ABSTRACT

For the previous couple of years, text mining has been gaining significant importance. Since Knowledge is now available to users through sort of sources i.e. electronic media, digital media, medium , and lots of more. thanks to huge availability of text in numerous forms, tons of unstructured data has been recorded by research experts and have found numerous ways in literature to convert this scattered text into defined structured volume, commonly referred to as text classification. specialize in full text classification i.e. full news, huge documents, long length texts etc. is more prominent as compared to the short length text. During this paper, we've discussed the text classification process, classifiers, and various feature extraction methodologies like, naive bayes, logistic regression, etc. but beat the context of short texts i.e. news classification supported their headlines. Existing classifiers and their working methodologies are being compared and results are presented effectively. We used naive Bayes , support vector machines, and logistic regression and we did comparisons.

ACKNOWLEDGMENT

Without taking help of other people is it not possible to complete this project. So we have the opportunity to say thank them to all who have helped us directly or indirectly to make the project successful.

Firstly, we would like to thank our guide Prof. Dhaval Bhoi. We are grateful to him prolonged interest in our work and excellence guidance. He has been a constant source of motivation to us by providing us with suitable media performance, a platform to show our potential and a chance to prove our skills by the way of project development.

We are grateful to Dr. Ritesh Patel., HOD of the Computer Department for allowing us to take projects at cspit. We are sincerely thankful to him for his time to time and valuable guidance during the training period. We are also thankful to my family members to provide mental strength during my project preparation.

With Sincerely

Table Of Content

List of Figures

List of Tables

INTRODUCTION

This report on classification algorithms puts an overview of different classification methods commonly used in data mining techniques with different principles. Classification is a technique which categorizes data into a distinct number of classes and in turn labels are assigned to each class. The main target of classification is to identify the class to launch new data by analysis of the training set by seeing proper boundaries. In a general way, predicting the target class and the above process is called classification.

For instance, the hospital management records the patient's name, address, age, previous history of the patient's health to diagnosis them, this helps to classify the patients. They can be characterized into two phases: a learning phase and evaluation phase. Learning phase models the approach base don a training data whereas the evaluation phase predicts the output for the given data. We could find their applications in email spam, bank loan prediction, Speech recognition, Sentiment analysis. The technique includes mathematical function f with input X and output Y.

we learn different types of news classification algorithms and we implement each algorithm using python and we get to know which has highest accuracy and which has lowest accuracy. From that we select three algorithms to compare with each other.

Classification?

We use the training dataset to get better boundary conditions which could be used to determine each target class. Once the boundary conditions are determined, the next task is to predict the target class. The whole process is known as classification.

Target class examples:

Analysis of the customer data to predict whether he will buy computer accessories (Target class: Yes or No)
Classifying fruits from features like color, taste, size, weight (Target classes: Apple, Orange, Cherry, Banana)
Gender classification from hair length (Target classes: Male or Female)

Basic Terminology in Classification Algorithms

- **Classifier**: An algorithm that maps the input data to a specific category.
- **Classification model**: A classification model tries to draw some conclusion from the input values given for training. It will predict the class labels/categories for the new data.
- **Feature**: A feature is an individual measurable property of a phenomenon being observed.
- **Binary Classification**: Classification task with two possible outcomes. Eg: Gender classification (Male / Female)
- **Multi-class classification**: Classification with more than two classes. In multi-class classification, each sample is assigned to one and only one target label. Eg: An animal can be a cat or dog but not both at the same time.
- **Multi-label classification**: Classification task where each sample is mapped to a set of target labels (more than one class). Eg: A news article can be about sports, a person, and location at the same time.

Applications of Classification Algorithms

- Email spam classification
- Bank customers loan pay willingness prediction.
- Cancer tumor cells identification.
- Sentiment analysis
- Drugs classification
- Facial key points detection
- Pedestrians detection in an automotive car driving.

