Mahatma Gandhi Central University, Bihar

Detecting Fake News using NLP

Group Member -

Md Tanveer Alam

MGCU2019CSIT4008

Ashish Kumar

MGCU2019CSIT4003

Guide By-

Dr. Vipin Kumar

(Asst. Professor)

Deptt. Of CS and IT

Outline

- ❑Introduction
- ❑Objective
- ❑Mean Accuracy
- ❑Sentiment Analysis
- ❑Voting Methods
- ❑Comparison
- ❑Conclusion

Introduction

Fake News is news designed to deliberately spread hoaxes propaganda and disinformation. Fake News stories usually Fake news spread through social media sites Like Facebook , Twitter, WhatsApp.

This is often done to further or impose certain ideas and is often achieved with political agendas. Such news items may contain false and/or exaggerated claims, and may end up being viralized by algorithms, and users may end up in a filter bubble.

About The Project

Well detecting the fake news and Performance Improvement is not a easy task but their can be possible using scikit-learn to make a computer understand the news as there is a vast material text involved.

We got a data set of 6335 article which was tagged as fake or Real ,Data Size trained 29.2 MB

Model was trained using KNN,NB,DT,SVM and NN classifier and after applied Principal Component Analysis and Truncated SVD tell us how well our model performance improve.

Objective

Task-1

To build a model to accurately classify a piece of news as REAL or FAKE. This project of detecting fake news deals with fake and real news. Using scikit-learn, we build a TfidfVectorizer on our dataset. Then, we initialize a Model was trained using KNN,NB,DT,SVM and NN classifier. After compare the mean accuracy score and then Principal Component Analysis and Truncated SVD tell us how well our model performance improve.

Task – 2

we use Sentiment analysis Using SentiWordNet. And compare this result to Task-1 accuracy results.

Task -3

We use voting Methods of the ensemble learning and compare it with Task -1 , Task-2 and Task-3 accuracy results.

The Dataset

The dataset we'll use for this project- we'll call it news.csv. This dataset has a shape of 6335×4. The first column identifies the news, the second and third are the title and text, and the fourth column has labels denoting whether the news is REAL or FAKE.

```
In [3]: df
```

Out[3]:

	Unnamed: 0	title	text	label
0	8476	You Can Smell Hillary's Fear	Daniel Greenfield, a Shillman Journalism Fello...	FAKE
1	10294	Watch The Exact Moment Paul Ryan Committed Pol...	Google Pinterest Digg Linkedin Reddit Stumbleu...	FAKE
2	3608	Kerry to go to Paris in gesture of sympathy	U.S. Secretary of State John F. Kerry said Mon...	REAL
3	10142	Bernie supporters on Twitter erupt in anger ag...	— Kaydee King (@KaydeeKing) November 9, 2016 T...	FAKE
4	875	The Battle of New York: Why This Primary Matters	It's primary day in New York and front-runners...	REAL
...	...	...	...	...
6330	4490	State Department says it can't find emails fro...	The State Department told the Republican Natio...	REAL
6331	8062	The 'P' in PBS Should Stand for 'Plutocratic' ...	The 'P' in PBS Should Stand for 'Plutocratic' ...	FAKE
6332	8622	Anti-Trump Protesters Are Tools of the Oligarc...	Anti-Trump Protesters Are Tools of the Oligar...	FAKE
6333	4021	In Ethiopia, Obama seeks progress on peace, se...	ADDIS ABABA, Ethiopia —President Obama convene...	REAL
6334	4330	Jeb Bush Is Suddenly Attacking Trump. Here's W...	Jeb Bush Is Suddenly Attacking Trump. Here's W...	REAL

6335 rows × 4 columns

Converting Text Model To Features

To analyze and model text after it has been preprocessed, it must be converted into features, Technique may included TF-IDF.

The TfidfVectorizer converts a collection of raw documents into a matrix of TF-IDF features

- **TF (Term Frequency):** The number of times a word appears in a document is its Term Frequency.

- **IDF (Inverse Document Frequency):** Words that occur many times a document
.The TfidfVectorizer converts a collection of raw documents into a matrix of TF-IDF features.

Topic Modelling

Topic Modelling allows us to describe and summarize the documents in a corpus without having to read each individual article. It works by finding pattern in the co-occurrence of words using the frequency of words in each documents

Features in the dataset

- Total number of features available in the dataset is 67351.

