorithm. Using Laplace smoothing, we can represent P(w'|positive) as

Reference:Image Source

Here

alpha represents smoothing parameter

K represents the number of dimensions(features) in the data, and

N represents the number of reviews positive

Testing the model:

After performing smoothing on the data we test the model we build using the test dataset.
The accuracy of the model achieved is:

Accuracy of the model is :50.30%

Contribution:

I have used the string library for removing punctuation from dataset instead of using hardcoded values or regular expression. For the initial split of the data i took 65% of the data for train data set, remaining 35% for dev and test sets. With this experiment it has given accuracy of 45% only. After changing the split ratio to *0% for train data set and remaining 20% for dev and test set the accuracy given by the model is 50.30%.
Used smoothing technique for the classifier to avoid zero probability problem.

Challenges and Solution:

Faced challenges for building Naive Bayes Text classifier wihout using "scikit" library.
Building word vocabulary to calculate probability of words and implementing laplace smoothing on the dataset.
Using the below references i am able to build the text classifier model without using scikit library and also implementing smoothing on the data.

References:

https://www.kaggle.com/code/yaswanthjk/sentiment-analysis/notebook
Bayes Theorem : https://towardsdatascience.com/text-classification-using-naive-bayes-theory-a-working-example-2ef4b7eb7d5a
Smoothing: https://towardsdatascience.com/laplace-smoothing-in-na%C3%AFve-bayes-algorithm-9c237a8bdece
https://tejnir1421.wixsite.com/pradyumnateja/text-classifier-using-naive-bias
https://www.kaggle.com/code/yaswanthjk/sentiment-analysis/notebook
https://courses.cs.washington.edu/courses/cse446/20wi/Section7/naive-bayes.pdf
https://www.kdnuggets.com/2020/07/spam-filter-python-naive-bayes-scratch.html
https://machinelearningmastery.com/implement-resampling-methods-scratch-python
https://sebastianraschka.com/Articles/2014_naive_bayes_1.html
https://pandas.pydata.org/docs/user_guide/index.html#user-guide
https://www.kdnuggets.com/2020/07/spam-filter-python-naive-bayes-scratch.html
https://machinelearningmastery.com/implement-resampling-methods-scratch-python/