Types of Classification Algorithms

- *Linear Classifiers*
  - Logistic regression
  - Naive Bayes classifier
- *Support vector machines*
  - Least squares support vector machines
- *Kernel estimation*
  - k-nearest neighbor
- *Decision trees*
  - Random forests

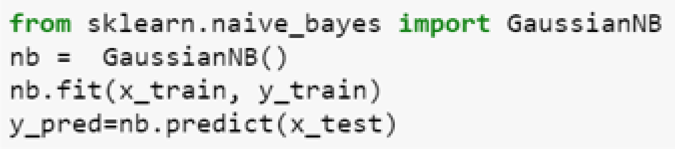
Explanation of Algorithms

1. Naive Bayes

Definition: Naive Bayes algorithm based on Bayes' theorem with the assumption of independence between every pair of features. Naive Bayes classifiers work well in many real-world situations such as document classification and spam filtering.

Advantages: This algorithm requires a small amount of training data to estimate the necessary parameters. Naive Bayes classifiers are extremely fast compared to more sophisticated methods.

Disadvantages: Naive Bayes is is known to be a bad estimator.



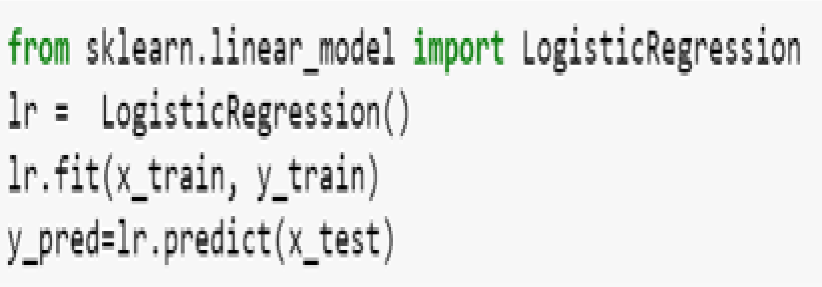
```
from sklearn.naive_bayes import GaussianNB
nb =  GaussianNB()
nb.fit(x_train, y_train)
y_pred=nb.predict(x_test)
```

Figure: 2.1(Naive Bayes)

2. Logistic Regression

Definition: Logistic regression is a machine learning algorithm for classification. In this algorithm, the probabilities describing the possible outcomes of a single trial are modelled using a logistic function.

Advantages: Logistic regression is designed for this purpose (classification), and is most useful for understanding the influence of several independent variables on a single outcome variable.Disadvantages: Works only when the predicted variable is binary, assumes all predictors are independent of each other, and assumes data is free of missing values.



```
from sklearn.linear_model import LogisticRegression
lr =  LogisticRegression()
lr.fit(x_train, y_train)
y_pred=lr.predict(x_test)
```

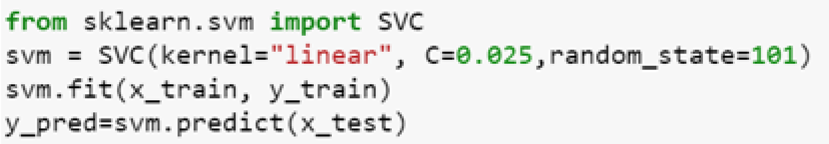
Figure:2.2 (Logistic Regression)

3. Support Vector Machines

Definition: Support vector machine is a representation of the training data as points in space separated into categories by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

Advantages: SVM Classifiers offer good accuracy and perform faster prediction compared to Naïve Bayes algorithm. They also use less memory because they use a subset of training points in the decision phase. SVM works well with a clear margin of separation and with high dimensional space

Disadvantages: The algorithm does not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation.



```
from sklearn.svm import SVC
svm = SVC(kernel="linear", C=0.025,random_state=101)
svm.fit(x_train, y_train)
y_pred=svm.predict(x_test)
```

