Bekir Özkan 150319557
Merve Yayın 150116051
Metehan Ertan 150117051
Aleyna Bozacı 150319630

CSE4062
Introduction to Data Science and Analytics
Fake Review Detection

1. Topic

Online product reviews are a fundamental part of the decision-making process for customers as well as vendors on e- commerce. Prior to purchasing services or goods, customers first review the online comments submitted by previous customers. However, these comments can be deceiving as they can be spam or fake. These misleading reviews can cause huge damages to company reputation and services or goods. This is a strong incentive for people to game the system and manipulate user sentiment by posting fake opinions or reviews to promote or to discredit some target products. Our project will tackle this problem of spam/fake reviews by developing a model, which could classify a given review as either fake or genuine, thereby helping to make more meaningful review information available to the customers.

2. Dataset

Deceptive Opinion Spam Corpus v1.4 consist of 4 parts.

- 400 truthful positive reviews from TripAdvisor
- 400 deceptive positive reviews from Mechanical Turk
- 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp
- 400 deceptive negative reviews from Mechanical Turk

The Yelp dataset is a subset of Yelp's businesses, reviews and user data. This dataset is opensource for personal, educational and academic purposes. Dataset is available as a JSON file. This data includes 608,598 reviews for restaurants. This dataset contains reviews from 5,044 restaurants by 260,277 reviewers. In this dataset, there exist 13.22% filtered reviews by 23.91% spammers.

CSE 4062 - Semester 2021

Intro to Data Science and Analytics

Fake Review Detection

Group 2 Members:

Aleyna BOZACI 150319630 - aleynabozacii@gmail.com

Bekir ÖZKAN 150319557 - bekirozkan9698@gmail.com

Merve YAYIN 150116051 - yayinm8@gmail.com

Metehan ERTAN 150117051 - metehan.ertan@hotmail.com

Mustafa USCA 150416054 - uscamustafa64@gmail.com

1- Project Description

Online product reviews are a fundamental part of the decision-making process for customers as well as vendors on e- commerce. Prior to purchasing services or goods, customers first review the online comments submitted by previous customers. However, these comments can be deceiving as they can be spam or fake. These misleading reviews can cause huge damages to company reputation and services or goods. This is a strong incentive for people to game the system and manipulate user sentiment by posting fake opinions or reviews to promote or to discredit some target products. Our project will tackle this problem of spam/fake reviews by developing a model, which could classify a given review as either fake or genuine, thereby helping to make more meaningful review information available to the customers.

We will be using CNN (Convolutional Neural Network), NB (Naïve Bayes), RF (Random Forest) algorithms. We will be improving these methods in order to achieve more efficiency and better results.

2- Dataset

2.1 - Deceptive Opinion Spam Corpus v1.4s. We will be using negative reviews for testing.

- 400 truthful positive reviews from TripAdvisor
- 400 deceptive positive reviews from Mechanical Turk
- 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp
- 400 deceptive negative reviews from Mechanical Turk

2.2 - We extracted a smaller dataset from YELP Dataset as it was too large. We are going to use 800 truthful and 800 deceptive reviews for the learning part and 200 reviews for the testing part.

For our project each column will be a word from review and each row will be that review. Column count changes with the length of review. Deceptive Opinion Spam Corpus v1.4s has 800 rows for learning and 800 rows for testing. For the Yelp dataset there are 1600 rows for learning and 200 rows for learning. Target attribute is truthful or deceptive as boolean.

CSE 4062 - Semester 2021

Intro to Data Science and Analytics

Delivery #2: Exploring your data

Group 2 Members:

Aleyna BOZACI 150319630 - aleynabozacii@gmail.com

Bekir ÖZKAN 150319557 - bekirozkan9698@gmail.com

Merve YAYIN 150116051 - yayinm8@gmail.com

Metehan ERTAN 150117051 - metehan.ertan@hotmail.com

Mustafa USCA 150416054 - uscamustafa64@gmail.com

1- Dataset

1 - Deceptive Opinion Spam Corpus v1.4s (shortly Turk). We will be using negative reviews for testing.

- 400 truthful positive reviews from TripAdvisor
- 400 deceptive positive reviews from Mechanical Turk
- 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp
- 400 deceptive negative reviews from Mechanical Turk

2 - We extracted a smaller dataset from YELP Dataset as it was too large. We are going to use 800 truthful and 800 deceptive reviews for the learning part and 200 reviews for the testing part.

Deceptive Opinion Spam Corpus v1.4s has 800 rows for learning and 800 rows for testing. For the Yelp dataset there are 1600 rows for learning and 200 rows for learning. Target attribute is truthful or deceptive as boolean.

2 - Statistics

2.1- Yelp Train

Number of documents: 1600

Number of words: 148339

Number of characters: 769776

Min word count: 2

Average word count: 92.711875

Max word count: 903

Min character count: 11

Average character count: 481.11

Max character count: 4809

2.2- Yelp Test

Number of documents: 200

Number of words: 21059

Number of characters: 108822

Min word count: 1

Average word count: 105.295

Max word count: 432

Min character count: 4

Average character count: 544.11

Max character count: 2240

2.3- Turk Train

Number of documents: 800

Number of words: 145407

Number of characters: 747588

Min word count: 32

Average word count: 181.75875

Max word count: 797

Min character count: 174

Average character count: 934.485

Max character count: 4048

2.4- Turk Test

Number of documents: 800

Number of words: 97604

Number of characters: 513557

Min word count: 25

Average word count: 122.005

Max word count: 438

Min character count: 147

Average character count: 641.94625

Max character count: 2389

2 - Bag of Words

2.1- Yelp Train

Size of bag of words is 10137 , without stop words size of bag of words is 8326:

With stop words		Without stop words	
Word	Count	Word	Count
the	6440	food	1095
and	5126	place	861
a	3787	good	815
I	3660	great	723

to	2893	service	565
was	2844	restaurant	464
of	2127	like	455
is	1952	pizza	425
it	1822	one	421
for	1689	really	405

2.2- Yelp Test

Size of bag of words is 3310 , without stop words size of bag of words is 2701:

With stop words		Without stop words	
Word	Count	Word	Count
the	928	food	130
and	765	place	126
a	569	good	108
I	535	french	101
was	450	great	97
to	392	brunch	92

of	285	back	89
it	276	like	84
for	251	delicious	69
is	250	chicken	67

2.3- Turk Train

Size of bag of words is 8697 , without stop words size of bag of words is 7480 :

With stop words		Without stop words	
Word	Count	Word	Count
the	7770	room	1850
to	4300	hotel	1755
and	4090	stay	649
I	4033	chicago	636
a	3602	would	572
was	3464	service	470
in	2311	one	464
of	1915	get	384

room	1799	desk	380
for	1701	night	370

2.4- Turk Test

Size of bag of words is 6467 , without stop words size of bag of words is 5437 :

With stop words		Without stop words	
Word	Count	Word	Count
the	5046	hotel	1582
and	3718	room	970
a	2746	chicago	891
to	2407	great	664
was	2360	stay	655
I	2343	staff	473
in	1634	stayed	359
The	1628	rooms	355
of	1376	would	351
hotel	1287	service	349

CSE 4062 - Semester 2021

Intro to Data Science and Analytics

Delivery #3: Exploring your data

Group 2 Members:

Aleyna BOZACI 150319630 - aleynabozacii@gmail.com

Bekir ÖZKAN 150319557 - bekirozkan9698@gmail.com

Merve YAYIN 150116051 - yayinm8@gmail.com

Metehan ERTAN 150117051 - metehan.ertan@hotmail.com

Mustafa USCA 150416054 - uscamustafa64@gmail.com

1- Dataset

1 - Deceptive Opinion Spam Corpus v1.4s (shortly Turk). We will be using negative reviews for testing.

- 400 truthful positive reviews from TripAdvisor
- 400 deceptive positive reviews from Mechanical Turk
- 400 truthful negative reviews from Expedia, Hotels.com, Orbitz, Priceline, TripAdvisor and Yelp
- 400 deceptive negative reviews from Mechanical Turk

2 - We extracted a smaller dataset from YELP Dataset as it was too large. We are going to use 800 truthful and 800 deceptive reviews for the learning part and 200 reviews for the testing part.

Deceptive Opinion Spam Corpus v1.4s has 800 rows for learning and 800 rows for testing. For the Yelp dataset there are 1600 rows for learning and 200 rows for learning. Target attribute is truthful or deceptive as boolean.

2 - Statistics

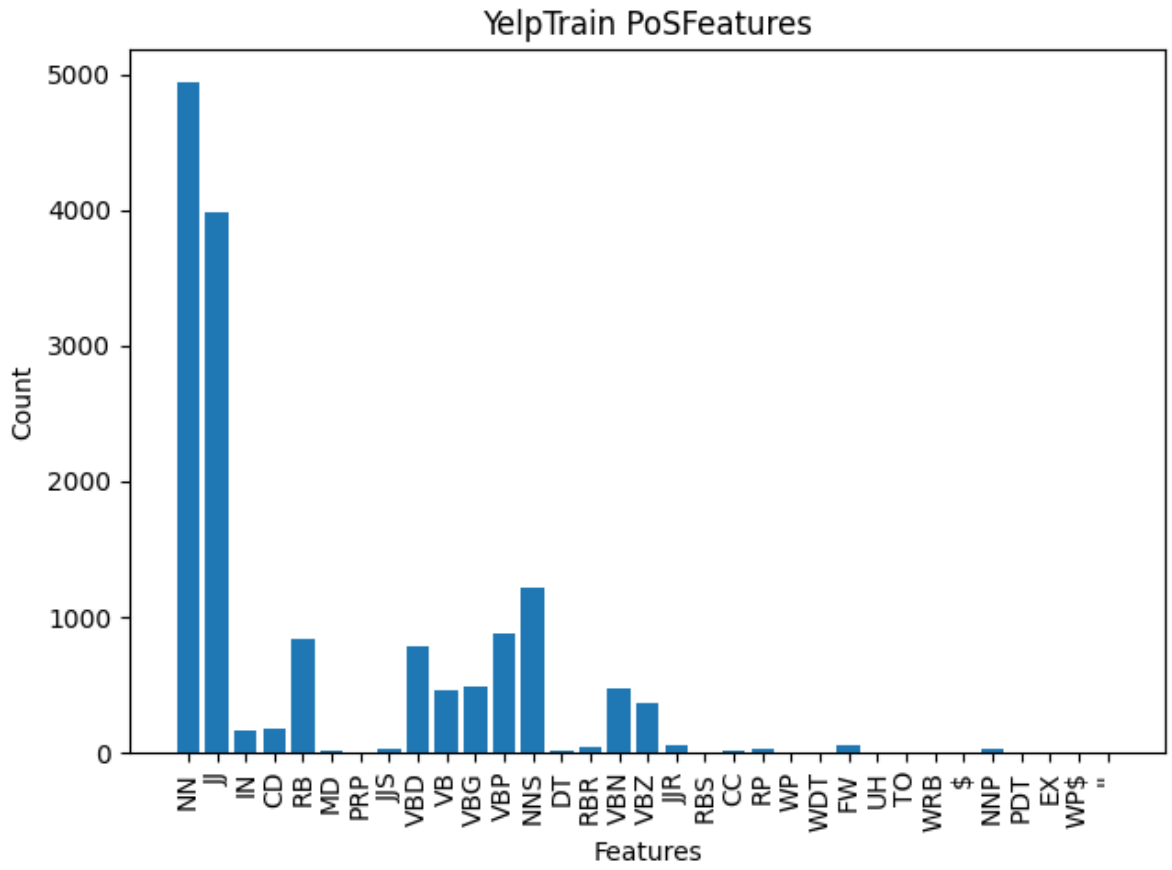
To create charts from attributes we used nltk and spacy libraries. We started with nltk and found PoS tags and IoB tags, later discovered the spacy library and found NER tags using spacy.

