Project Report

Done by :

1- Ali Ahmed Hamed Shaker 20201701725

2- Mina George Raouf Iskander 20201701741

3-Ali Rafeeq Amer 20201701705

4-Diaaelden Amr 20201701720

Preprocessing

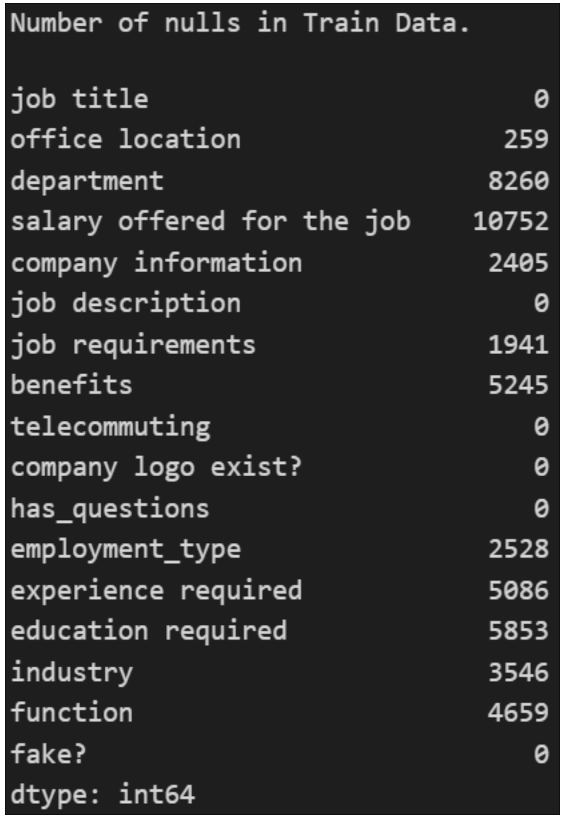
pre-processing techniques

1- Dropping some columns

Drop salary offered for the job column in train data because it has 84 % approx null values.

Drop salary offered for the job column in test data because it has 84 % approx null values.

Also we dropped telecommuting, has_questions, company logo exist? columns because they does not have any logical meaning.



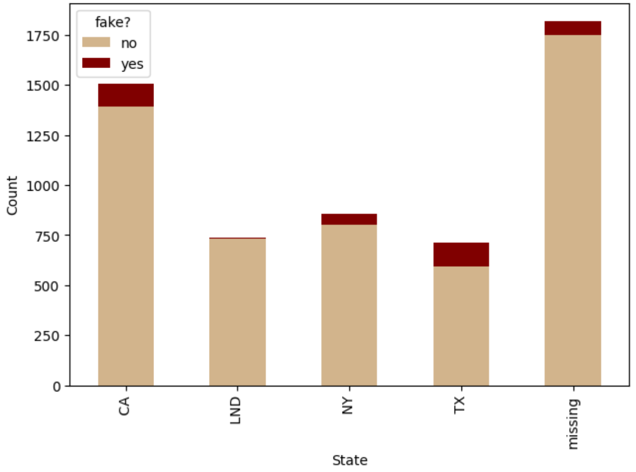
```
Number of nulls in Train Data.

job title                       0
office location               259
department                   8260
salary offered for the job  10752
company information          2405
job description                 0
job requirements             1941
benefits                     5245
telecommuting                   0
company logo exist?             0
has_questions                   0
employment_type              2528
experience required          5086
education required           5853
industry                     3546
function                     4659
fake?                           0
dtype: int64
```