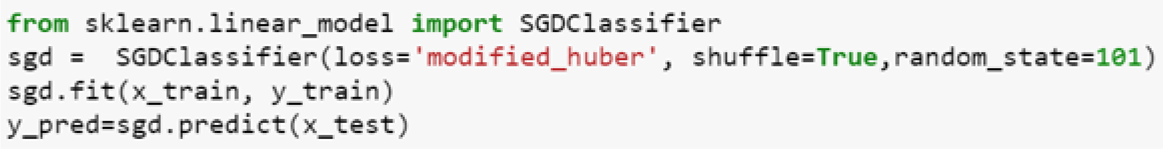
Figure: 2.3 (Support Vector Machines)

4. Stochastic Gradient Descent

Definition: Stochastic gradient descent is a simple and very efficient approach to fit linear models. It is particularly useful when the number of samples is very large. It supports different loss functions and penalties for classification.

Advantages: Efficiency and ease of implementation.

Disadvantages: Requires a number of hyper-parameters and it is sensitive to feature scaling.



```
from sklearn.linear_model import SGDClassifier
sgd =  SGDClassifier(loss='modified_huber', shuffle=True,random_state=101)
sgd.fit(x_train, y_train)
y_pred=sgd.predict(x_test)
```

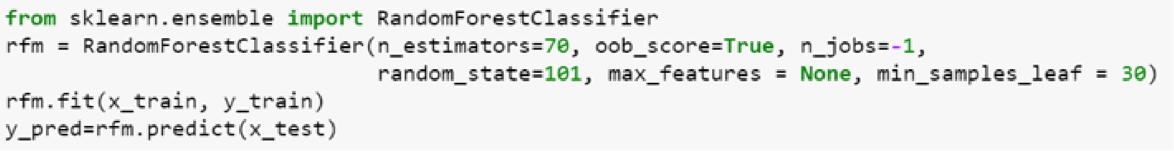
Figure: 2.4 (Stochastic Gradient Descent)

5. Random Forest

Definition: Random forest classifier is a meta-estimator that fits a number of decision trees on various sub-samples of datasets and uses average to improve the predictive accuracy of the model and controls over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement.

Advantages: Reduction in over-fitting and random forest classifiers is more accurate than decision trees in most cases.

Disadvantages: Slow real time prediction, difficult to implement, and complex algorithm.



```
from sklearn.ensemble import RandomForestClassifier
rfm = RandomForestClassifier(n_estimators=70, oob_score=True, n_jobs=-1,
                             random_state=101, max_features = None, min_samples_leaf = 30)
rfm.fit(x_train, y_train)
y_pred=rfm.predict(x_test)
```

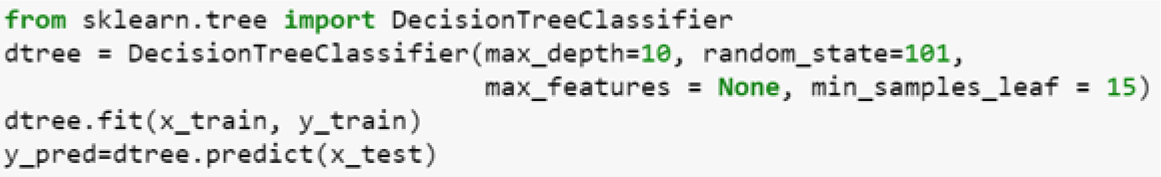
Figure2.5 (Random Forest)

6. Decision Tree

Definition: Given a data of attributes together with its classes, a decision tree produces a sequence of rules that can be used to classify the data.

Advantages: Decision Tree is simple to understand and visualise, requires little data preparation, and can handle both numerical and categorical data.