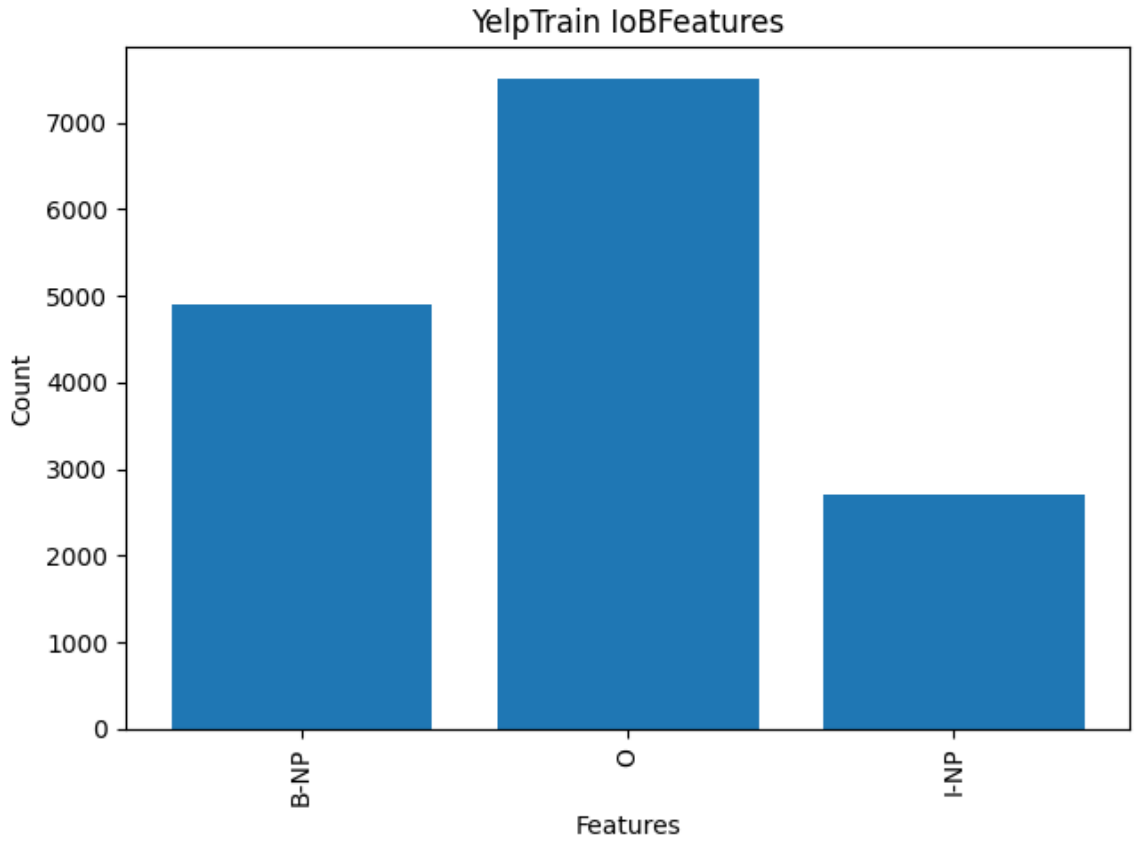
Firstly we tokenized the whole dataset and removed stopwords. Removing stop words will increase the accuracy and remove noise. Later PoS tagged the dataset. Lastly used this information to find IoB tags. After all this processes used the spacy library to find NER tags.

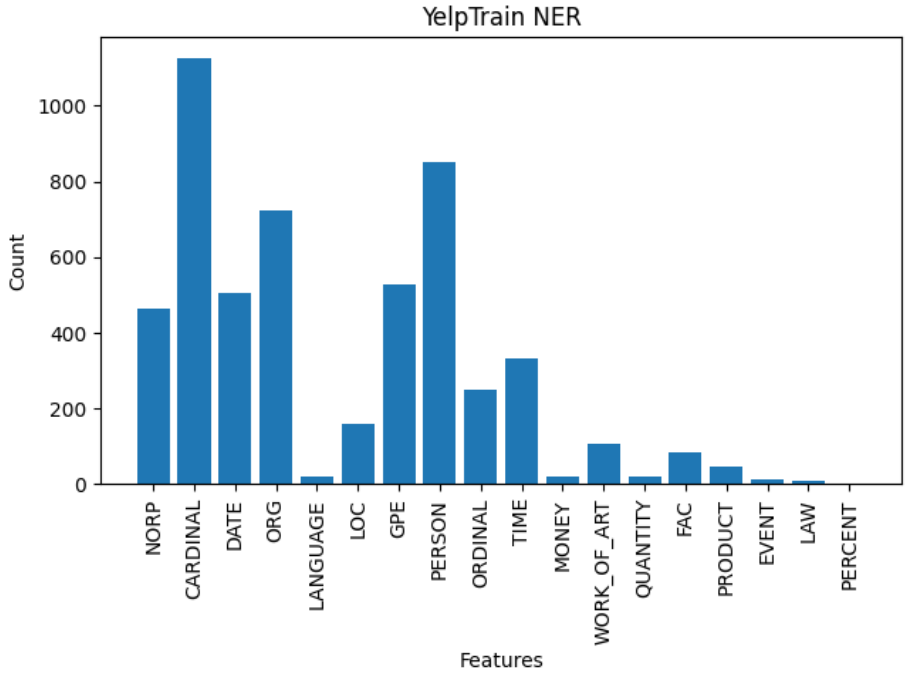
Currently we have 18 attributes : 'NORP', 'CARDINAL', 'DATE', 'ORG', 'LANGUAGE', 'LOC', 'GPE', 'PERSON', 'ORDINAL', 'TIME', 'MONEY','WORK_OF_ART', 'QUANTITY', 'FAC', 'PRODUCT', 'EVENT', 'LAW', 'PERCENT'. Result is 0 if review is fake, 1 if it’s true.

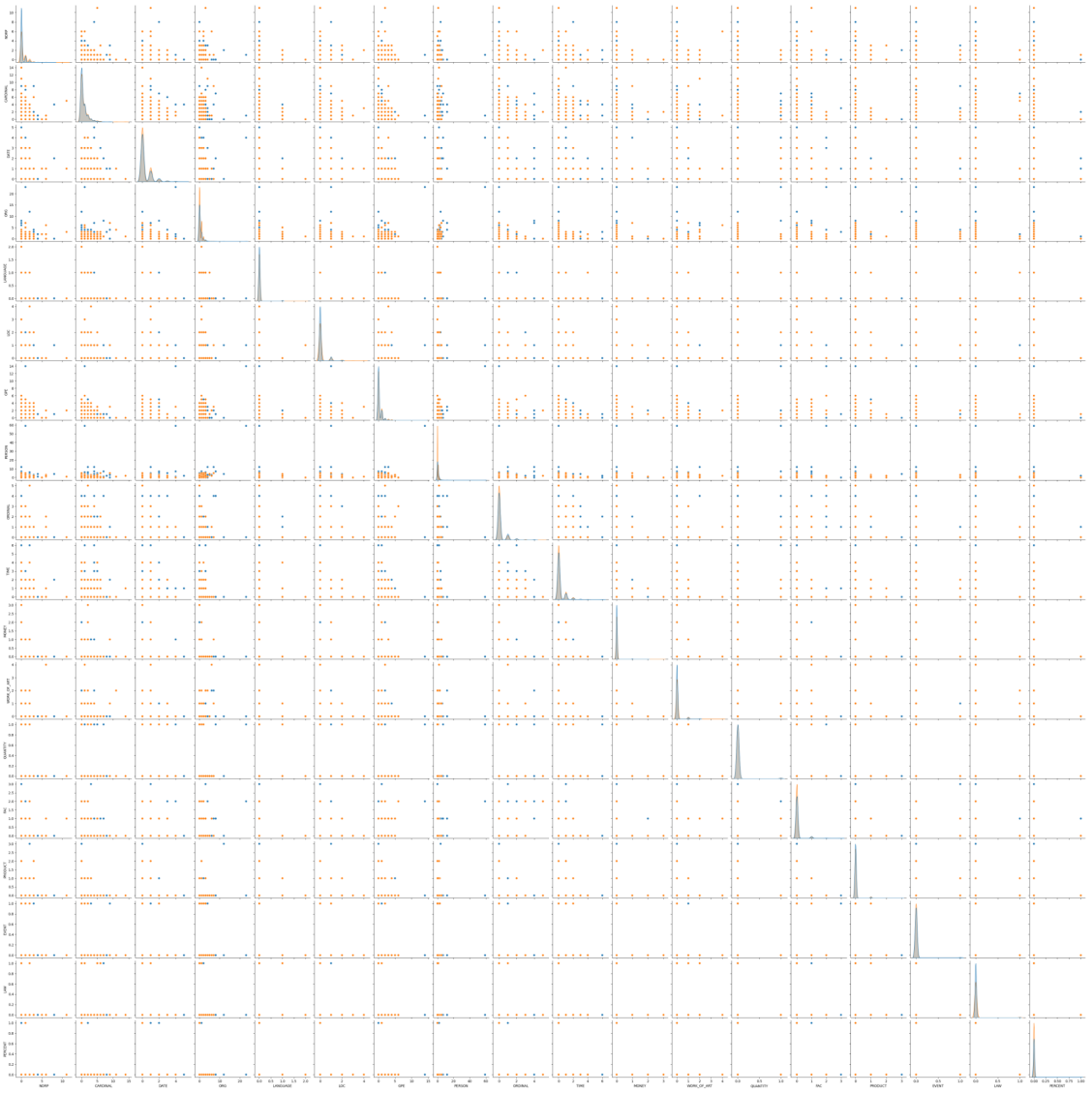
Created scatter plot with seaborn library. All charts are uploaded on github with higher resolution.