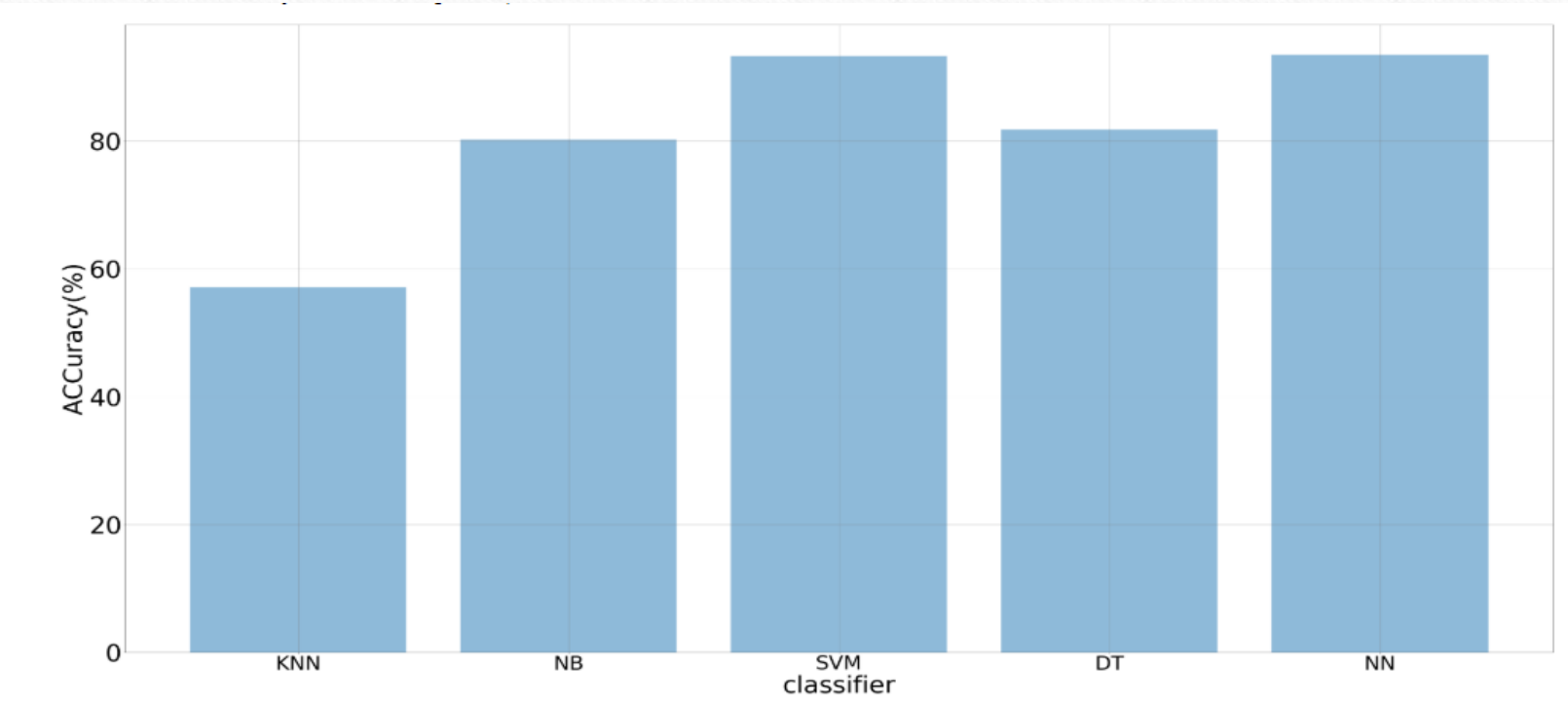
feature name

```
In [12]: print(X.shape)
         print(tfidf_vectorizer.get_feature_names())
```

```
(6335, 67351)
['00', '000', '0000', '000000031', '00000031', '000035', '00006', '0001', '0001pt', '0002', '000bil
lion', '000ft', '000km', '000x', '001', '0011', '002', '003', '004', '004s', '005', '005s', '006',
'00684', '006s', '007', '007s', '008', '008s', '009', '0099', '00am', '00p', '00pm', '01', '010',
'011', '012', '013', '013c2812c9', '014', '015', '016', '018', '01am', '02', '020', '022', '023',
'024', '025', '027', '02714', '028', '02870', '02welcome', '03', '030', '031', '032', '0325', '03
3', '034', '035', '037', '03747', '039', '03eb', '04', '040', '0400', '042', '044', '047', '048',
'049', '04pm', '05', '050', '0509245d29', '052', '053', '056', '058', '06', '0600', '062', '063',
'0640', '066', '068', '06pm', '07', '0700', '071', '075', '0750', '076', '079', '07dryempjx', '08',
'080', '081', '082', '084', '0843', '085', '0851', '089', '0891', '09', '091', '093', '098263', '09
am', '09pm', '0_65b67362bd', '0_jgdktlmn', '0_kvyhphja', '0a_merrill', '0b6njlny5j', '0d', '0dpbdk6
rjd', '0fjjvowyhg8qtskiz', '0h4at2yetra17uxetni02ls2jeg0mty45jrcu7mrzsrpcbq464i', '0hour', '0hq3vb2
giv', '0in', '0jsn6pjkan', '0oeekvljlt', '0pt', '0t5', '0txrbwvobzz4fi5nksw6k5a6cxzbb3juxthmdiz93cb
y8gvrqiypzhajvjnt2', '0womdwalmi', '0x', '10', '100', '1000', '10000s', '10009020', '1000s', '100
1', '10021', '10028', '100515p', '100bn', '100c', '100k', '100m', '100percentfedup', '100s', '100t
h', '101', '1010359', '1013', '1014', '1019', '101st', '102', '1024026', '1026', '102816', '1028360
02', '102k', '102m', '103', '1033', '103rd', '104', '1040', '1040s', '104396002', '1046870142', '10
46870197', '104893', '104th', '105', '1050', '106', '106116001', '1066', '107', '1070', '107th', '1
```