Disadvantages: Decision tree can create complex trees that do not generalise well, and decision trees can be unstable because small variations in the data might result in a completely different tree being generated.



```
from sklearn.tree import DecisionTreeClassifier
dtree = DecisionTreeClassifier(max_depth=10, random_state=101,
                               max_features = None, min_samples_leaf = 15)
dtree.fit(x_train, y_train)
y_pred=dtree.predict(x_test)
```

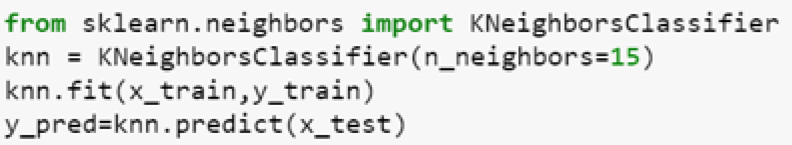
Figure:2.6 (Decision Tree)

7. K-nearest neighbors

Definition: Neighbours based classification is a type of lazy learning as it does not attempt to construct a general internal model, but simply stores instances of the training data. Classification is computed from a simple majority vote of the k nearest neighbours of each point.

Advantages: This algorithm is simple to implement, robust to noisy training data, and effective if training data is large.

Disadvantages: Need to determine the value of K and the computation cost is high as it needs to computer the distance of each instance to all the training samples.



```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=15)
knn.fit(x_train,y_train)
y_pred=knn.predict(x_test)
```

Figure:2.7 (K-Nearest Neighbour)

Accuracy Table

Link: https://github.com/Yukta1Khatsuria

In the above link we implement each algorithm and then we get this accuracy.

Classification Algorithm	**Accuracy**	**F1-score**
logistic Regression	84.60%	0.6337
naive Bayes	80.11%	0.6005
stochastic Gradient Descent	82.20%	0.5780
K-Nearest Neighbours	83.56%	0.5924
Decision Tree	84.23%	0.6308
Random Forest	84.33%	0.6275
Support Vector Machine	84.09%	0.6145

Table : 1.1 (List of Algorithm Accuracy)

from above table we can see that logistic regression and naive bayes hase highest accuracy and support vector machine has lowest accuracy among rest of algorithm so we use this three algorithm in our project. this is also recommended algorithm in real scenarios.

ALGORITHMS

1. Naive Bayes Algorithm

Classification Workflow

Whenever you perform classification, the first step is to understand the problem and identify potential features and label. Features are those characteristics or attributes which affect the results of the label. For example, in the case of a loan distribution, bank manager's identify customer's occupation, income, age, location, previous loan history, transaction history, and credit score. These characteristics are known as features which help the model classify customers.

The classification has two phases, a learning phase, and the evaluation phase. In the learning phase, classifier trains its model on a given dataset and in the evaluation phase, it tests the classifier performance. Performance is evaluated on the basis of various parameters such as accuracy, error, precision, and recall.

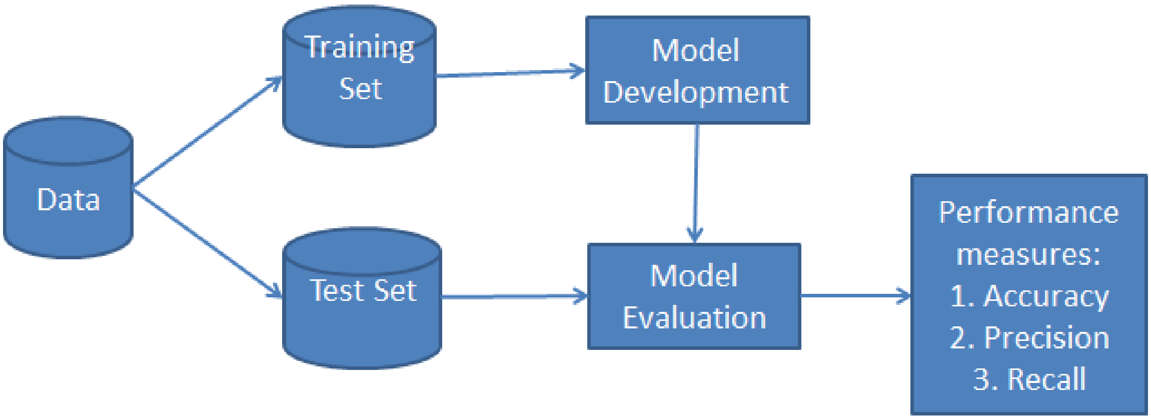


Figure 3.1(Workflow of naive bayes)

What is Naive Bayes Classifier?