2.1- Yelp Train

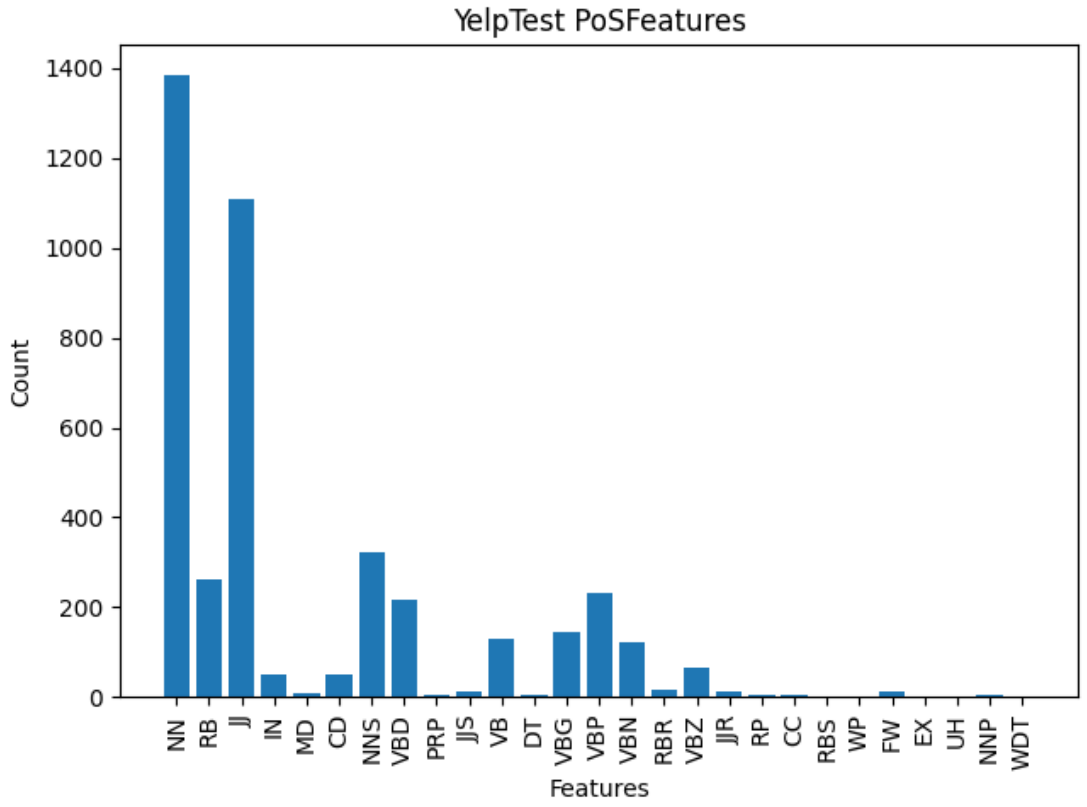


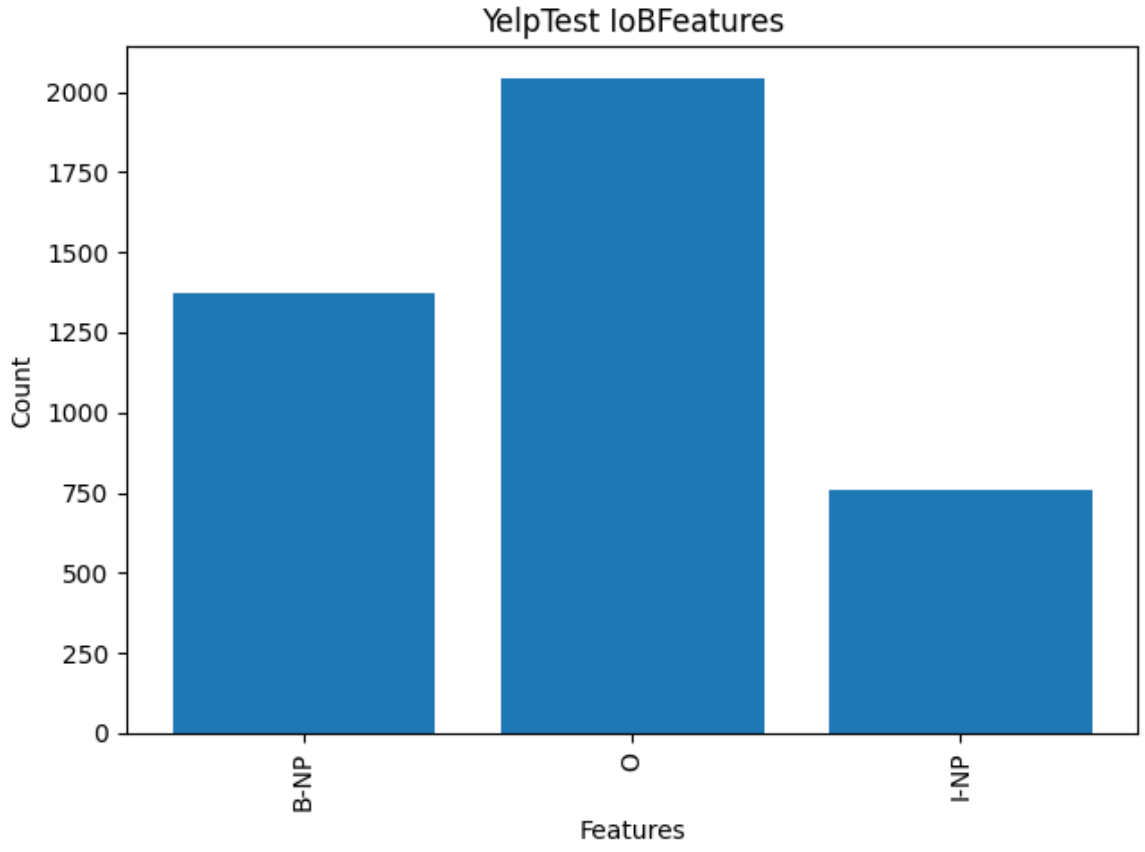


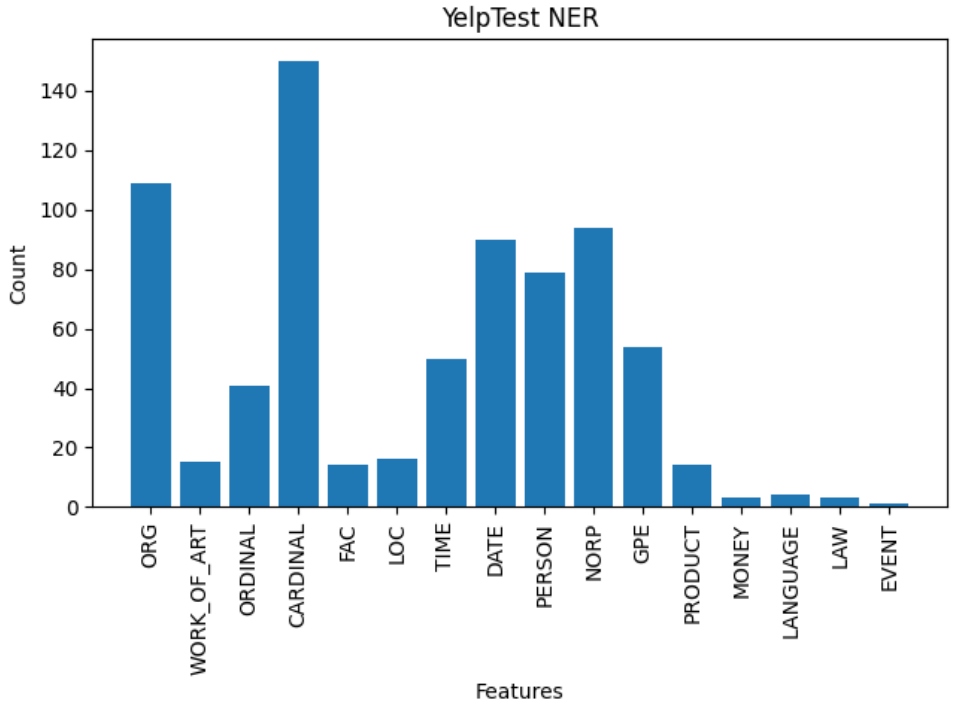

YelpTrain NER
Count
1000
800
600
400
200
0
NORP
CARDINAL
DATE
ORG
LANGUAGE
LOC
GPE
PERSON
ORDINAL
TIME
MONEY
WORK_OF_ART
QUANTITY
FAC
PRODUCT
EVENT
LAW
PERCENT
Features

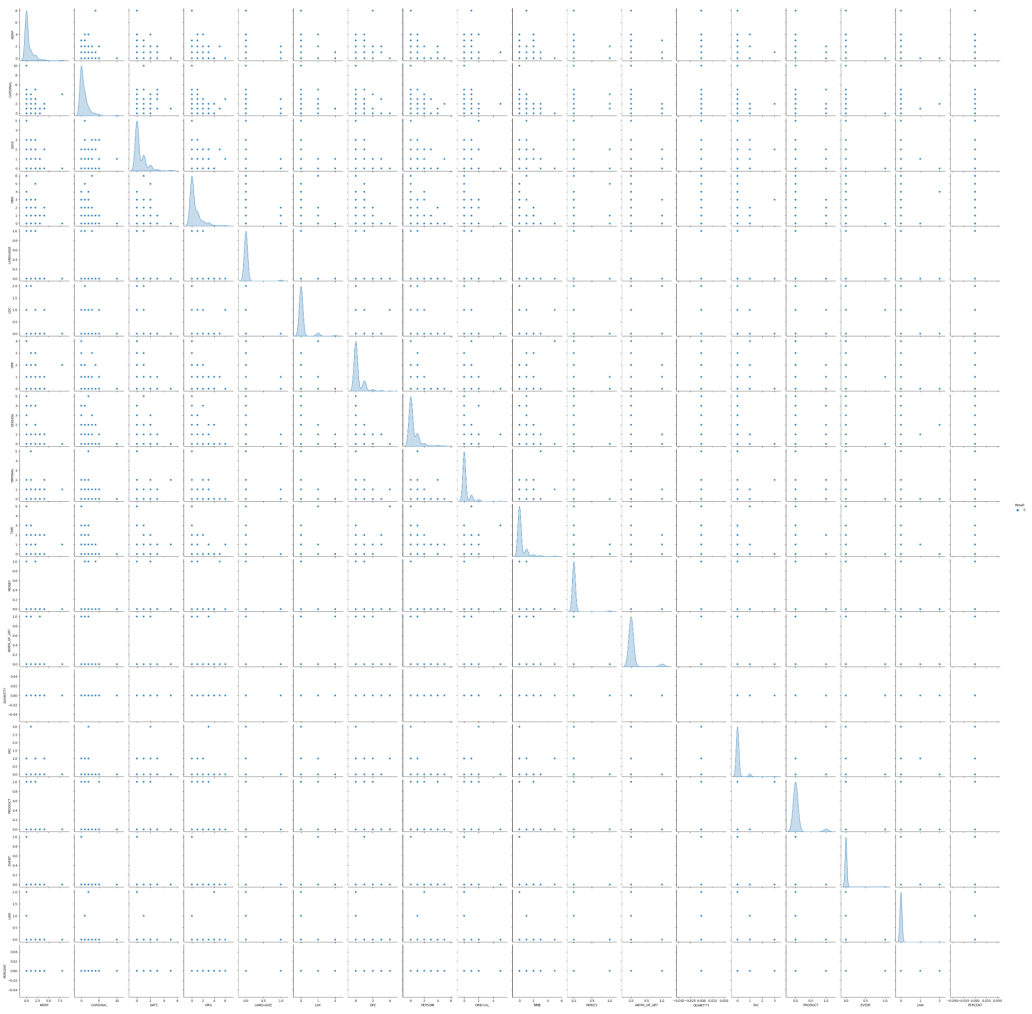


2.2- Yelp Test

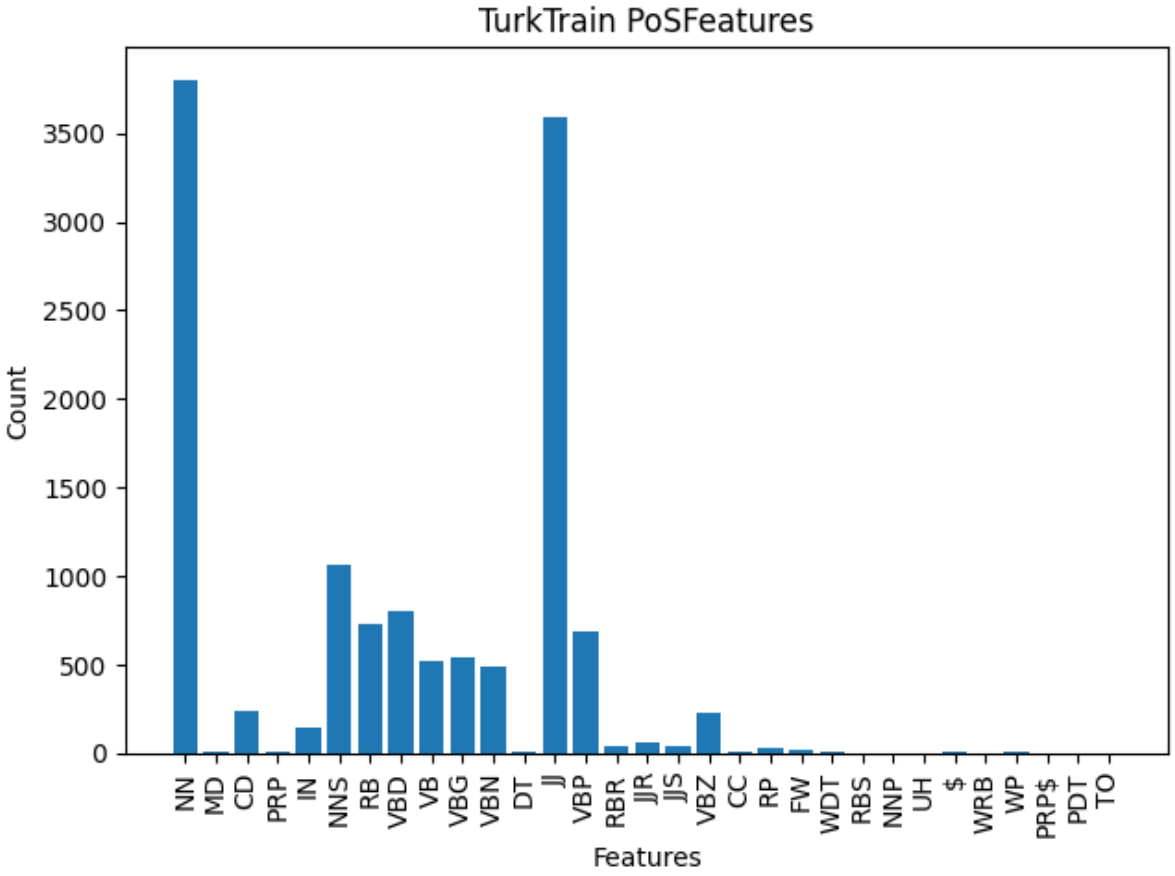

YelpTest PoSFeatures
Count
Features
0
200
400
600
800
1000
1200
1400
NN
RB
JJ
IN
MD
CD
NNS
VBD
PRP
JJS
VB
DT
VBG
VBP
VBN
RBR
VBZ
JJR
RP
CC
RBS
WP
FW
EX
UH
NNP
WDT

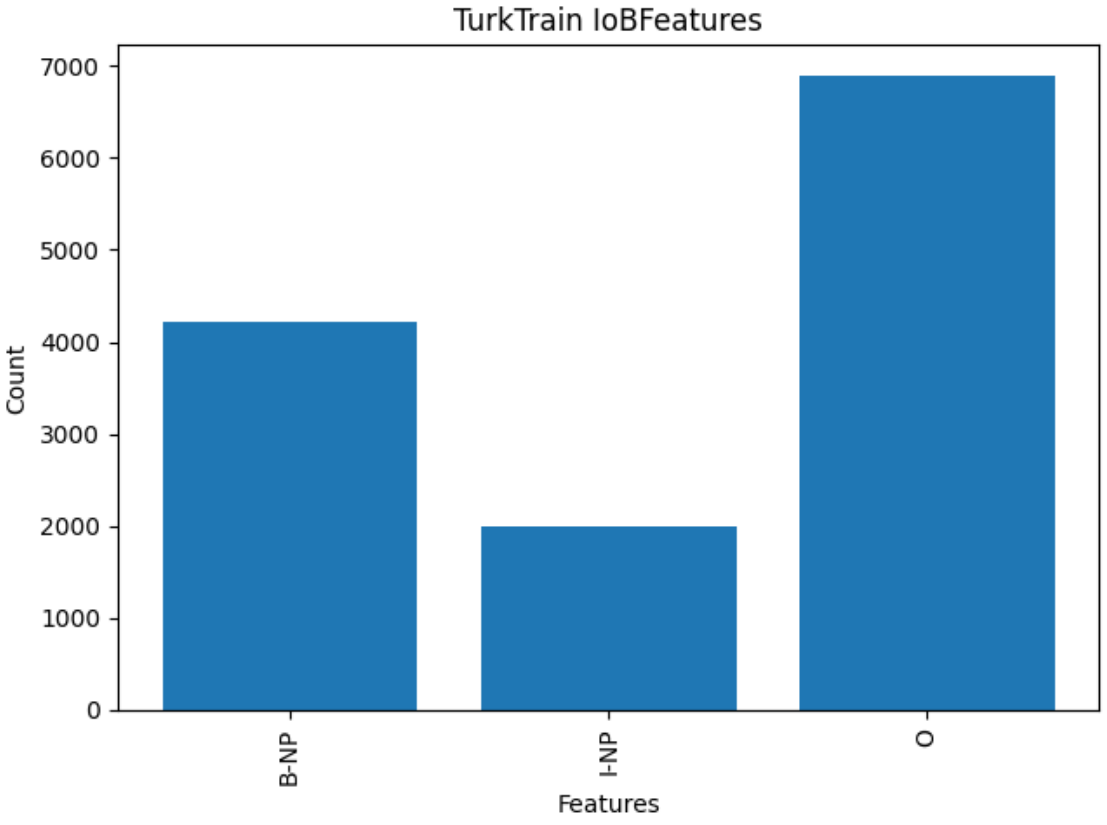

YelpTest IoBFeatures
Count
Features
0
250
500
750
1000
1250
1500
1750
2000
B-NP
O
I-NP