- **Mean Accuracy of KNN,NB,SVM,DT and NN**

Classifier	KNN	NB	SVM	DT	NN
Accuracy (%)	57.048	80.251	93.307	81.910	90.876



KNN- K-nearest neighbor

NB-Naive Bayes

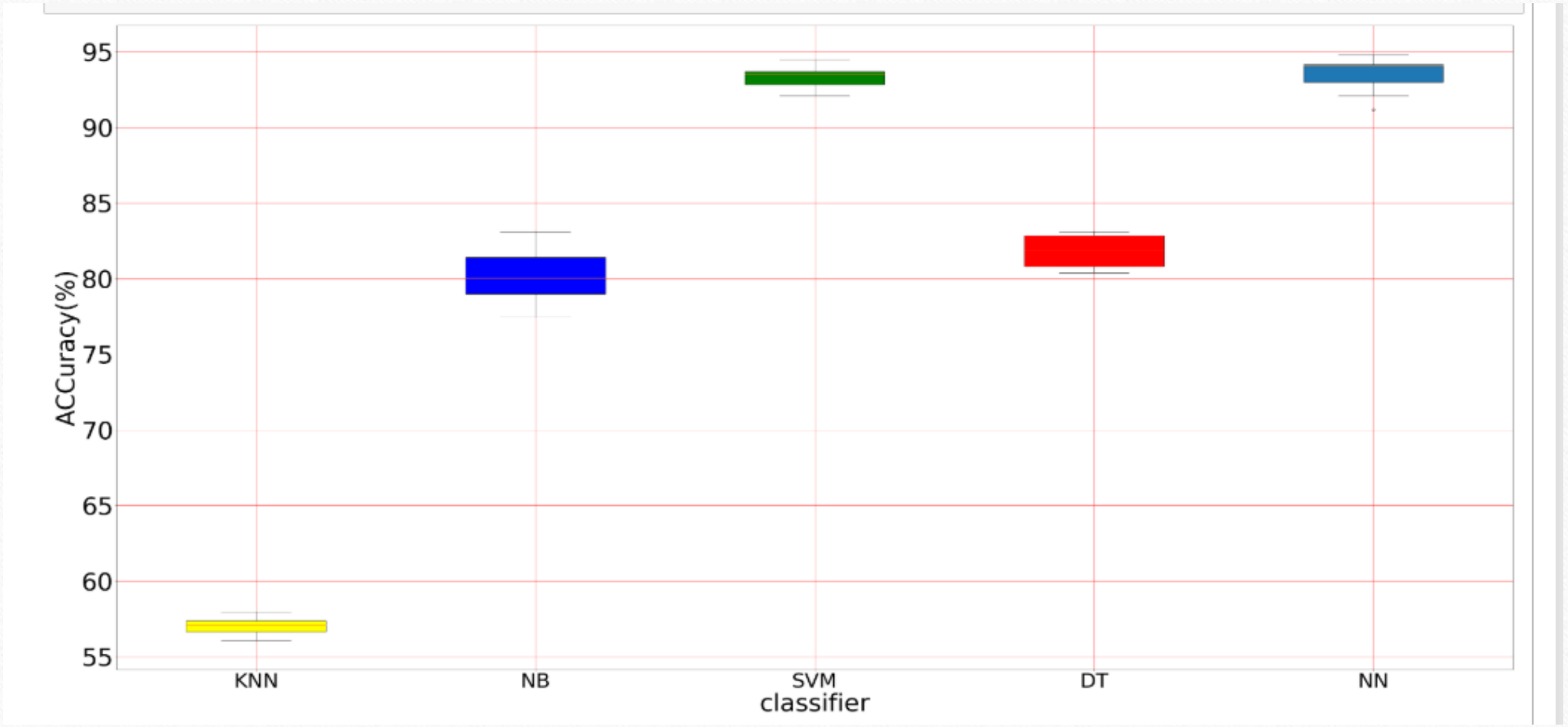
SVM- Support Vector Machine

DT-Decision Trees

NN- Neural Network

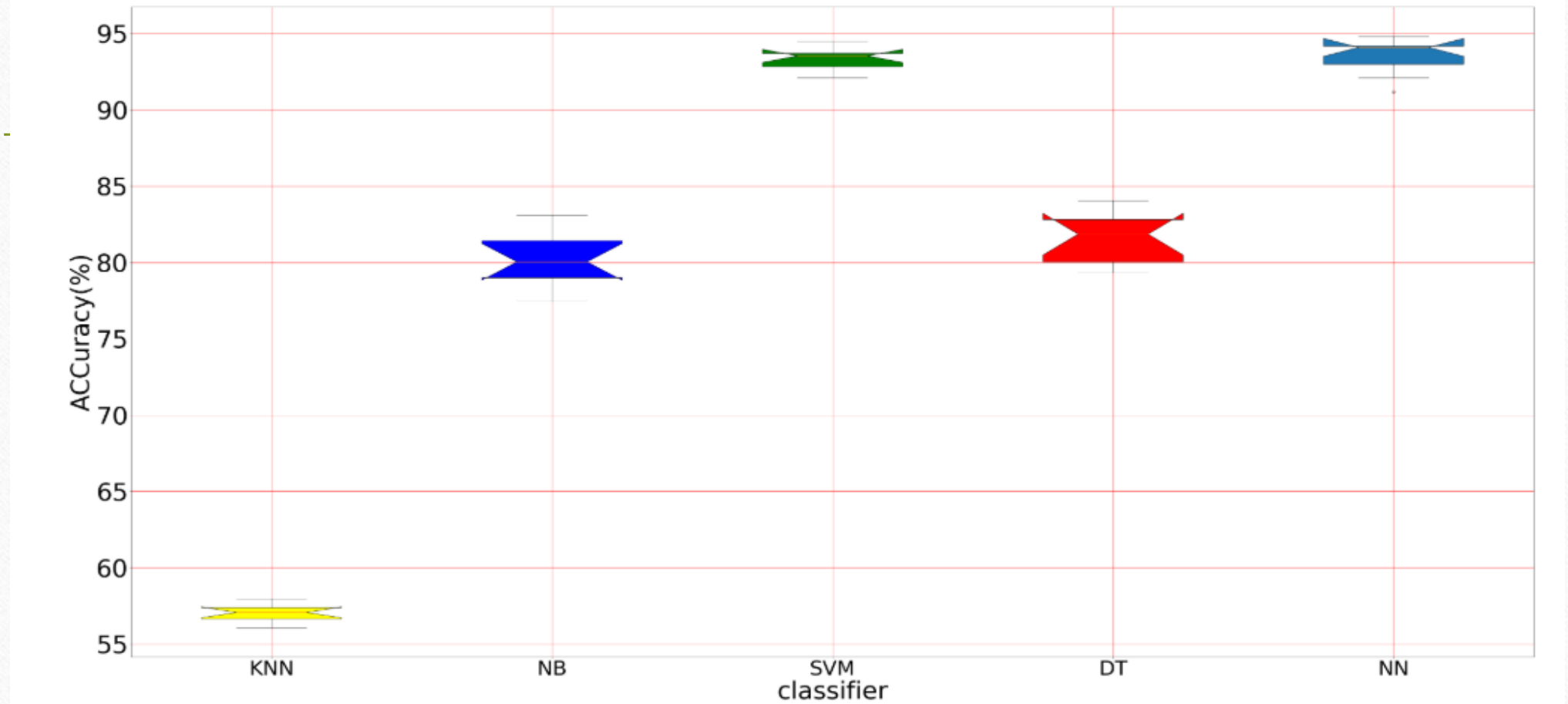
Box-Plot

- Mean accuracy of KNN , NB, SVM , DT and NN

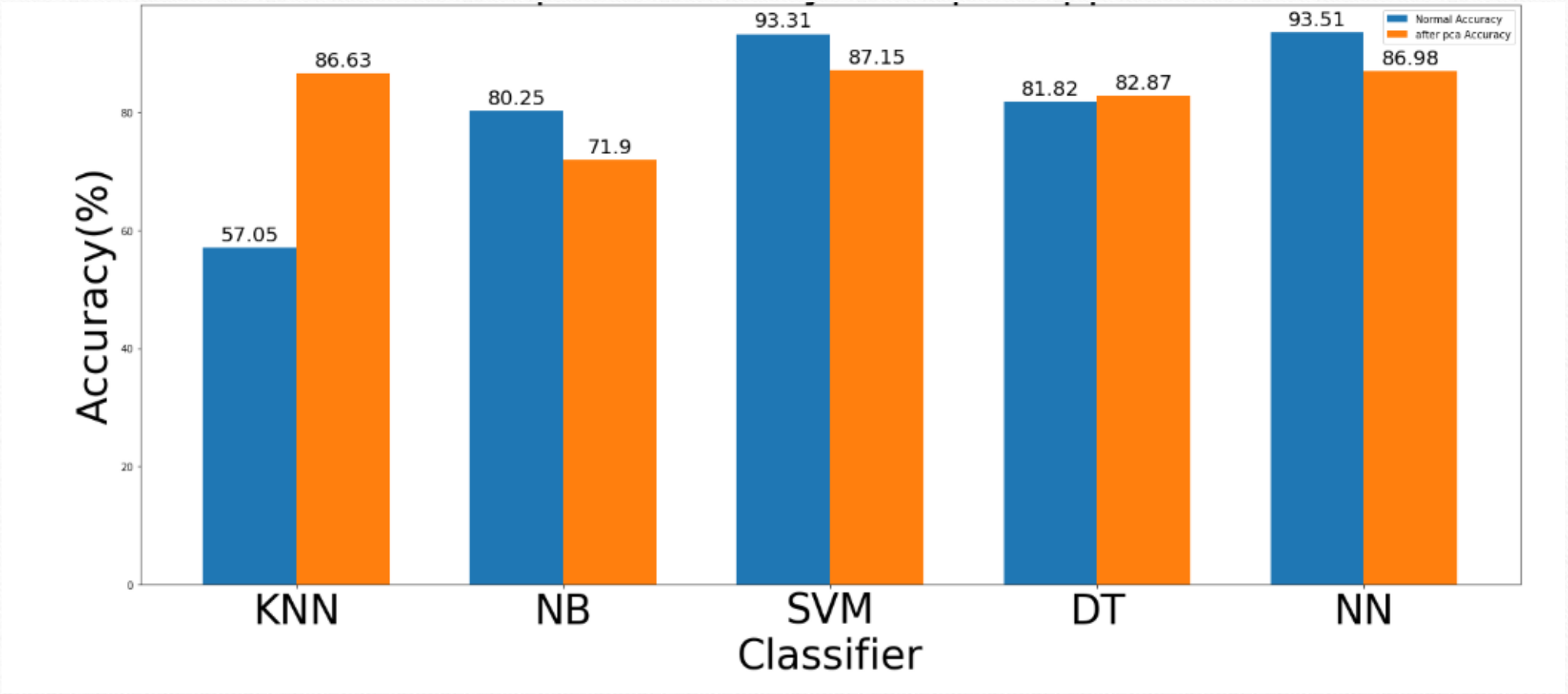


Notched Box-Plot

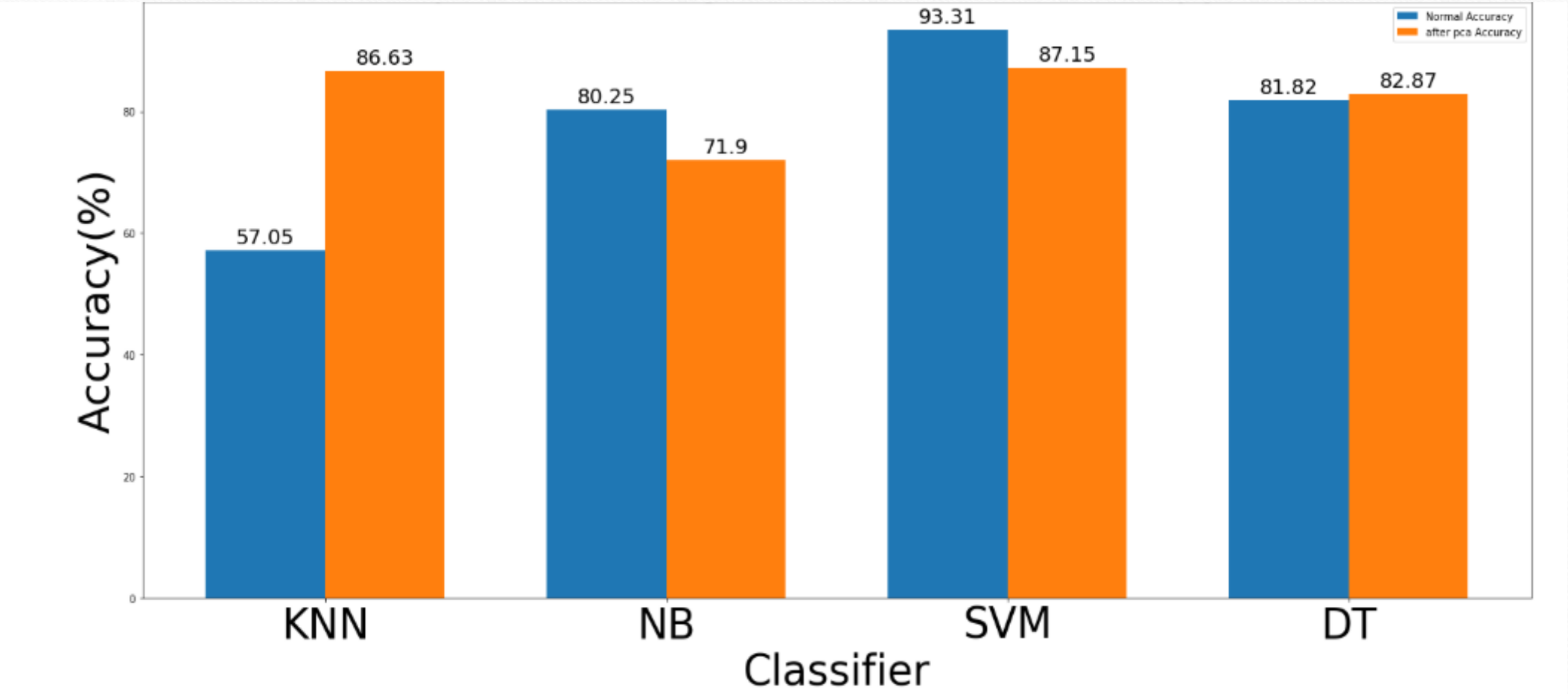
- Mean accuracy of KNN , NB, SVM , DT and NN



