CSE 4062 - Semester 2021

Intro to Data Science and Analytics

Delivery #4 - Predictive Analytics

MARMARA ÜNİVERSİTESİ MÜHENDİSLİK FAKÜLTESİ

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

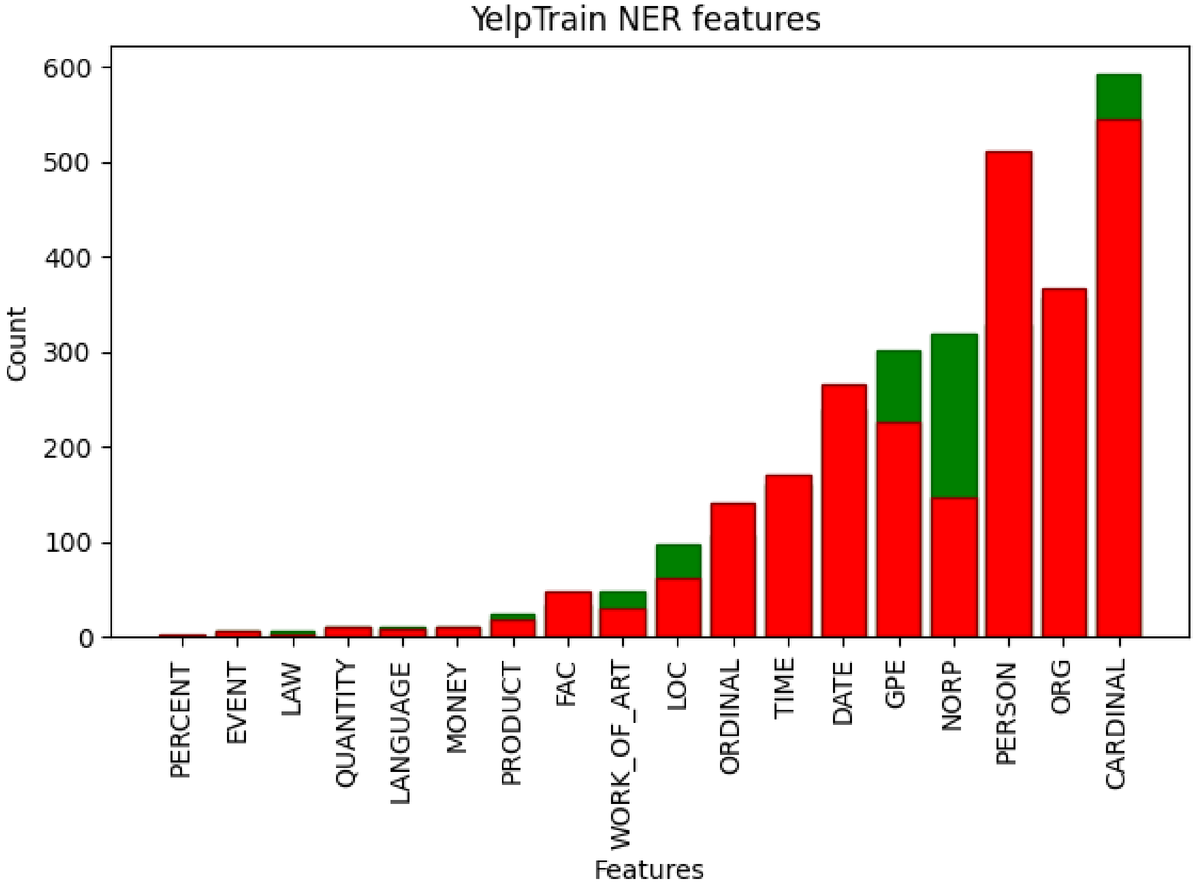
1 - Statistics

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.