Naive Bayes is a statistical classification technique based on Bayes Theorem. It is one of the simplest supervised learning algorithms. Naive Bayes classifier is the fast, accurate and reliable algorithm. Naive Bayes classifiers have high accuracy and speed on large datasets.

Naive Bayes classifier assumes that the effect of a particular feature in a class is independent of other features. For example, a loan applicant is desirable or not depending on his/her income, previous loan and transaction history, age, and location. Even if these features are interdependent, these features are still considered independently. This assumption simplifies computation, and that's why it is considered as naive. This assumption is called class conditional independence.

$$P(h|D) = \frac{P(D|h)P(h)}{P(D)}$$

- **P(h):** the probability of hypothesis h being true (regardless of the data). This is known as the prior probability of h.
- **P(D):** the probability of the data (regardless of the hypothesis). This is known as the prior probability.
- **P(h|D):** the probability of hypothesis h given the data D. This is known as posterior probability.
- **P(D|h):** the probability of data d given that the hypothesis h was true. This is known as posterior probability.

Advantages

- It is not only a simple approach but also a fast and accurate method for prediction.
- Naive Bayes has very low computation cost.
- It can efficiently work on a large dataset.
- It performs well in case of discrete response variable compared to the continuous variable.
- It can be used with multiple class prediction problems.
- It also performs well in the case of text analytics problems.
- When the assumption of independence holds, a Naive Bayes classifier performs better compared to other models like logistic regression.

Disadvantages

- The assumption of independent features. In practice, it is almost impossible that model will get a set of predictors which are entirely independent.
- If there is no training tuple of a particular class, this causes zero posterior probability. In this case, the model is unable to make predictions. This problem is known as Zero Probability/Frequency Problem.

2. Logistic Regression Algorithm

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.

It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.

Linear Regression Equation:

$$y = \beta 0 + \beta 1X1 + \beta 2X2 + \ldots\ldots + \beta nXn$$

Where, y is a dependent variable and x1, x2 ... and Xn are explanatory variables.

Sigmoid Function:

$$p = 1/1 + e^{-y}$$

Apply Sigmoid function on linear regression:

$$p = 1/1 + e^{-(\beta 0+\beta 1X1+\beta 2X2.....\beta nXn)}$$

Properties of Logistic Regression:

- The dependent variable in logistic regression follows Bernoulli Distribution.
- Estimation is done through maximum likelihood.
- No R Square, Model fitness is calculated through Concordance, KS-Statistics.

Sigmoid Function

The sigmoid function, also called logistic function, gives an ‘S’ shaped curve that can take any real-valued number and map it into a value between 0 and 1. If the curve goes to positive infinity, y predicted will become 1, and if the curve goes to negative infinity, y predicted will become 0. If the output of the sigmoid function is more than 0.5, we can classify the outcome as 1 or YES, and if it is less than 0.5, we can classify it as 0 or NO. The outputcannotFor example: If the output is 0.75, we can say in terms of probability as: There is a 75 percent chance that patient will suffer from cancer.

$$f(x) = \frac{1}{1 + e^{-(x)}}$$

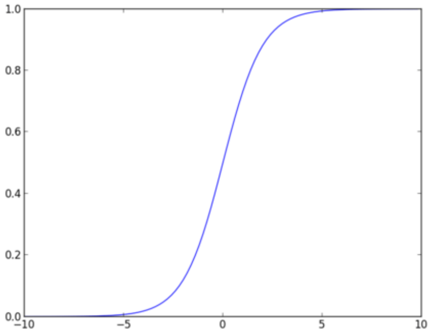


Figure 3.2(Flow of sigmoid function)

Types of Logistic Regression:

- **Binary Logistic Regression**: The target variable has only two possible outcomes such as Spam or Not Spam, Cancer or No Cancer.
- **Multinomial Logistic Regression:** The target variable has three or more nominal categories such as predicting the type of Wine.
- **Ordinal Logistic Regression:** the target variable has three or more ordinal categories such as restaurant or product.