- **After applied Principal Component Analysis (PCA) enhance the accuracy in that the classifier KNN and DT**



- After applied Truncated SVD enhance the accuracy in that the classifier KNN and DT



Comparison

Classifier	KNN	NB	SVM	DT	NN
Accuracy	57.04	80.25	93.30	81.91	90.87
Accuracy after PCA applied	86.97	71.90	87.19	82.38	86.33
Accuracy After Truncated SVD	86.63	71.90	87.15	82.87	

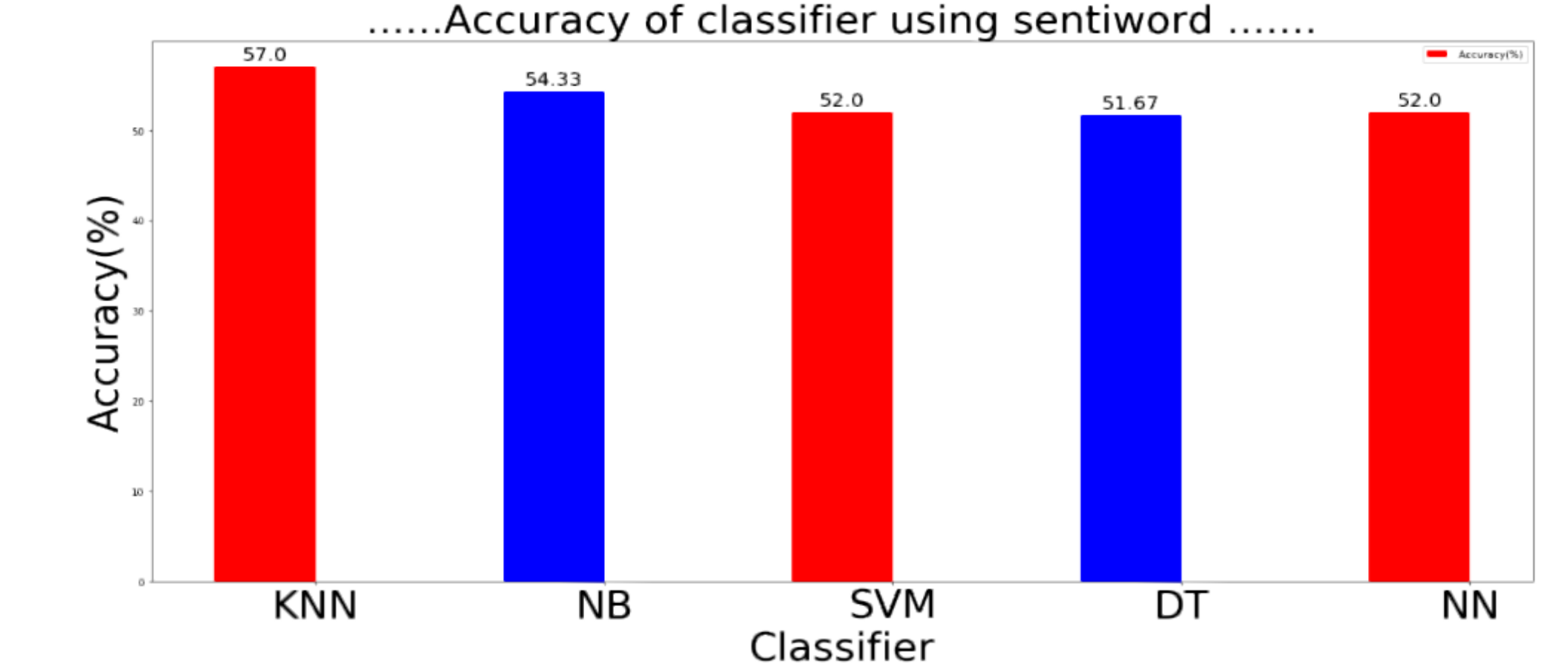
Sentiment analysis using SentiWordNet

```
In [14]: xdataq # first column is positive second is negative third is neutral

Out[14]: [[0.071, 0.074, 0.855],
          [0.079, 0.036, 0.885],
          [0.053, 0.037, 0.91],
          [0.055, 0.056, 0.889],
          [0.064, 0.044, 0.892],
          [0.071, 0.057, 0.872],
          [0.077, 0.054, 0.87],
          [0.04, 0.035, 0.924],
          [0.059, 0.046, 0.896],
          [0.05, 0.034, 0.916],
          [0.067, 0.041, 0.892],
          [0.061, 0.067, 0.871],
          [0.057, 0.022, 0.921],
          [0.048, 0.032, 0.92],
          [0.074, 0.012, 0.914],
          [0.067, 0.044, 0.889],
          [0.057, 0.022, 0.921],
          [0.113, 0.054, 0.833],
          [0.061, 0.042, 0.897],
```