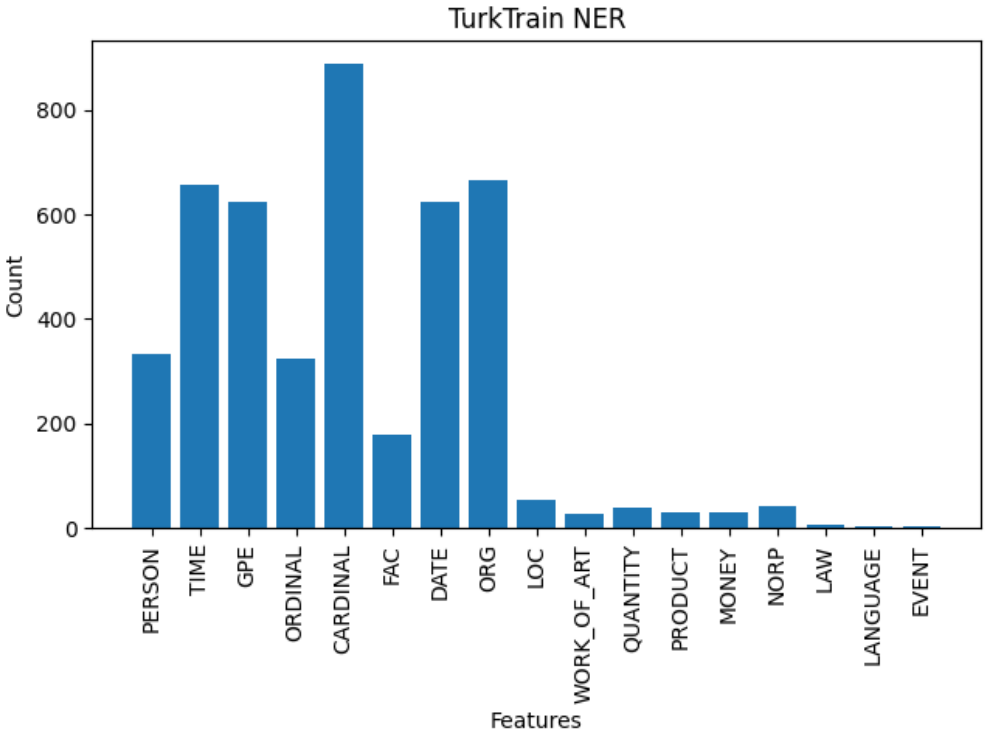

YelpTest NER
Count
0
20
40
60
80
100
120
140
ORG
WORK_OF_ART
ORDINAL
CARDINAL
FAC
LOC
TIME
DATE
PERSON
NORP
GPE
PRODUCT
MONEY
LANGUAGE
LAW
EVENT
Features