Advantages

Because of its efficient and straightforward nature, doesn't require high computation power, easy to implement, easily interpretable, used widely by data analyst and scientist. Also, it doesn't require scaling of features. Logistic regression provides a probability score for observations.

Disadvantages

Logistic regression is not able to handle a large number of categorical features/variables. It is vulnerable to overfitting. Also, can't solve the non-linear problem with the logistic regression that is why it requires a transformation of non-linear features. Logistic regression will not perform well with independent variables that are not correlated to the target variable and are very similar or correlated to each other.

3. Support Vector Machines

Generally, Support Vector Machines is considered to be a classification approach, it but can be employed in both types of classification and regression problems. It can easily handle multiple continuous and categorical variables. SVM constructs a hyperplane in multidimensional space to separate different classes. SVM generates optimal hyperplane in an iterative manner, which is used to minimize an error. The core idea of SVM is to find a maximum marginal hyperplane(MMH) that best divides the dataset into classes.

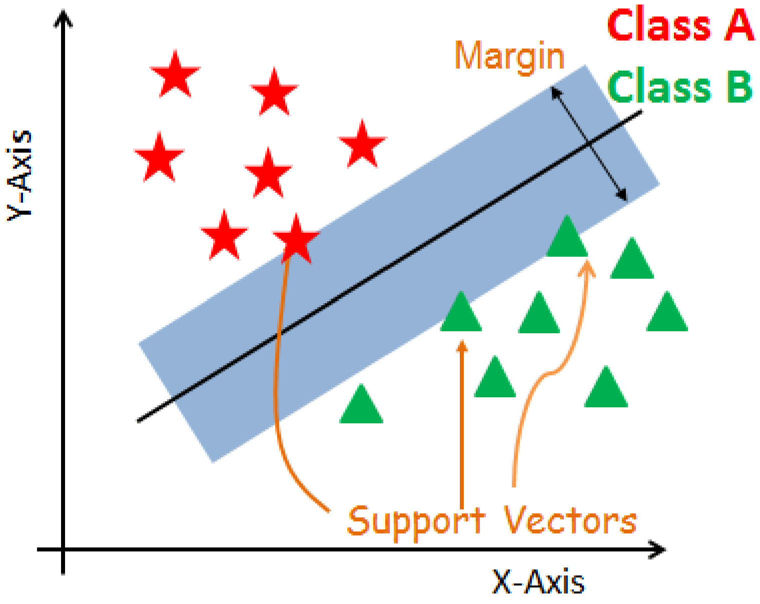


Figure: 3.3 (SVM Graph)

Support Vectors

Support vectors are the data points, which are closest to the hyperplane. These points will define the separating line better by calculating margins. These points are more relevant to the construction of the classifier.

Hyperplane

A hyperplane is a decision plane which separates between a set of objects having different class memberships.

Margin

A margin is a gap between the two lines on the closest class points. This is calculated as the perpendicular distance from the line to support vectors or closest points. If the margin is larger in between the classes, then it is considered a good margin, a smaller margin is a bad margin.

How does SVM work?

The main objective is to segregate the given dataset in the best possible way. The distance between the either nearest points is known as the margin. The objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum marginal hyperplane in the following steps:

1. Generate hyperplanes which segregate the classes in the best way. Left-hand side figure showing three hyperplanes black, blue and orange. Here, the blue and orange have higher classification error, but the black is separating the two classes correctly.
2. Select the right hyperplane with the maximum segregation from the either nearest data points as shown in the right-hand side figure.