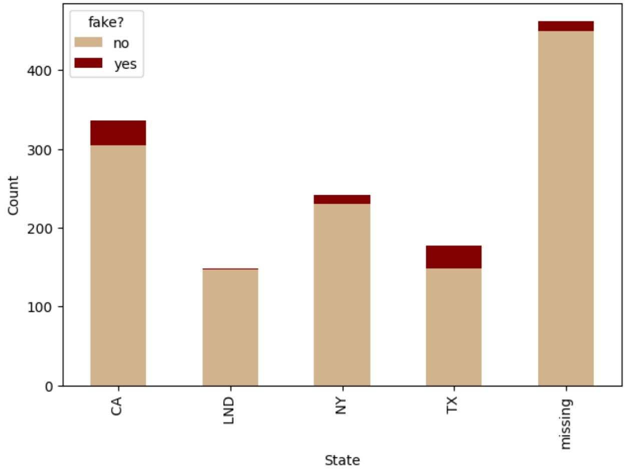
2- Splitting Column

# Split office location to country, state and city

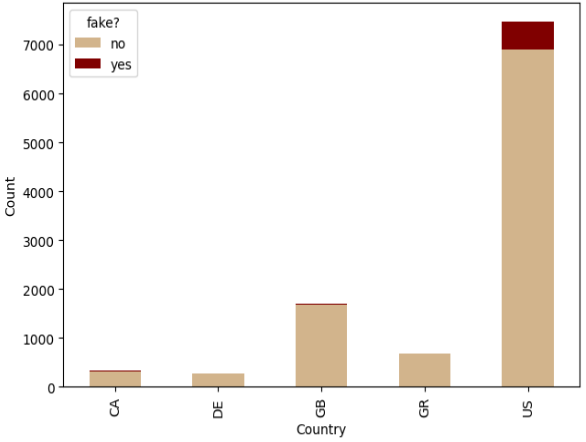
Distribution of Train Data for Fake and Non-Fake Jobs by State (Top 5)



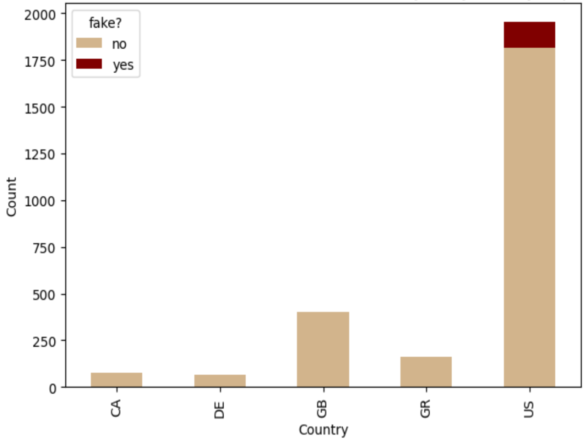
Distribution of Test Data for Fake and Non-Fake Jobs by State (Top 5)



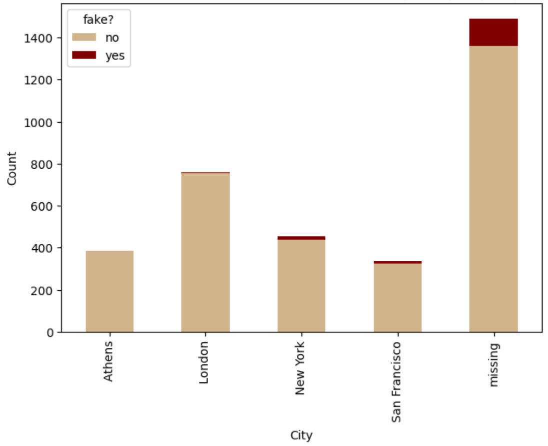
Distribution of Train Data for Fake and Non-Fake Jobs by Country (Top 5)



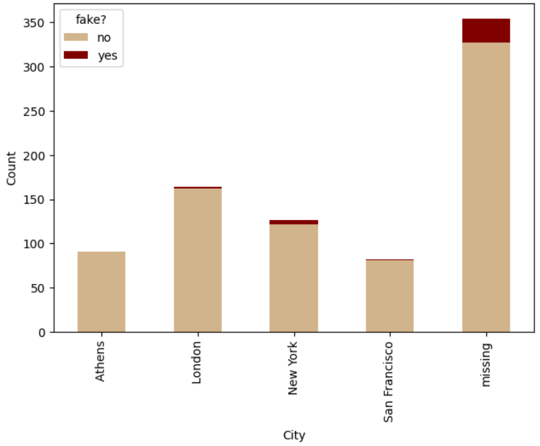
Distribution of Test Data for Fake and Non-Fake Jobs by Country (Top 5)



Distribution of Train Data for Fake and Non-Fake Jobs by City (Top 5)



Distribution of Test Data for Fake and Non-Fake Jobs by City (Top 5)