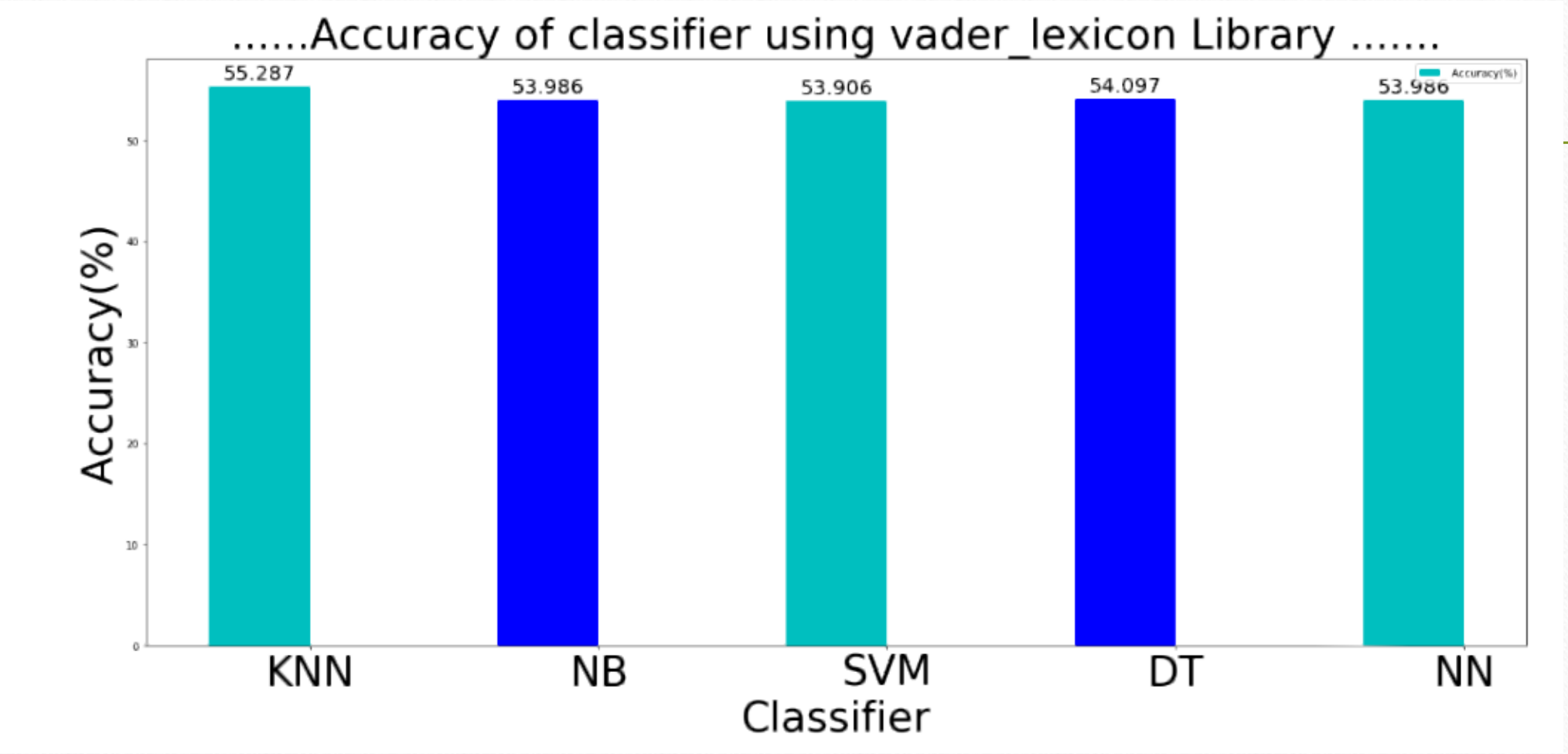
- **The accuracy of classifier using Senti-Words**

KNN is 57.0% ,NB is 54.33%,SVM is 50.0%, DT is 51.67% and NN is 52.0 %.