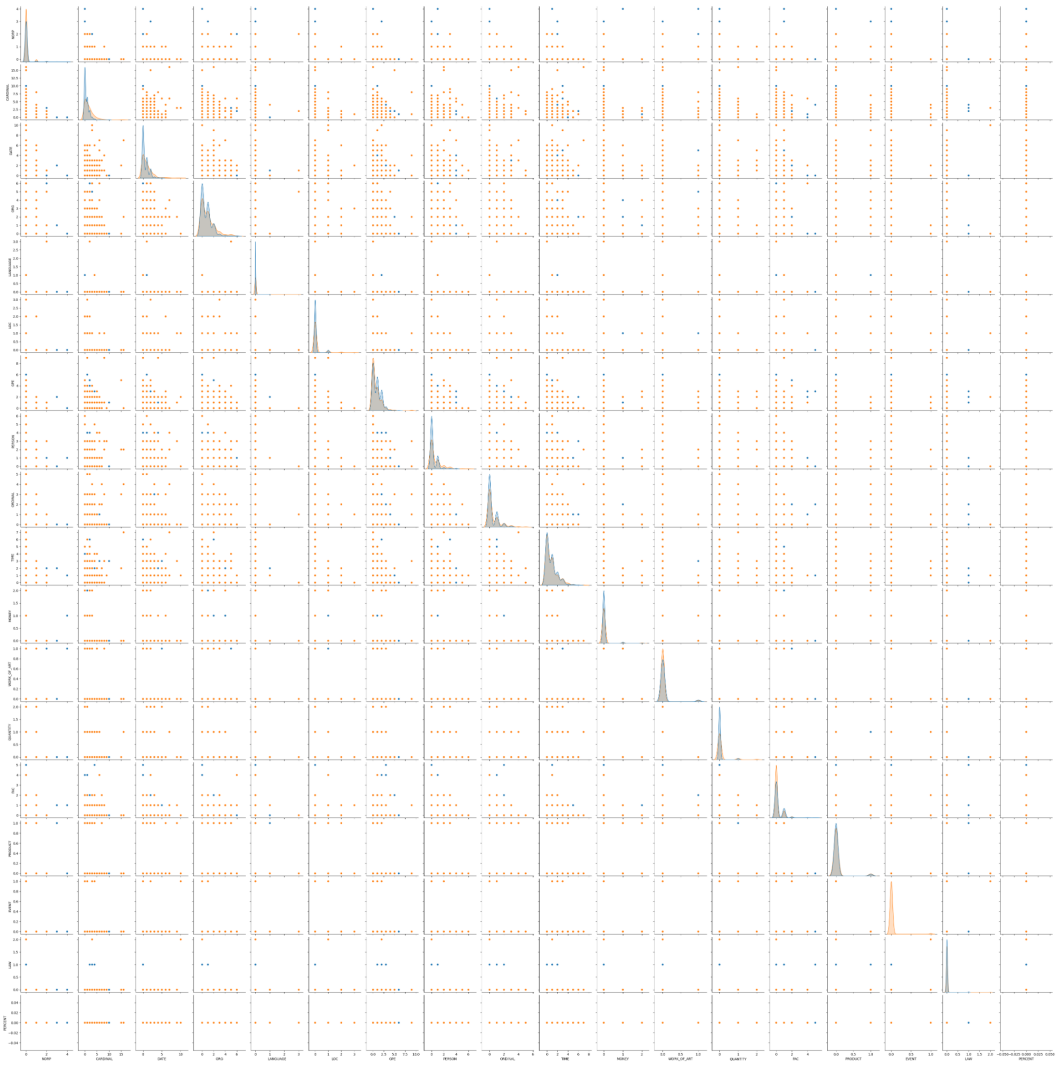


2.3- Turk Train

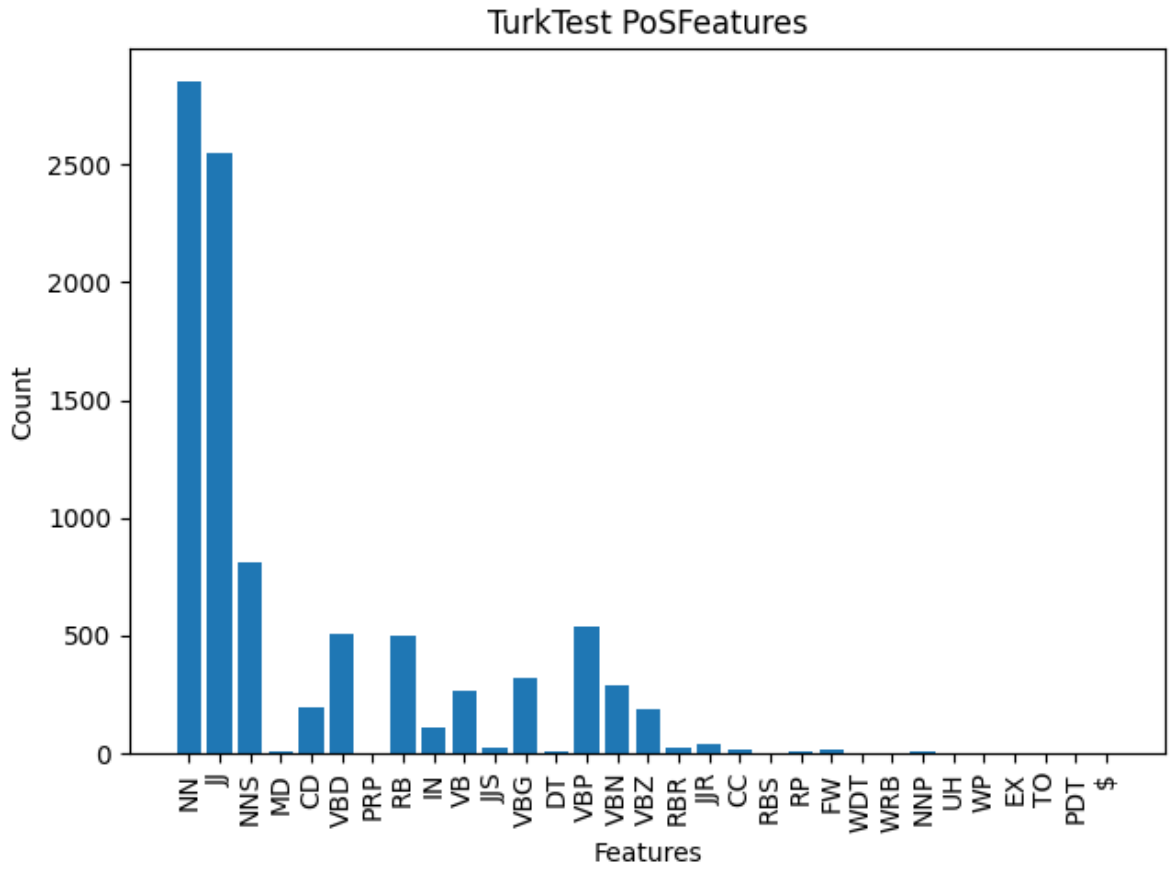


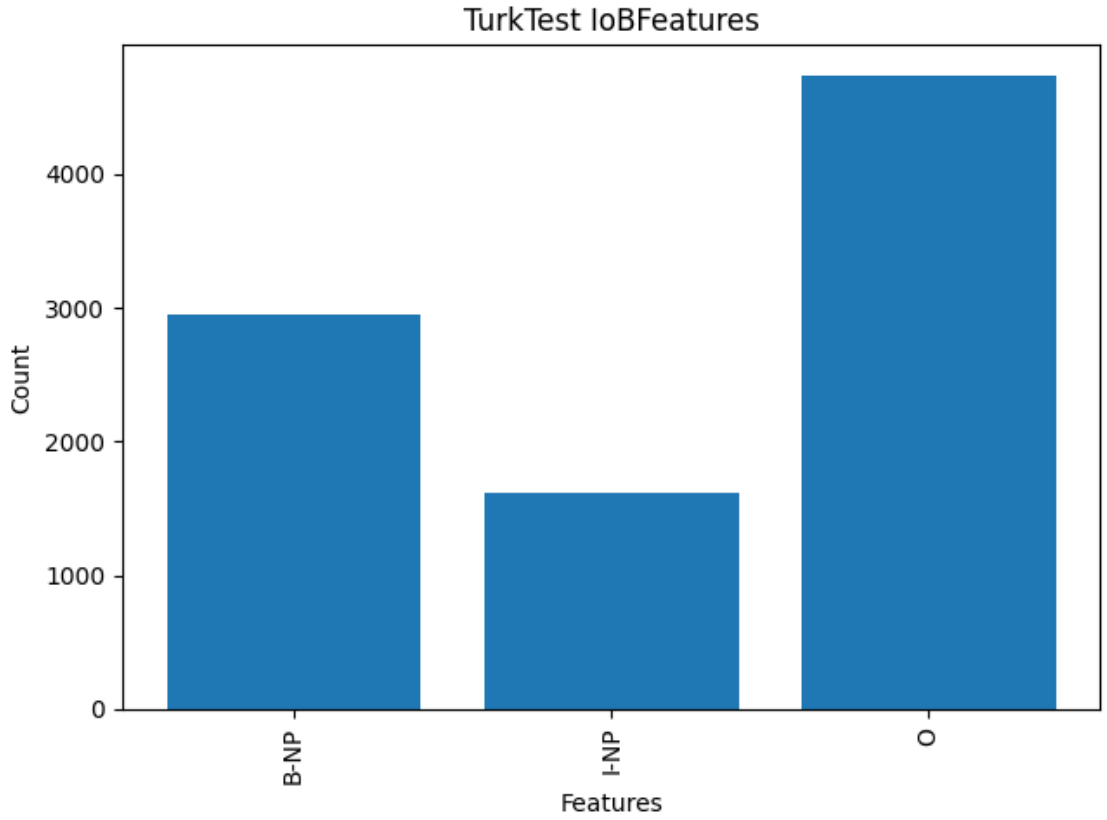


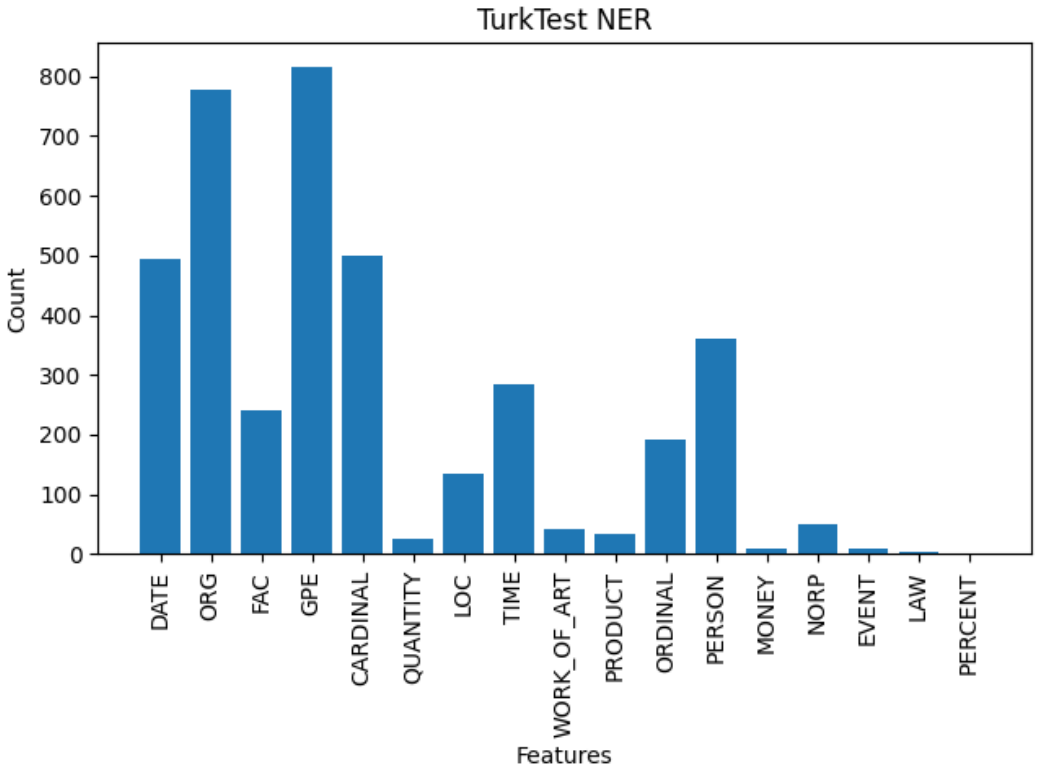

TurkTrain NER
Count
800
600
400
200
0
PERSON
TIME
GPE
ORDINAL
CARDINAL
FAC
DATE
ORG
LOC
WORK_OF_ART
QUANTITY
PRODUCT
MONEY
NORP
LAW
LANGUAGE
EVENT
Features

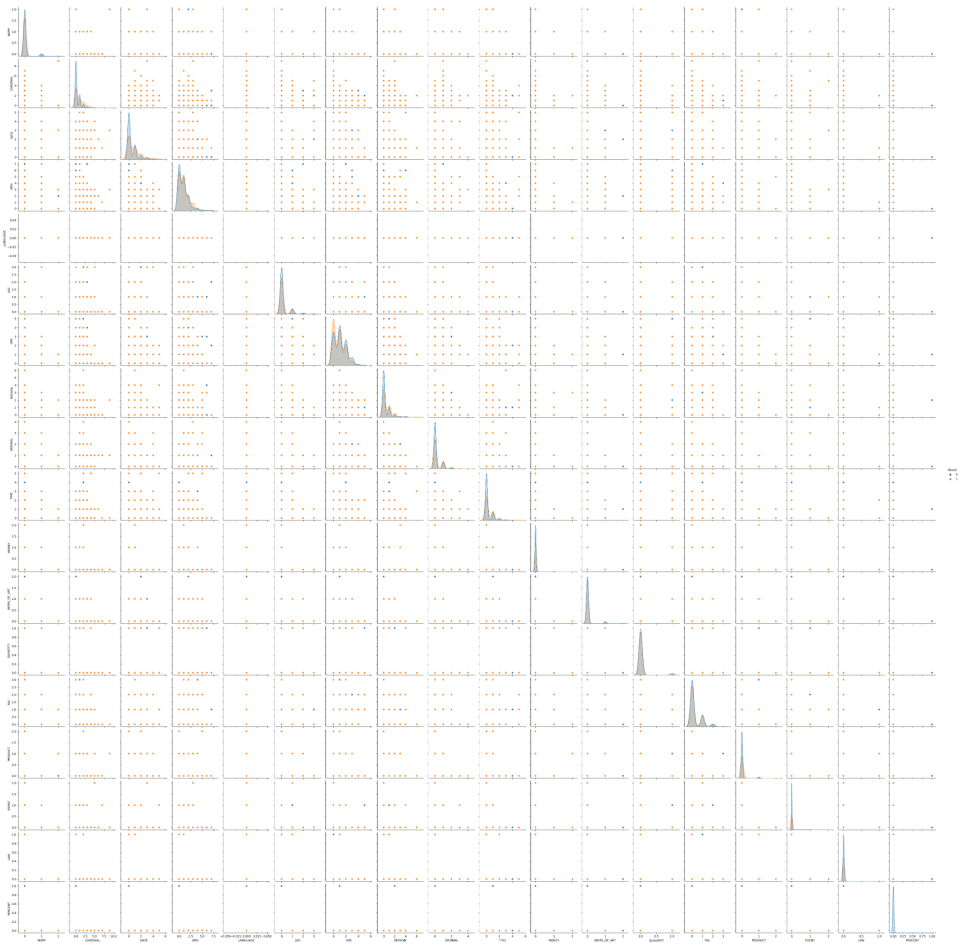


2.4- Turk Test


TurkTest PoSFeatures
Count
0
500
1000
1500
2000
2500
NN
JJ
NNS
MD
CD
VBD
PRP
RB
IN
VB
JJS
VBG
DT
VBP
VBN
VBZ
RBR
JJR
CC
RBS
RP
FW
WDT
WRB
NNP
UH
WP
EX
TO
PDT
$
Features


TurkTest IoBFeatures
Count
0
1000
2000
3000
4000
B-NP
I-NP
O
Features


TurkTest NER
Count
0
100
200
300
400
500
600
700
800
DATE
ORG
FAC
GPE
CARDINAL
QUANTITY
LOC
TIME
WORK_OF_ART
PRODUCT
ORDINAL
PERSON
MONEY
NORP
EVENT
LAW
PERCENT
Features



CSE 4062 - Semester 2021

Intro to Data Science and Analytics

Delivery #4 - Predictive Analytics

Group 2 Members:

Aleyna BOZACI 150319630 - aleynabozacii@gmail.com

Bekir ÖZKAN 150319557 - bekirozkan9698@gmail.com

Merve YAYIN 150116051 - yayinm8@gmail.com

Metehan ERTAN 150117051 - metehan.ertan@hotmail.com

Mustafa USCA 150416054 - uscamustafa64@gmail.com

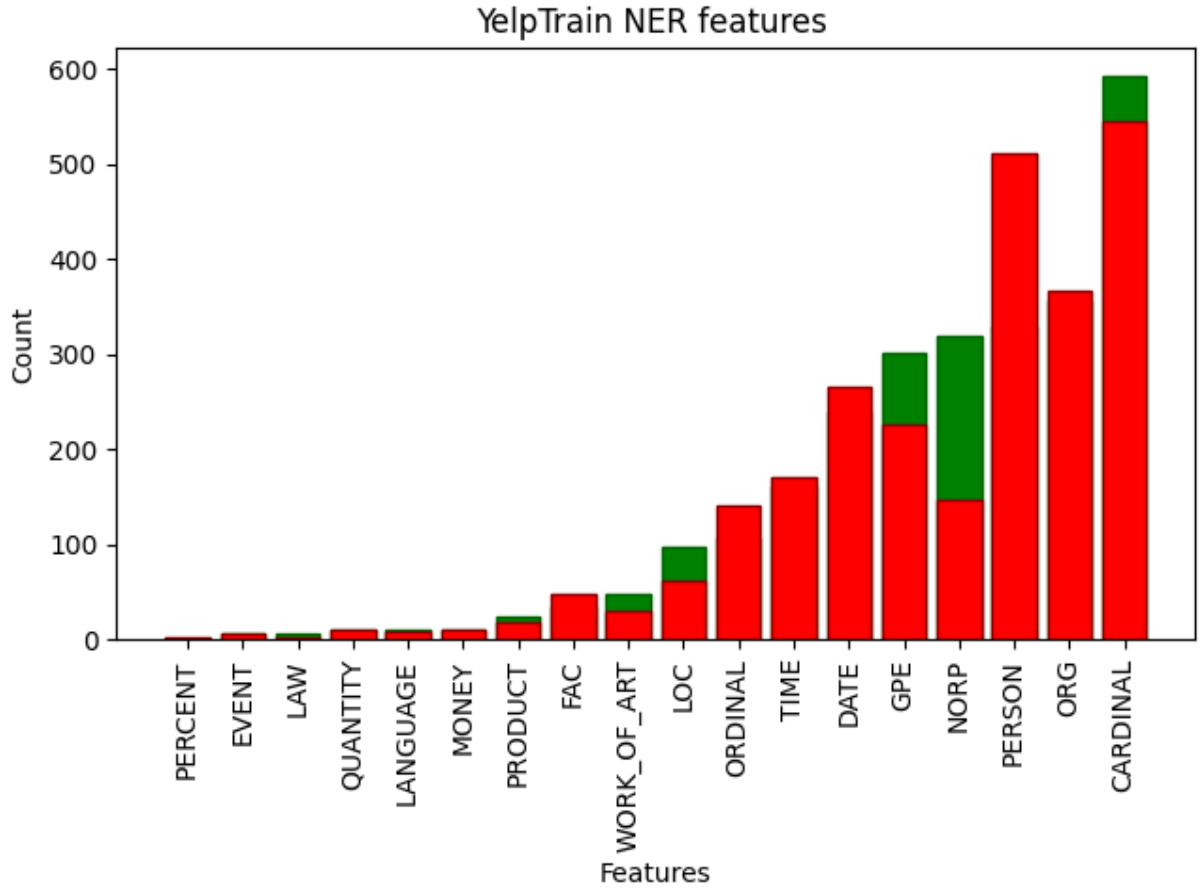
1 - Statistics

As our dataset is a text-based dataset, all data is our feature. Each algorithm uses each word as a feature so we cannot decrease or select any feature. Only methods we can use to decrease feature count are stopword removal and punctuation removal. We can use Named Entity Recognition to determine which attributes are more important and how they affect them.