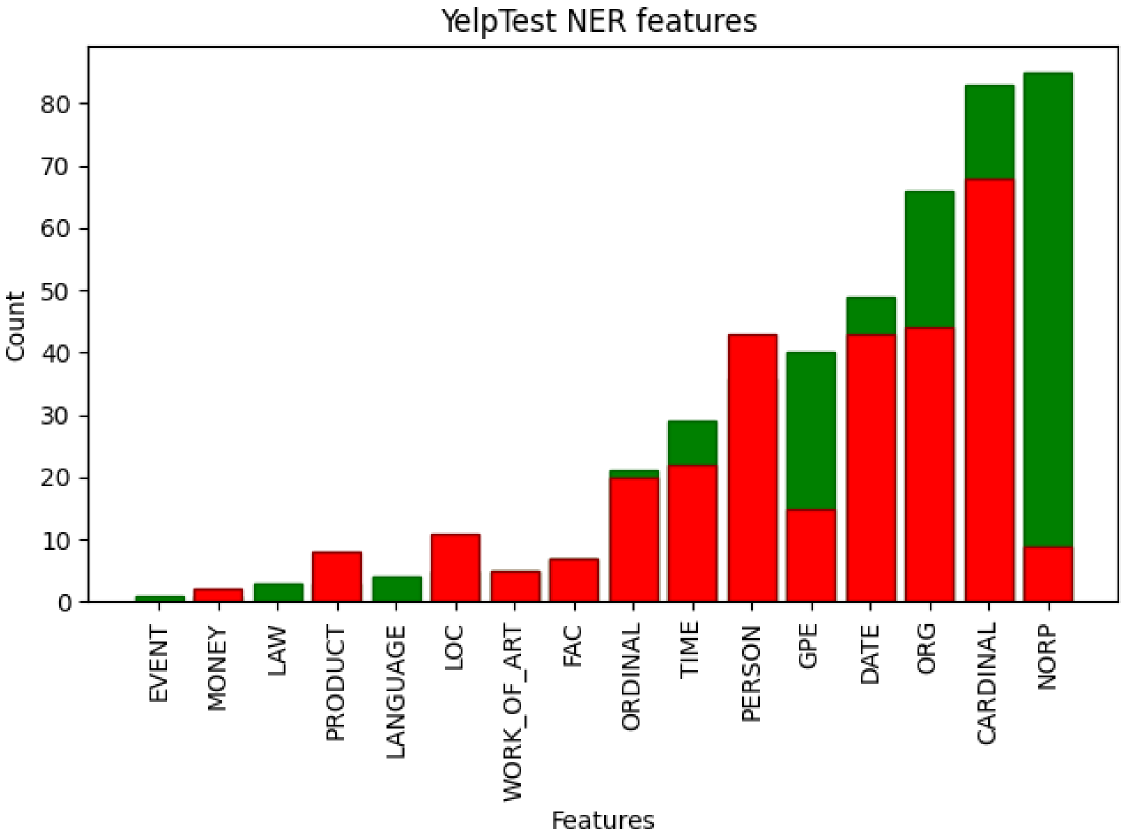
Currently we have 18 attributes : 'NORP', 'CARDINAL', 'DATE', 'ORG', 'LANGUAGE', 'LOC', 'GPE', 'PERSON', 'ORDINAL', 'TIME', 'MONEY','WORK_OF_ART', 'QUANTITY', 'FAC', 'PRODUCT', 'EVENT', 'LAW', 'PERCENT'.