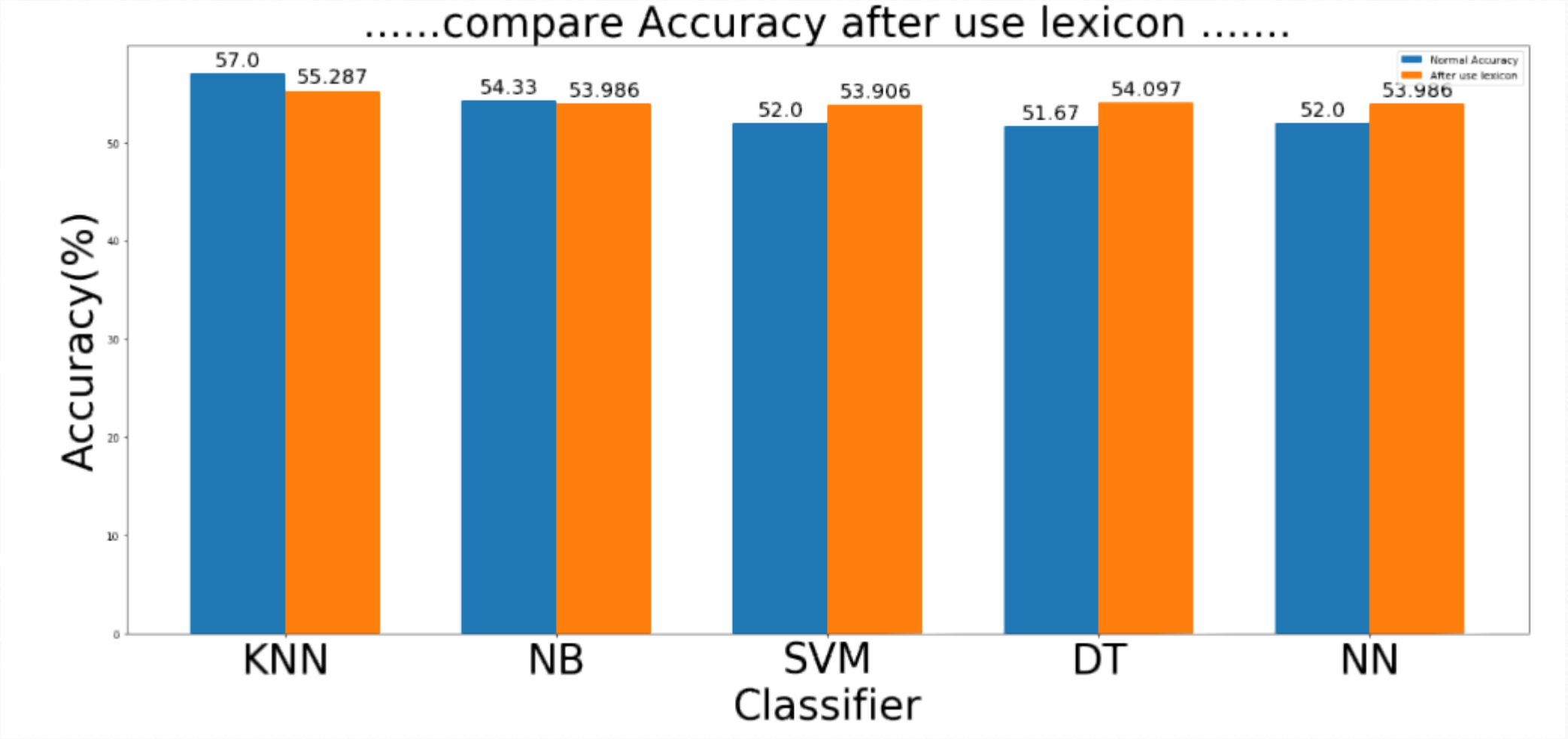


- **The accuracy of classifier using Vader Lexicon Library.**

KNN is 55.28% ,NB is 53.98%,SVM is 53.90%, DT is 54.09% and NN is 5.98 %.



Comparison


.....compare Accuracy after use lexicon .....
Normal Accuracy
After use lexicon
Accuracy(%)
0
10
20
30
40
50
57.0
55.287
54.33
53.986
52.0
53.906
51.67
54.097
52.0
53.986
KNN
NB
SVM
DT
NN
Classifier

Comparison

Classifier	KNN	NB	SVM	DT	NN
Accuracy	57.04	80.25	93.30	81.91	90.87
Accuracy after PCA applied	86.97	71.90	87.19	82.38	86.33
Accuracy After Truncated SVD	86.63	71.90	87.15	82.87	

After Sentiment Analysis :					
Senti Words	57.0	54.33	52.0	51.67	52.0
Lexicon	55.28	53.98	53.90	54.09	53.98

Using Ensemble Methods:

```
In [76]: np.mean(accknn_t1),np.mean(accnb_t1),np.mean(accdt_t1),np.mean(accsvm_t1),np.mean(accnn_t1)

Out[76]: (0.5704818076257967,
          0.8025199216589172,
          0.8140455793602145,
          0.9330707013320974,
          0.9351274039300114)
```