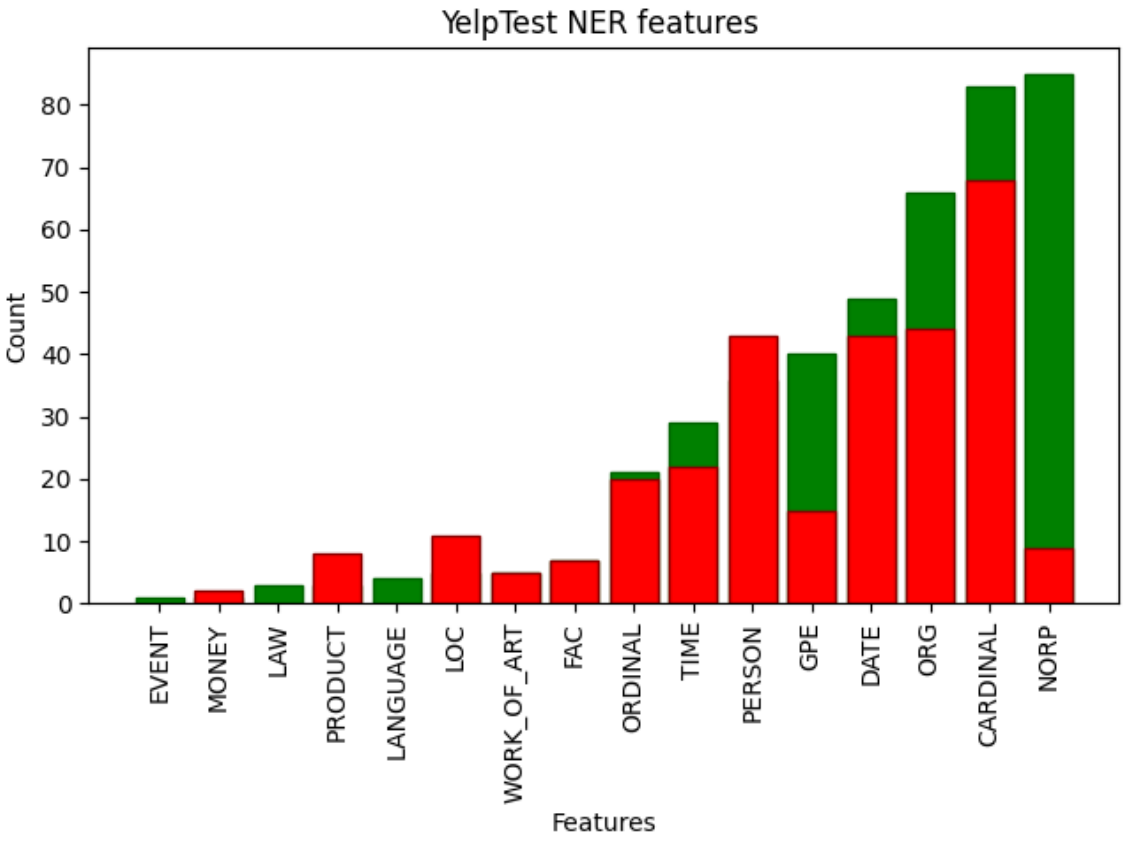
Currently we have 18 attributes : 'NORP', 'CARDINAL', 'DATE', 'ORG', 'LANGUAGE', 'LOC', 'GPE', 'PERSON', 'ORDINAL', 'TIME', 'MONEY','WORK_OF_ART', 'QUANTITY', 'FAC', 'PRODUCT', 'EVENT', 'LAW', 'PERCENT'.

On the chart, red bars are for Fake and green bars are for True reviews.

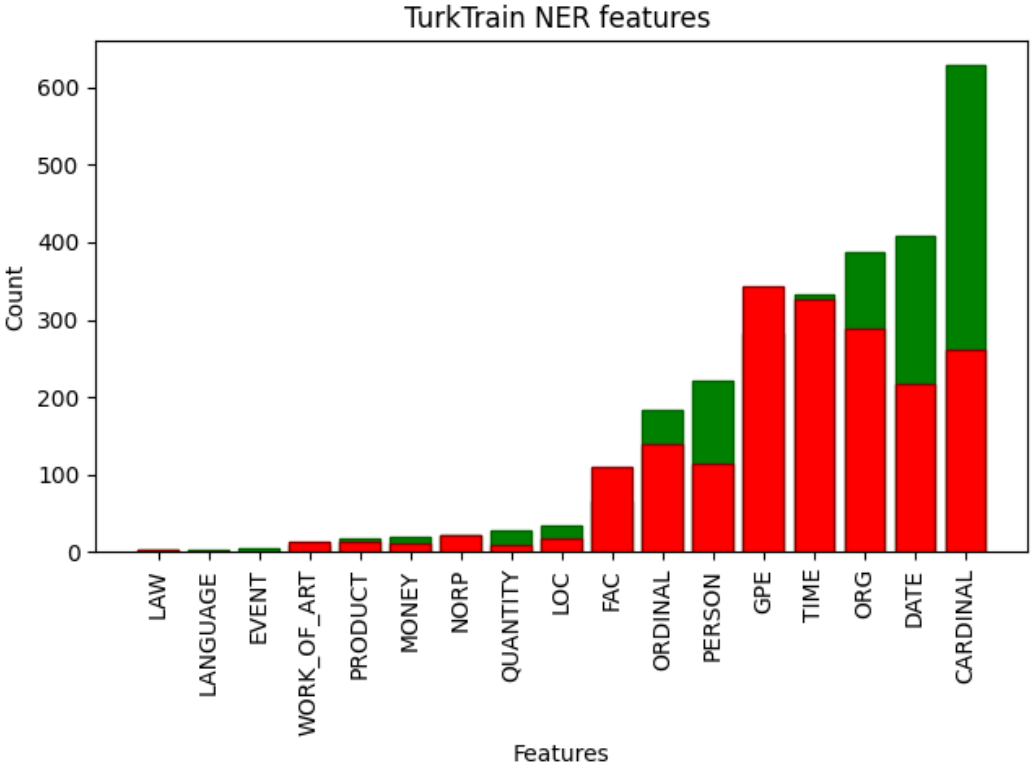
1.1 - Yelp train dataset



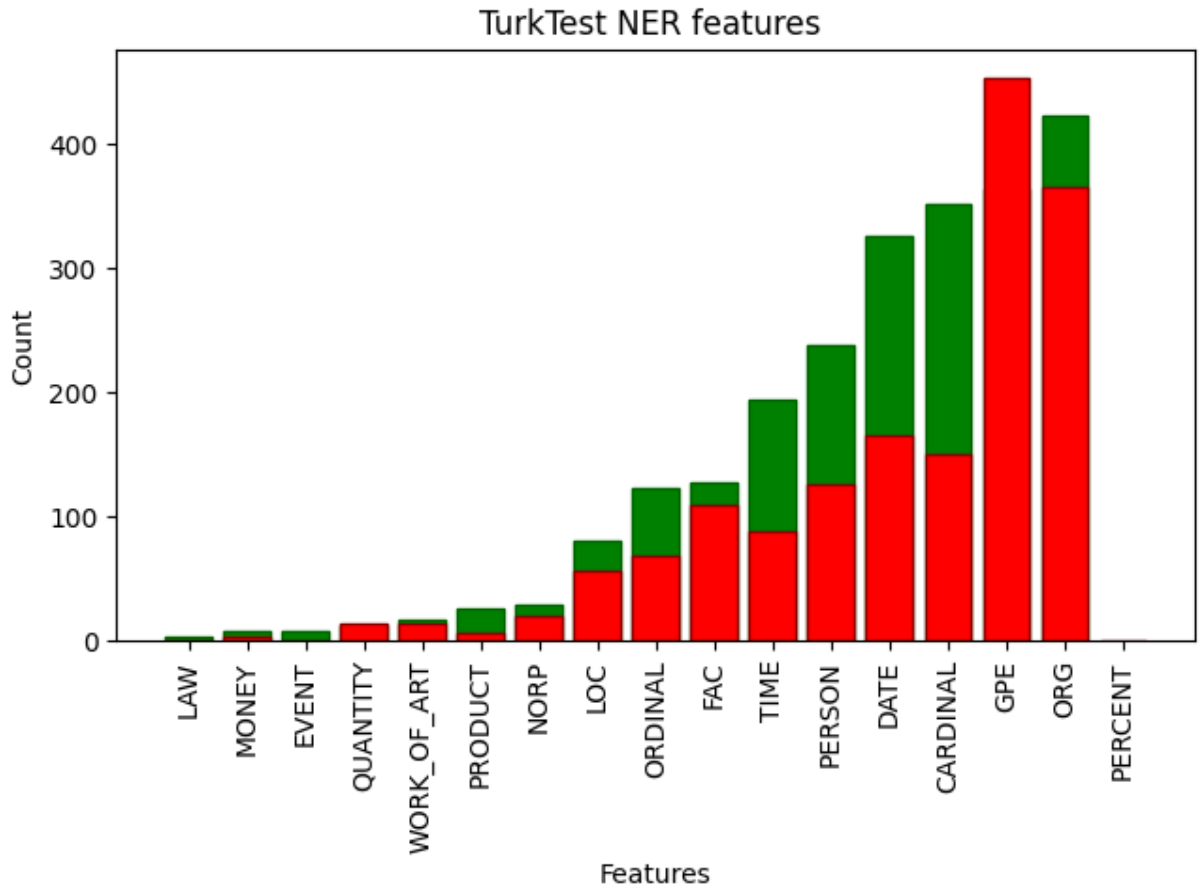
1.2 - Yelp test dataset



1.3 - Turk train dataset



1.4 - Turk test dataset



2 - Classification

For the classification part we decided to use 3 algorithms. These algorithms are Multinomial Naive Bayes, Random Forest and Convolutional Neural Network. All these algorithms are supervised algorithms. We decided that these algorithms will fit great for our project and datasets. Each algorithm is trained and tested for our 2 different datasets. Each algorithm uses each word as a feature as our dataset is a text-based dataset.

For MNB and RF algorithms we splitted data to 0.8 for training and 0.2 for testing. For CNN we extracted 200 reviews for yelp and used negative turk for training and positive turk for testing. All Accuracy, Precision, Recall and F1 scores are listed below.

2.1 - Multinomial Naive Bayes (MNB)

2.1.1 - Yelp dataset

Accuracy: 80.83

Precision Score: 0.8083333333333333

Recall Score: 0.8083333333333333

F1 Score: 0.808333333333333

2.1.2 - Turk dataset

Accuracy: 84.17

Precision Score: 0.8416666666666667

Recall Score: 0.8416666666666667

F1 Score: 0.8416666666666667

2.2 - Random Forest (RF)

2.2.1 - Yelp dataset

Accuracy: 75.62

Precision Score: 0.75625

Recall Score: 0.75625

F1 Score: 0.75625

2.2.2 - Turk dataset

Accuracy: 80.62

Precision Score: 0.80625

Recall Score: 0.80625

F1 Score: 0.8062499999999999

2.3 - Convolutional Neural Network (CNN)

2.3.1 - Yelp dataset

2.3.1.1 - Training part

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 404, 100)	204800
conv1d (Conv1D)	(None, 397, 32)	25632
max_pooling1d (MaxPooling1D)	(None, 198, 32)	0
flatten (Flatten)	(None, 6336)	0
dense (Dense)	(None, 10)	63370
dense_1 (Dense)	(None, 1)	11

Total params: 293,813

Trainable params: 293,813

Non-trainable params: 0

Epoch 10/10 - loss: 0.0066 - accuracy: 0.9981

2.3.1.1 - Testing part

Accuracy: 73.50

Precision Score: 0.7355889724310777

Recall Score: 0.735

F1 Score: 0.7348342714196372

2.3.2 - Turk dataset

2.3.2.1 - Training part

Model: "sequential"

Layer (type)	Output Shape	Param #
embedding (Embedding)	(None, 333, 100)	194100
conv1d (Conv1D)	(None, 326, 32)	25632
max_pooling1d (MaxPooling1D)	(None, 163, 32)	0
flatten (Flatten)	(None, 5216)	0
dense (Dense)	(None, 10)	52170
dense_1 (Dense)	(None, 1)	11