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

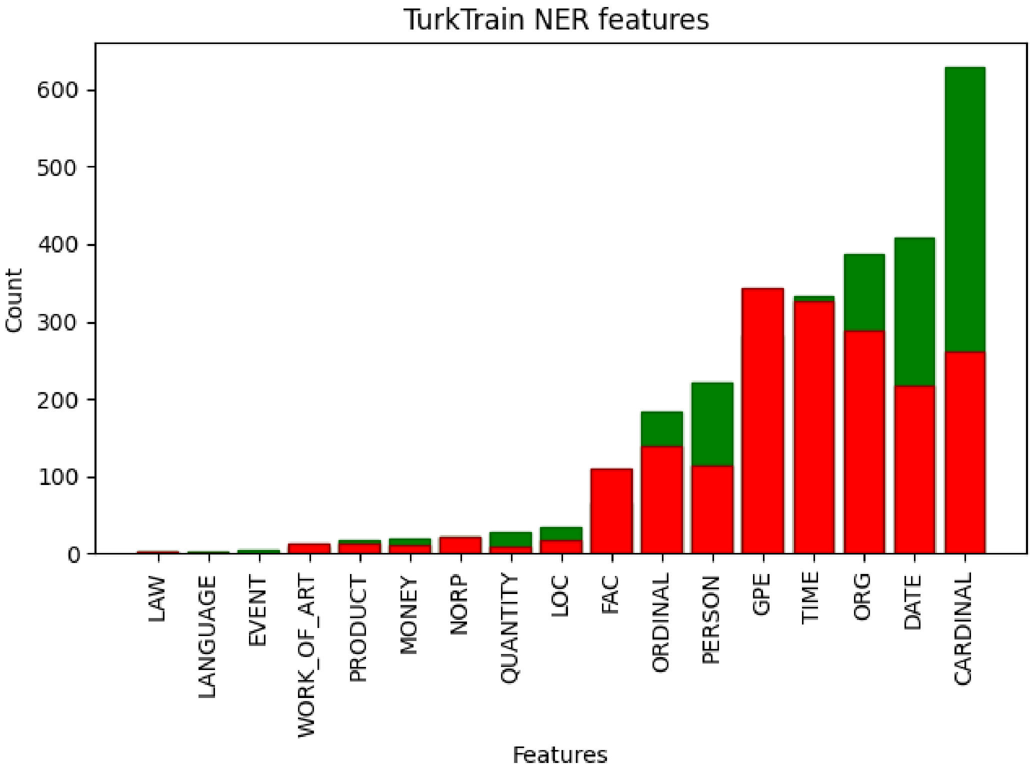
1.1 - Yelp train dataset



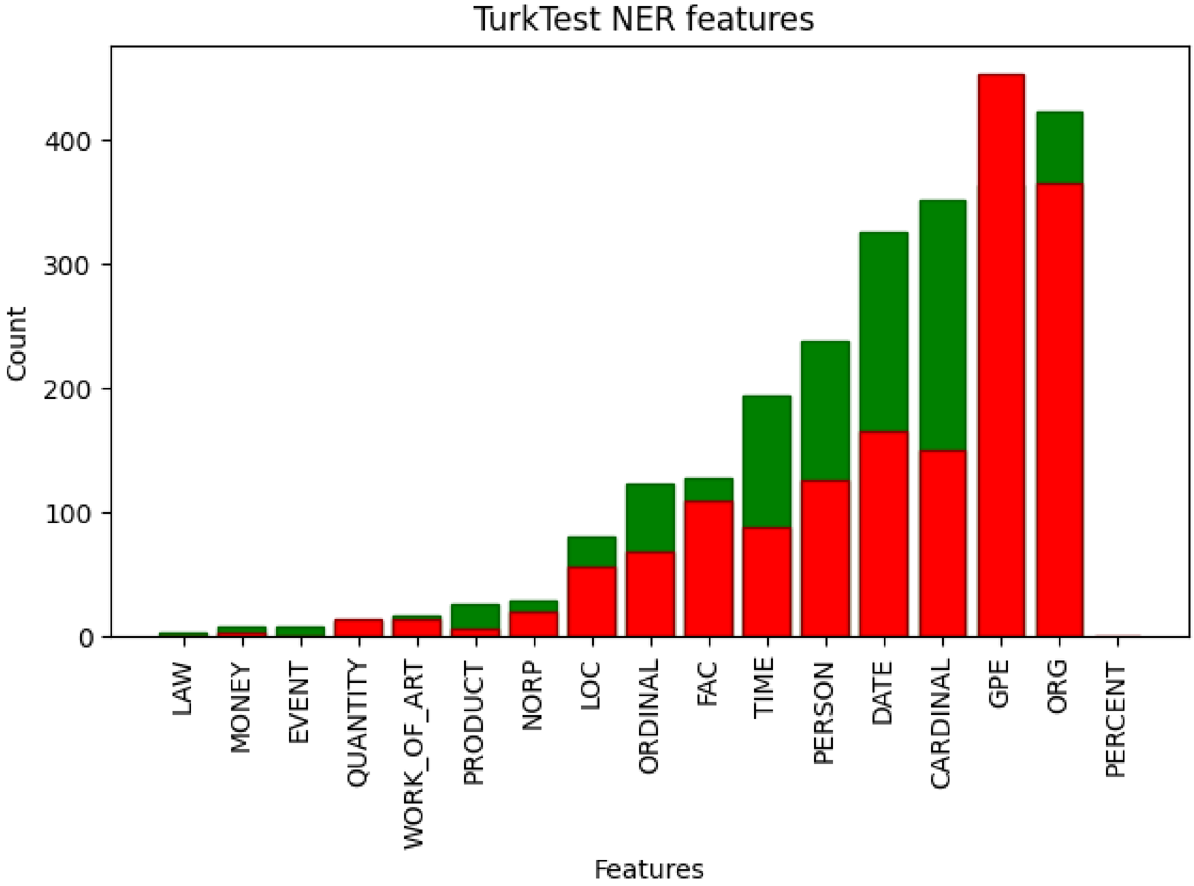
1.2 - Yelp test dataset