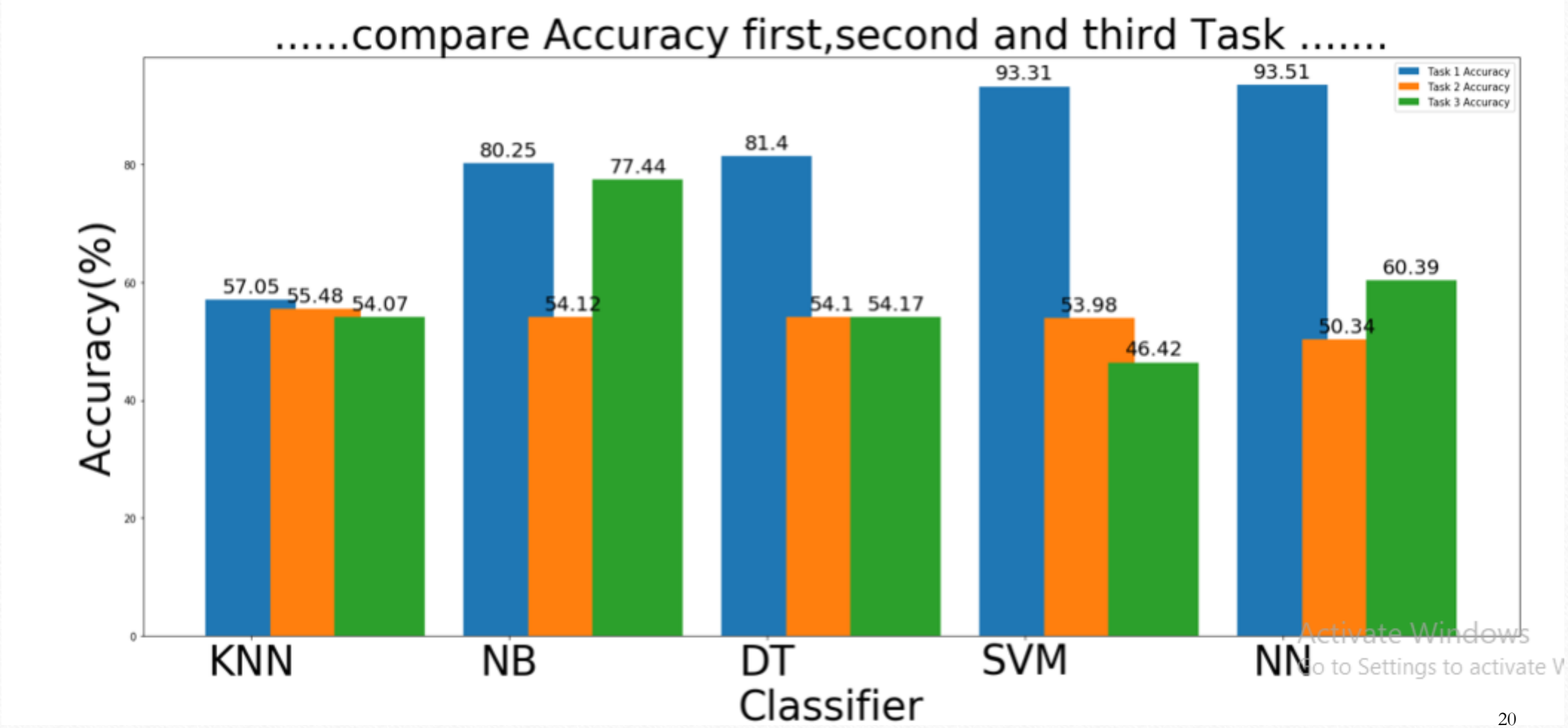
```
In [73]: accknn_t2,accnb_t2,accdt_t2,accsvm_t2,accnn_t2

Out[73]: (55.485, 54.12, 54.104, 53.977, 50.341)
```

```
In [74]: accknn_t3,accnb_t3,accdt_t3,accsvm_t3,accnn_t3

Out[74]: (54.07207, 77.44086, 54.16733, 46.42007, 60.39054)
```

Accuracy after using Voting Methods



Comparison of Task –I , II and III

Classifier	KNN	NB	SVM	DT	NN
Accuracy	57.04	80.25	93.30	81.91	90.87
Accuracy after PCA applied	86.97	71.90	87.19	82.38	86.33
Accuracy After Truncated SVD	86.63	71.90	87.15	82.87	
After Sentiment Analysis :					
Senti Words	57.0	54.33	52.0	51.67	52.0
Lexicon	55.28	53.98	53.90	54.09	53.98
After Ensemble Methods:					
Voting	54.07	77.44	54.17	46.42	60.39

Conclusion

- We got a data set of 6335 article which was tagged as fake or Real ,Data Size trained 29.2 MB.
- Model was trained using KNN,NB,DT,SVM and NN classifier.
- The Performance improvement after applied Principal Component Analysis and Truncated SVD first is KNN and second is Decision Tree. That Corresponding Mean of the accuracy is 86.63% and 82.87%.
- After Sentiment Analysis using Vader Lexicon Library on the classifier we get two best accuracy DT classifier is 54.02 , NN and NB classifier is almost same that is 53.98%.
- After using Voting Methods on all the classifier we get Decision Tree Classifier accuracy is 54.1%.

References

[1] https://sckit-learn.org
[2] https://data-flair.training/blogs/advanced-python-projectdetecting-fake-news/
[3]https://github.com/aesuli/SentiWordNet/blob/master/data/SentiWordNet_3.0.0.txt
[4] https://www.toptal.com/machine-learning/ensemble-methods-machine-learning

Thank You