Total params: 271,913

Trainable params: 271,913

Non-trainable params: 0

Epoch 10/10 - loss: 0.0012 - accuracy: 1.0000

2.3.2.1 - Testing part

Accuracy: 77.62

Precision Score: 0.7935102363355048

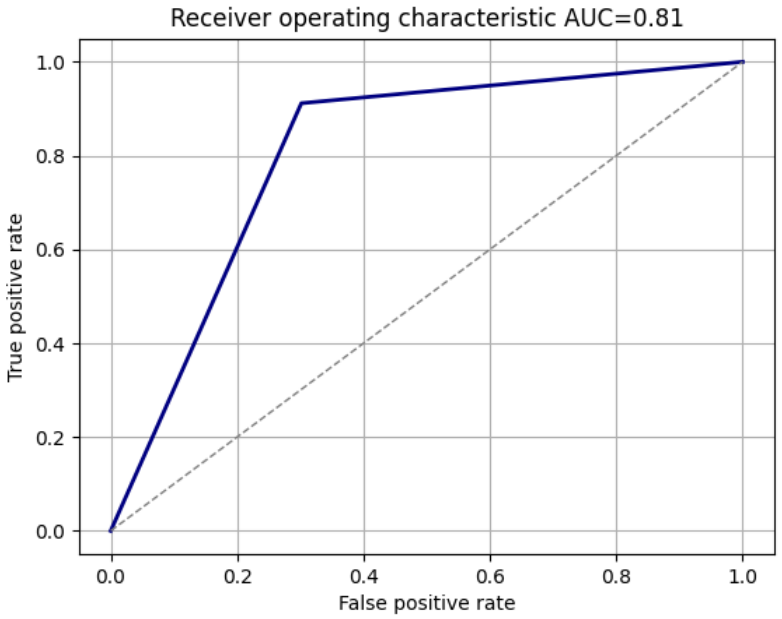
Recall Score: 0.77625

F1 Score: 0.7729114433919926

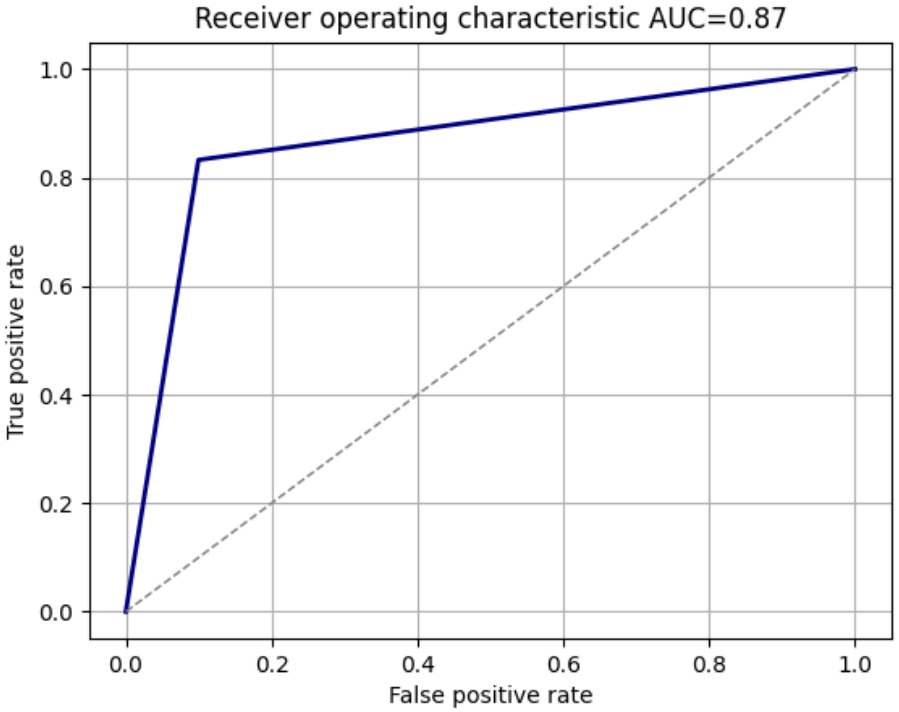
3 - ROC Curve

3.1 - Multinomial Naive Bayes (MNB)

3.1.1 - Yelp dataset

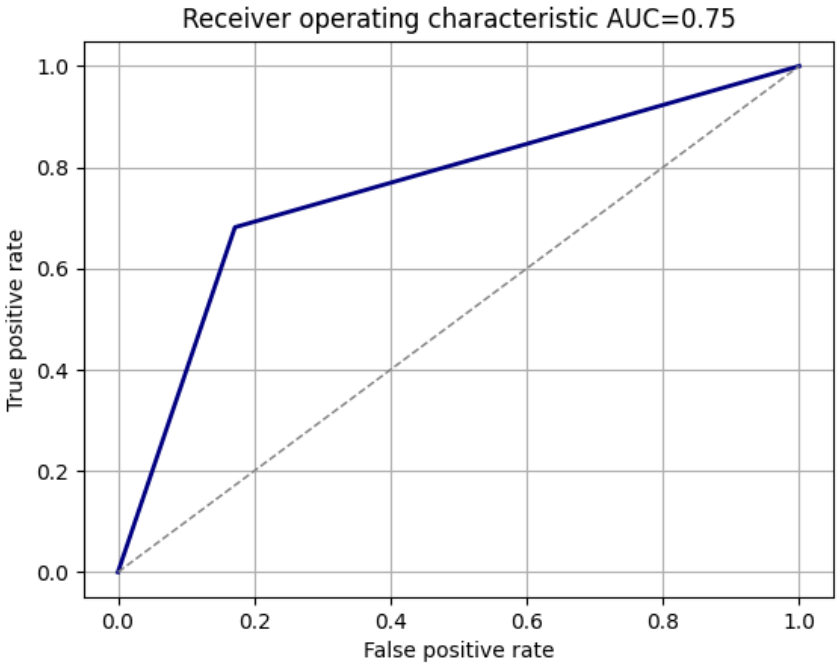


3.1.2 - Turk dataset

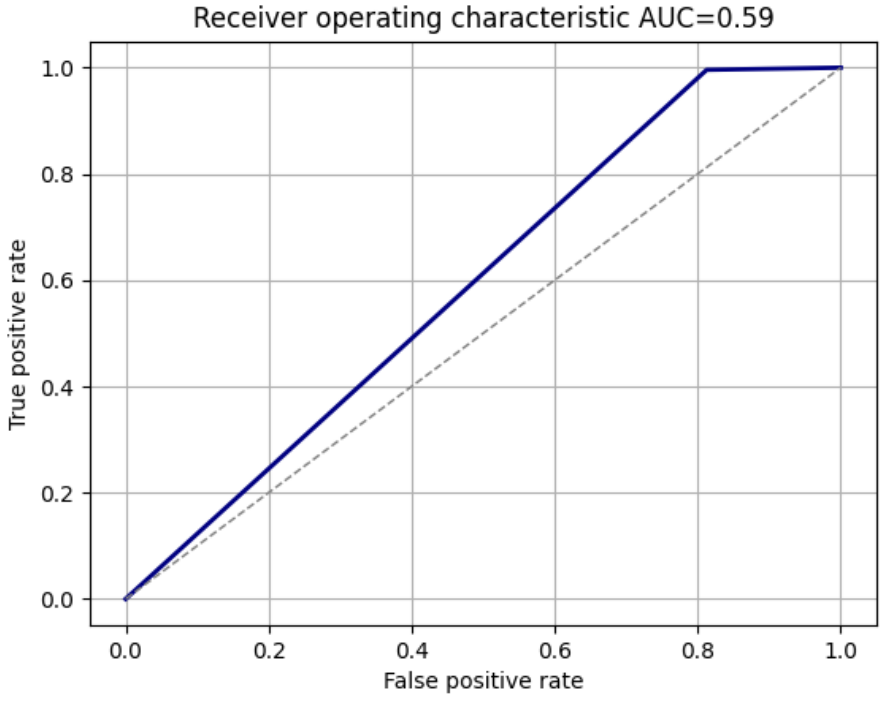


3.2 - Random Forest (RF)

3.2.1 - Yelp dataset

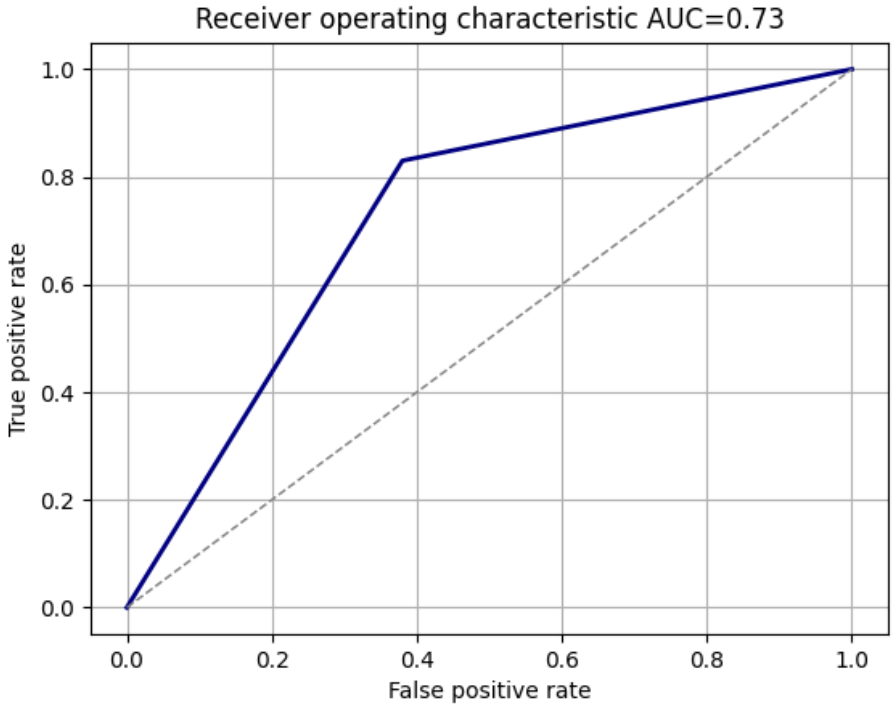


3.2.2 - Turk dataset

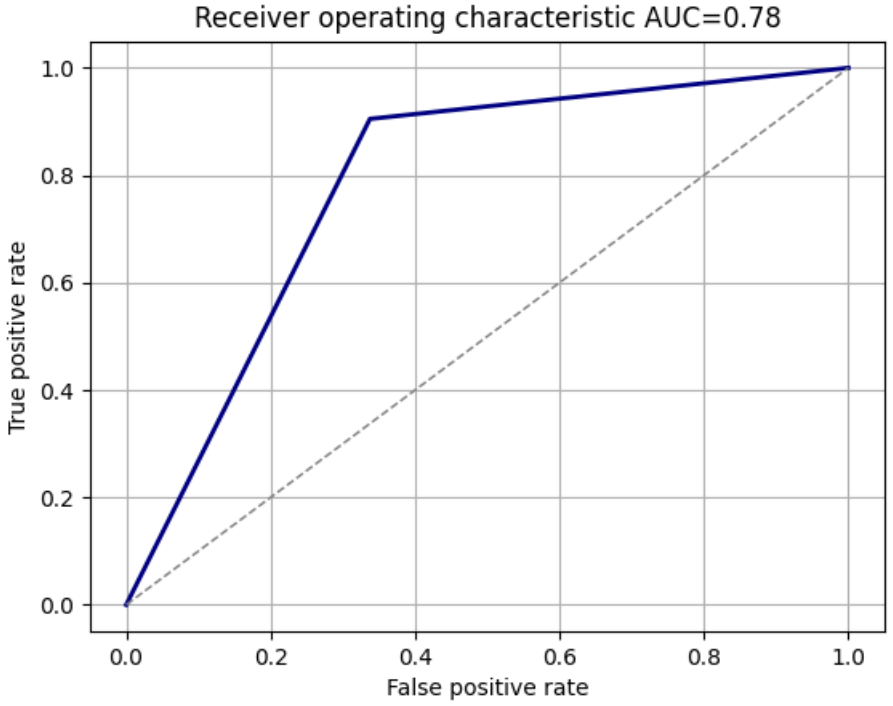


3.3 - Convolutional Neural Network (CNN)

3.3.1 - Yelp dataset



3.3.2 - Turk dataset



4 - Confusion matrix

4.1 - Yelp dataset

	Accuracy	Precision	Recall	F1 Score
MNB	80.83	0.808	0.808	0.808
RF	75.62	0.7562	0.7562	0.7562
CNN	73.50	0.7355	0.735	0.7348

4.1 - Turk dataset

	Accuracy	Precision	Recall	F1 Score
MNB	84.17	0.8416	0.8416	0.8416
RF	80.62	0.8062	0.8062	0.8062
CNN	77.62	0.7935	0.7762	0.7729

5 - Statistical significance analysis

Our best method is MNB with %84.17 accuracy. Closest competitor is RF with %80.62 accuracy. So we run 10-fold cross validation with 10 random hold-out with 6 repeats.