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

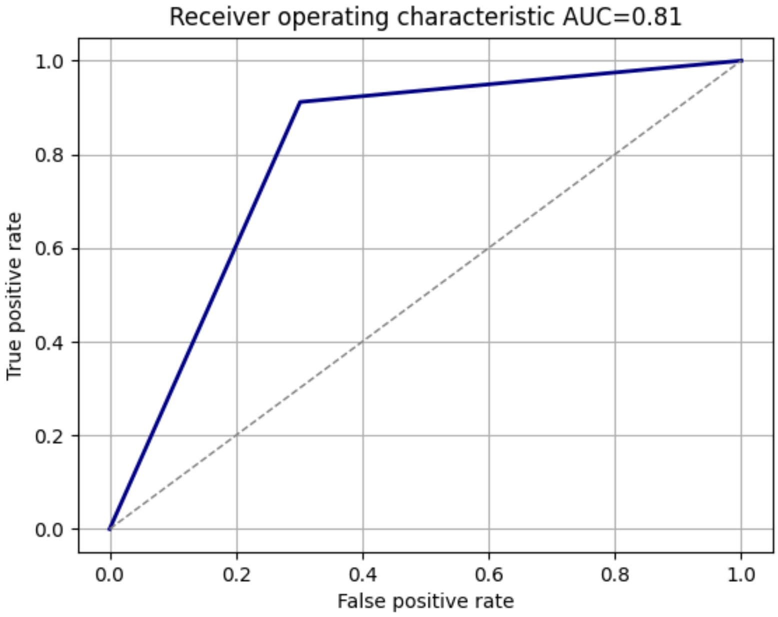
Recall Score: 0.77625

F1 Score: 0.7729114433919926

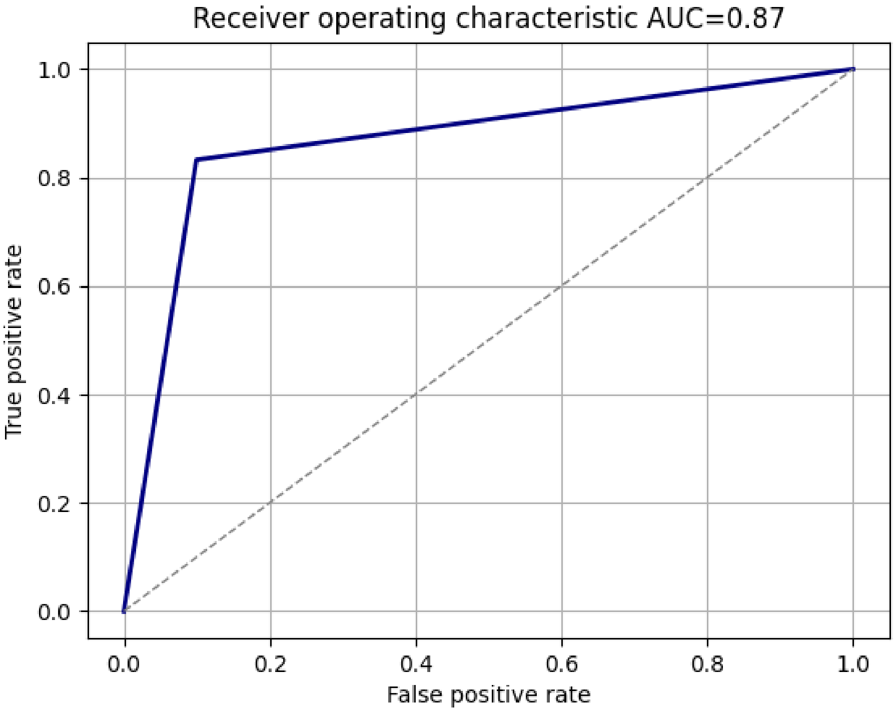
3 - ROC Curve

3.1 - Multinomial Naive Bayes (MNB)