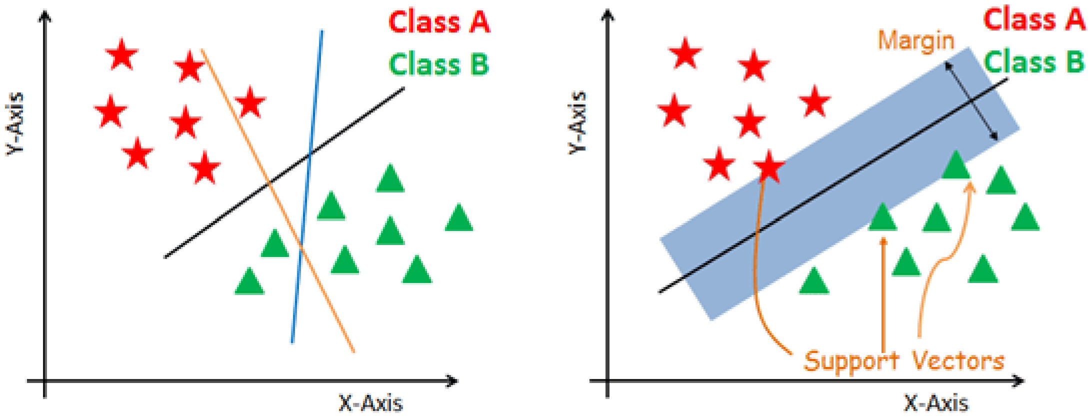


Figure: 3.4 (Working of SVM)

Advantages

SVM Classifiers offer good accuracy and perform faster prediction compared to Naïve Bayes algorithm. They also use less memory because they use a subset of training points in the decision phase. SVM works well with a clear margin of separation and with high dimensional space.

Disadvantages

SVM is not suitable for large datasets because of its high training time and it also takes more time in training compared to Naïve Bayes. It works poorly with overlapping classes and is also sensitive to the type of kernel used.

RESULT

We took one dataset and implemented three algorithms and also compared three algorithms.we got four results which we can see in below figures.

Comparison Matrix

- Accuracy: (True Positive + True Negative) / Total Population
    - Accuracy is a ratio of correctly predicted observation to the total observations. Accuracy is the most intuitive performance measure.
    - True Positive: The number of correct predictions that the occurrence is positive
    - True Negative: The number of correct predictions that the occurrence is negative
- F1-Score: (2 x Precision x Recall) / (Precision + Recall)
    - F1-Score is the weighted average of Precision and Recall used in all types of classification algorithms. Therefore, this score takes both false positives and false negatives into account. F1-Score is usually more useful than accuracy, especially if you have an uneven class distribution.
    - Precision: When a positive value is predicted, how often is the prediction correct?
    - Recall: When the actual value is positive, how often is the prediction correct?

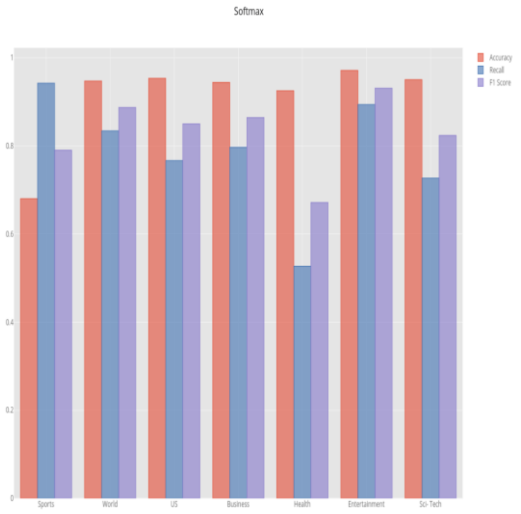


Figure 4.1: Logistic Regression (softmax)