5.1 - Yelp dataset

5.1.1 - MNB

[0.78571429 0.84821429 0.78571429 0.79464286 0.8125 0.8125
0.82142857 0.83928571 0.79464286 0.80357143 0.77678571 0.83035714
0.83035714 0.8125 0.74107143 0.85714286 0.83035714 0.83035714
0.84821429 0.71428571 0.76785714 0.83928571 0.74107143 0.78571429
0.85714286 0.85714286 0.82142857 0.8125 0.80357143 0.8125
0.85714286 0.78571429 0.85714286 0.69642857 0.79464286 0.83928571
0.8125 0.83035714 0.78571429 0.79464286 0.72321429 0.76785714
0.75892857 0.84821429 0.82142857 0.8125 0.79464286 0.875
0.83035714 0.84821429 0.83035714 0.80357143 0.82142857 0.79464286
0.80357143 0.82142857 0.75 0.80357143 0.83035714 0.77678571]

5.1.2 - RF

[0.78125 0.7734375 0.7421875 0.78125 0.765625 0.8515625 0.7734375
0.78125 0.7421875 0.7890625 0.7890625 0.734375 0.828125 0.796875
0.765625 0.8046875 0.765625 0.7734375 0.796875 0.78125 0.78125
0.765625 0.7890625 0.8046875 0.7421875 0.6953125 0.796875 0.7734375
0.7890625 0.8203125 0.78125 0.84375 0.734375 0.75 0.84375
0.8046875 0.7890625 0.765625 0.7890625 0.7421875 0.75 0.71875
0.7734375 0.71875 0.828125 0.7734375 0.75 0.84375 0.8515625
0.7890625 0.7890625 0.796875 0.8515625 0.7734375 0.8125 0.8203125
0.7734375 0.7421875 0.7421875 0.7578125]

5.2 - Turk dataset

5.2.1 - MNB

[0.78571429 0.84821429 0.78571429 0.79464286 0.8125 0.8125
0.82142857 0.83928571 0.79464286 0.80357143 0.77678571 0.83035714
0.83035714 0.8125 0.74107143 0.85714286 0.83035714 0.83035714
0.84821429 0.71428571 0.76785714 0.83928571 0.74107143 0.78571429
0.85714286 0.85714286 0.82142857 0.8125 0.80357143 0.8125
0.85714286 0.78571429 0.85714286 0.69642857 0.79464286 0.83928571
0.8125 0.83035714 0.78571429 0.79464286 0.72321429 0.76785714
0.75892857 0.84821429 0.82142857 0.8125 0.79464286 0.875
0.83035714 0.84821429 0.83035714 0.80357143 0.82142857 0.79464286
0.80357143 0.82142857 0.75 0.80357143 0.83035714 0.77678571]

5.2.2 - RF

[0.8046875 0.8125 0.8125 0.765625 0.7421875 0.78125 0.8125
0.75 0.8671875 0.8046875 0.765625 0.796875 0.7890625 0.8046875
0.828125 0.8203125 0.8125 0.765625 0.78125 0.7890625 0.828125
0.7421875 0.7734375 0.8515625 0.8046875 0.765625 0.8125 0.8359375
0.796875 0.8203125 0.8046875 0.78125 0.765625 0.78125 0.8125
0.765625 0.8203125 0.859375 0.828125 0.8203125 0.7890625 0.78125
0.7578125 0.7890625 0.8359375 0.8125 0.8828125 0.8046875 0.75
0.71875 0.7265625 0.78125 0.7890625 0.84375 0.71875 0.796875
0.8203125 0.8671875 0.796875 0.796875]

6 - Description of results

As you can see from the tables and the data that is above MNB got the best results on all comparisons, RF got the second best and CNN got the worst result. We can't directly say that MNB is superior to CNN or vice-versa as these statistics can change for each dataset and each method. But for the datasets and methods we can see MNB got the best results in each comparison.

As we use text-based dataset we cannot change the features. This limits our flexibility. MNB is the most simple method that we used therefore it was better on datasets like this. CNN was more complicated and precise therefore it would be better on more featured datasets. We can see this as it was originally designed for image-based datasets. On our dataset CNN overfitted therefore got worse. We saw that on Epoch 10, CNN got 1.0000 accuracy and overfitted the training dataset.

We also had better results on Opinion Spam Corpus (Turk dataset) for all methods. This can be because it had less training data and more testing data. We think that this prevents overfitting.

If we look at the time complexity of these methods. MNB was the fastest followed by RF method. As there are Epochs on CNN it took the most amount of time. Each Epoch nearly took the same amount of time as the MNB method.

To get better results we analyzed the output of these 3 methods. We wanted to combine these 3 methods output inorder to achieve better accuracy. But because of our dataset, mostly all of our 3 methods give the same wrong output on the same data. We tried to give weight to their outputs to get a higher accuracy. We weighted outputs with their methods corresponding accuracy. If the total of these weighted outputs is greater than a threshold we counted that review as True. But it didn't help because our class attribute is nominal. Multiplying with 1 or 0 eliminated the logic of giving weight. So it decreased the total accuracy as it mostly eliminated the single correct output when all other 2 are wrong.

CSE 4062 - Semester 2021

Intro to Data Science and Analytics

Delivery #5 - Descriptive Analytics

Group 2 Members:

Aleyna BOZACI 150319630 - aleynabozacii@gmail.com

Bekir ÖZKAN 150319557 - bekirozkan9698@gmail.com

Merve YAYIN 150116051 - yayinm8@gmail.com

Metehan ERTAN 150117051 - metehan.ertan@hotmail.com

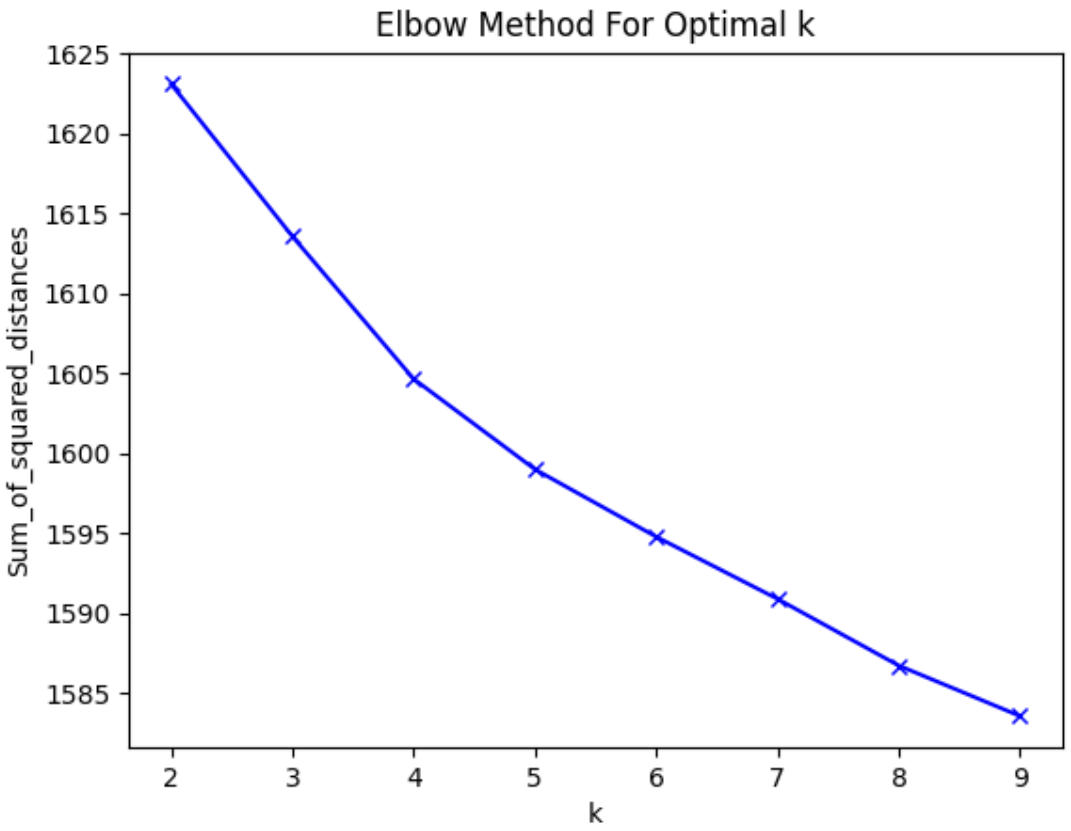
Mustafa USCA 150416054 - uscamustafa64@gmail.com

1 - Statistics

To find optimal K for k-means clustering we used the Elbow method. After finding the optimal K we created clusters and displayed most used words with WordCloud. As our datasets consist of the same topic most of the clusters got the same most used words. Most used ones change after the third ones.

1.1 - Yelp dataset

1.1.1 - Clustering Elbow method



1.1.2 - Clusters

id	cluster
1495	0
651	0
1459	0
1386	0
289	0
...	...
1318	6
1319	6
1320	6
1312	6
595	6

[1800 rows x 2 columns]

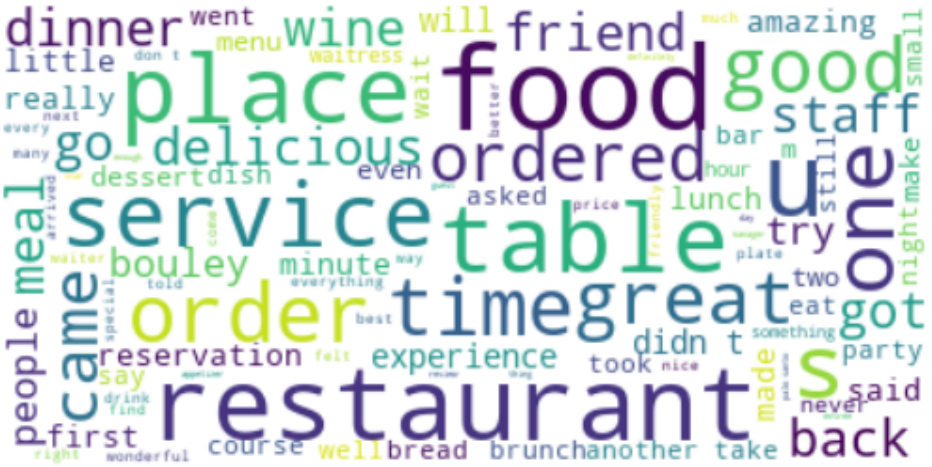
1.1.3 - Cluster 0

1.1.4 - Cluster 1

1.1.5 - Cluster 2

1.1.6 - Cluster 3

1.1.7 - Cluster 4

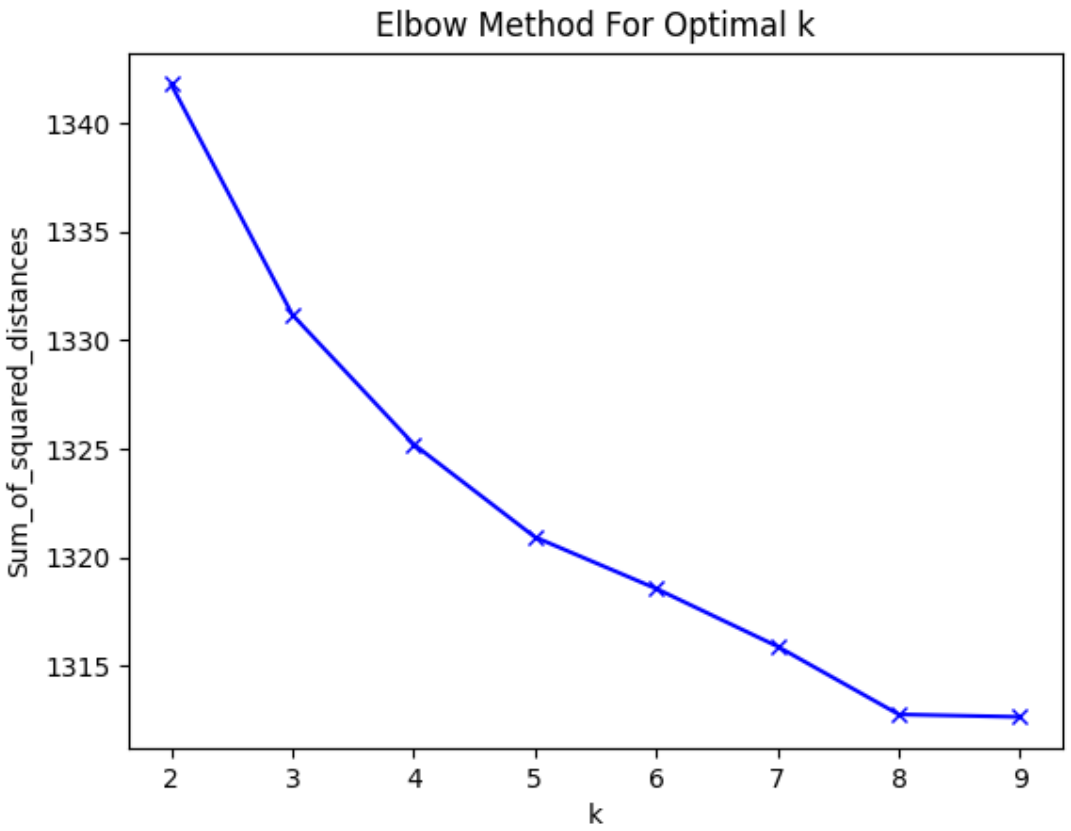


1.1.8 - Cluster 5

1.1.9 - Cluster 6

1.2 - Turk dataset

1.2.1 - Clustering Elbow method



1.2.2 - Clusters

id	cluster
1599	0
1087	0
1090	0
1091	0
1096	0
...	...

1358 6

1355 6

994 6

1352 6

1125 6

[1600 rows x 2 columns]

1.2.3 - Cluster 0

1.2.4 - Cluster 1

1.2.5 - Cluster 2

1.2.6 - Cluster 3

1.2.7 - Cluster 4

1.2.8 - Cluster 5

1.2.9 - Cluster 6

2 - Description of results

As mentioned before our datasets mostly focus on one topic. Because of this each cluster has some same words. For each cluster most used words start with the same 2 words. After that each cluster focusses on another topic. Clustering method is not really efficient for our project and datasets.