3.1.1 - Yelp dataset

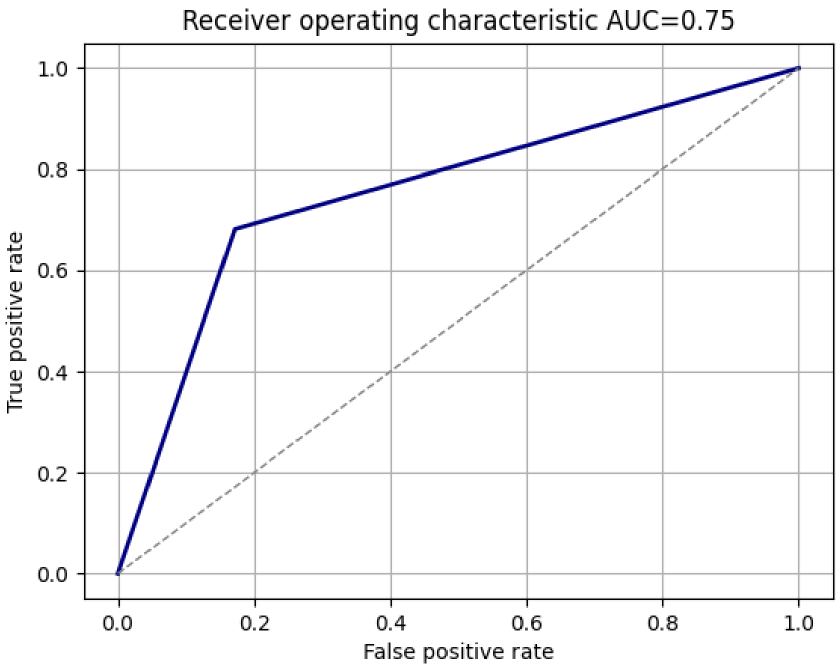


3.1.2 - Turk dataset

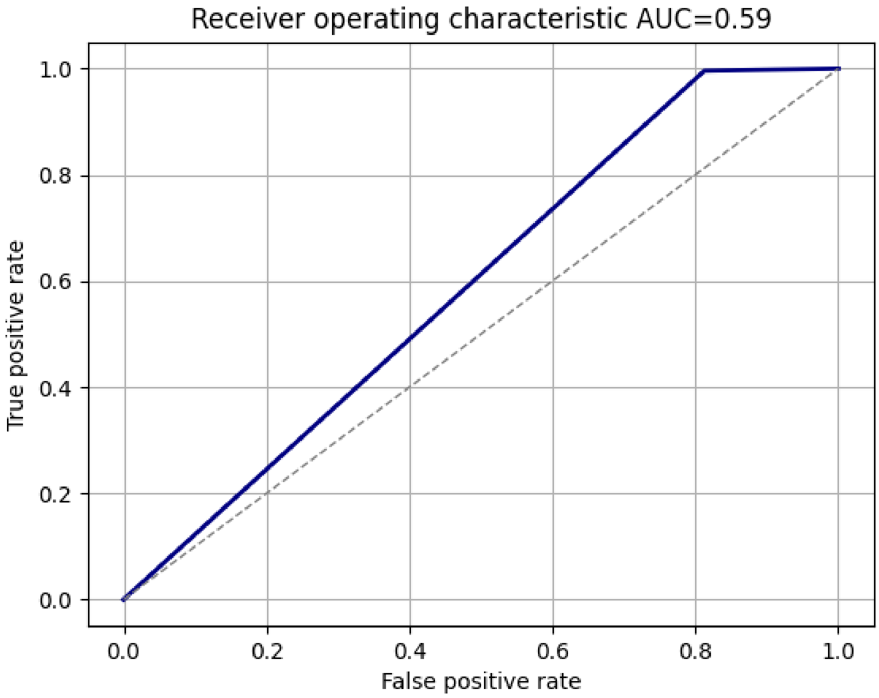


3.2 - Random Forest (RF)