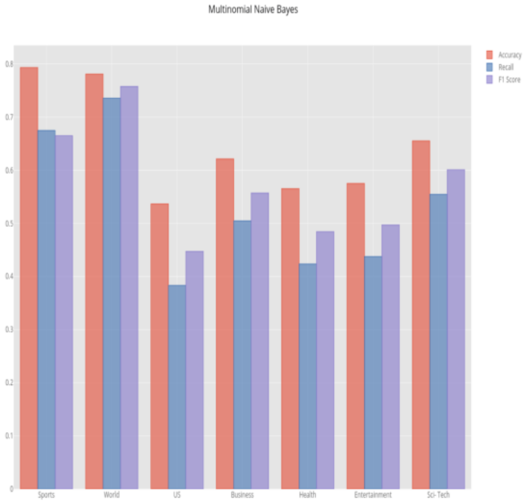


Figure 4.2 : Naive Bayes

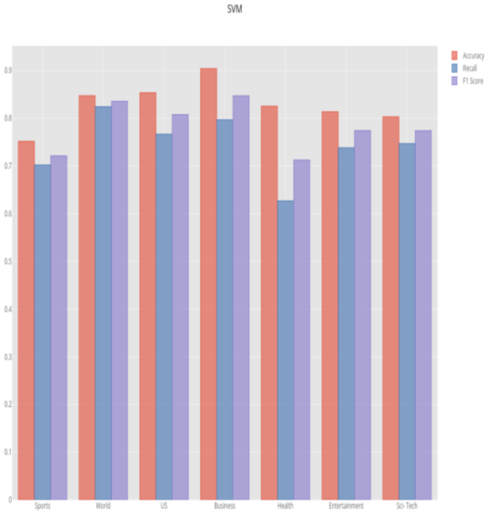


Figure 4.3 : support vector machine (svm)

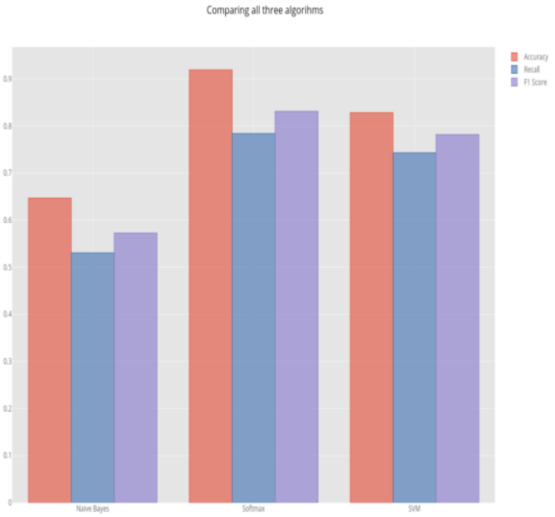


Figure 4.4 : comparing all three algorithm

CONCLUSION

From this we learn different types of classification algorithms and we also get knowledge of how to code into python language. we find different accuracy of algorithms and then from that we selected three algorithms and got to know about working of each algorithm from this project.and also learn how classification algorithms work on the news.

REFERENCES

1. Kaggle News Category Dataset. https://www.kaggle.com/rmisra/ news-category-dataset. Accessed: 2018-10-05.
2. The Huffington Post. https://www.huffingtonpost.com/.
3. T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
4. Keras: The Python Deep Learning library. https://keras.io/.
5. Scikit-learn 0.20.0 documentation. ”1.9 Naive Bayes”. https://scikit-learn.org/ stable/modules/naive_bayes.html.
6. Scikit-learn 0.20.0 documentation. ”1.1.11 Logistic Regression”. https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression.
7. Scikit-learn 0.20.0 documentation. ”1.4 Support Vector Machines”. https://scikit-learn.org/stable/modules/svm.html.
8. Scikit-learn 0.20.0 documentation. ”1.10 Decision Trees”. https://scikit-learn.org/ stable/modules/tree.html.
9. Jürgen Schmidhuber’s page on recurrent neural networks. http://people.idsia.ch/ ~juergen/rnn.html.
10. J. Pennington, R. Socher, and C. D. Manning. Glove: Global Vectors for Word Representation.*EMNLP*, 14:1532–1543, 2014.
11. Scikit-learn 0.20.1 documentation, t-distributed Stochastic Neighbor Embedding. https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html.
12. http://www.iosrjournals.org/iosr-jce/papers/Vol18-issue1/Version-3/D018132226.pdf