3.2.1 - Yelp dataset

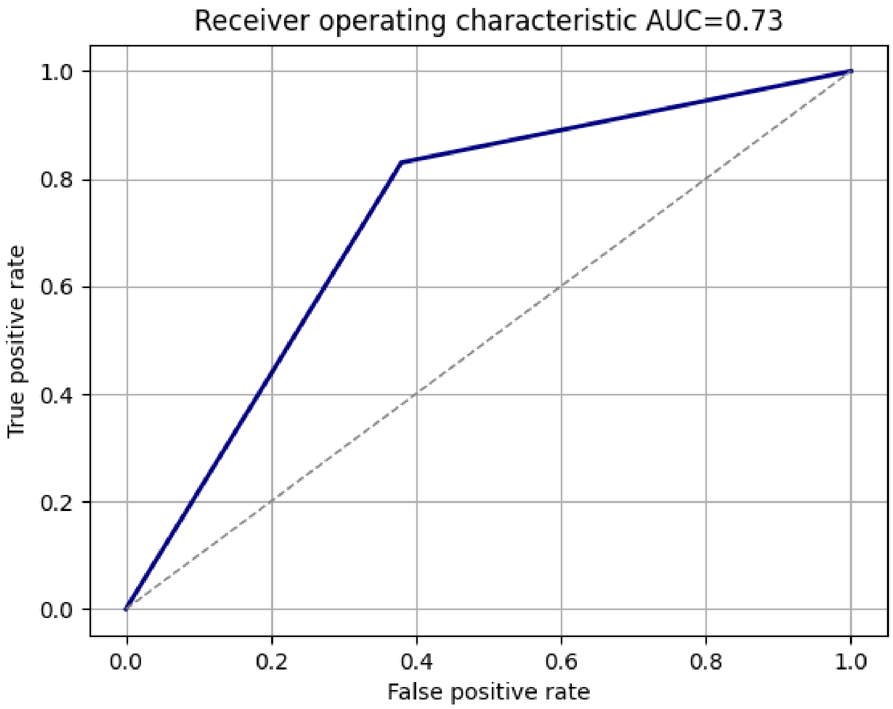


3.2.2 - Turk dataset

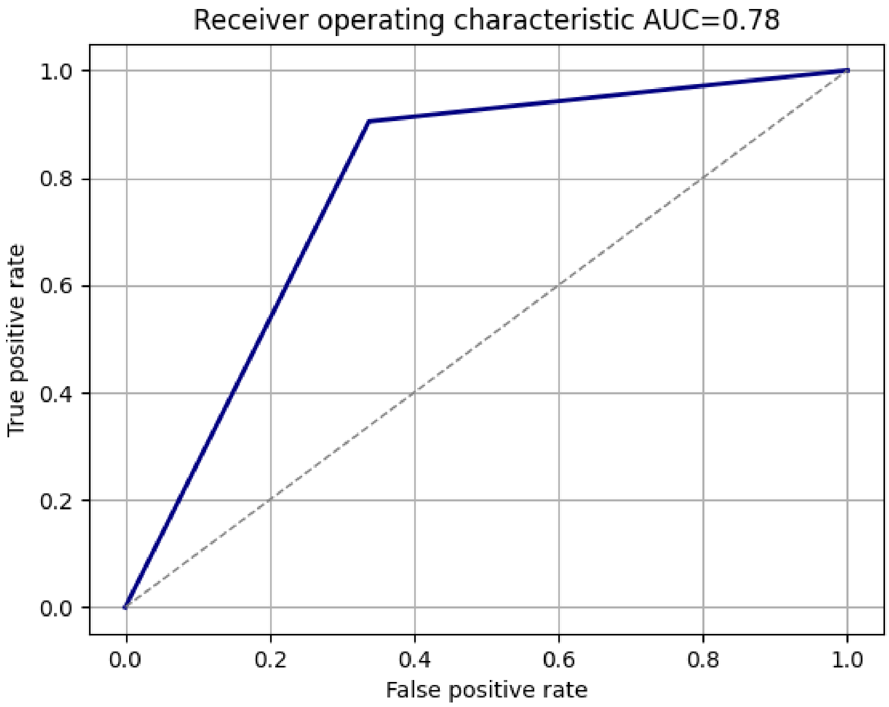


3.3 - Convolutional Neural Network (CNN)

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.