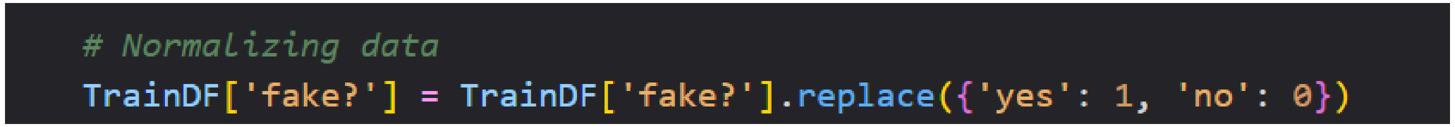


3- Replacing null values

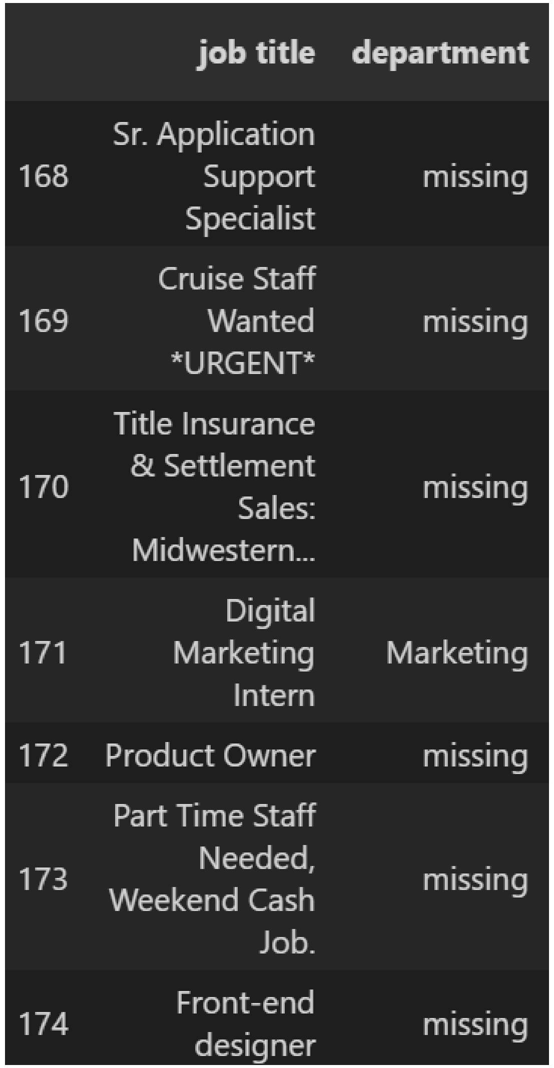
- Replace null values with key term “missing” in all columns
- *Merging text data into one column & deleting Nulls of ignored empty text*

('company information', 'job description', 'job requirements', 'benefits’)

4- Normalization



```
# Normalizing data
TrainDF['fake?'] = TrainDF['fake?'].replace({'yes': 1, 'no': 0})
```

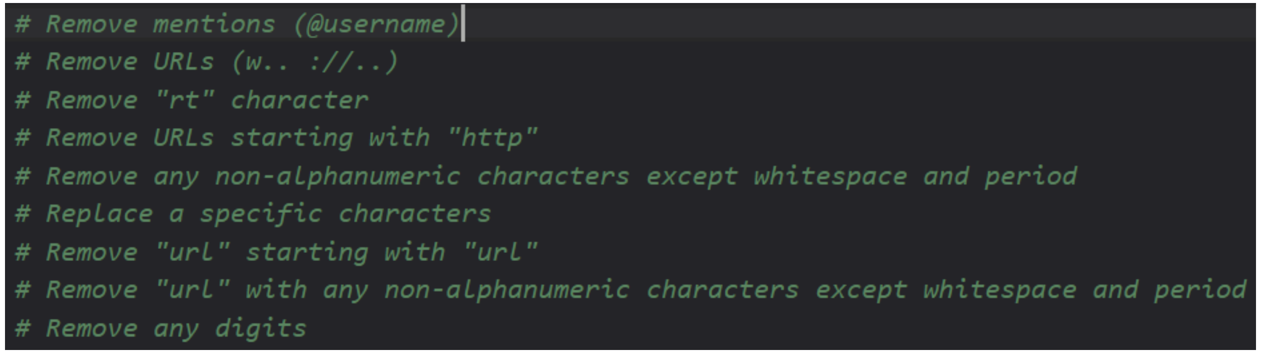


	job title	department
168	Sr. Application Support Specialist	missing
169	Cruise Staff Wanted *URGENT*	missing
170	Title Insurance & Settlement Sales: Midwestern...	missing
171	Digital Marketing Intern	Marketing
172	Product Owner	missing
173	Part Time Staff Needed, Weekend Cash Job.	missing
174	Front-end designer	missing

5- Text cleaning

# Removing special characters and tags using regular expressions

EX:

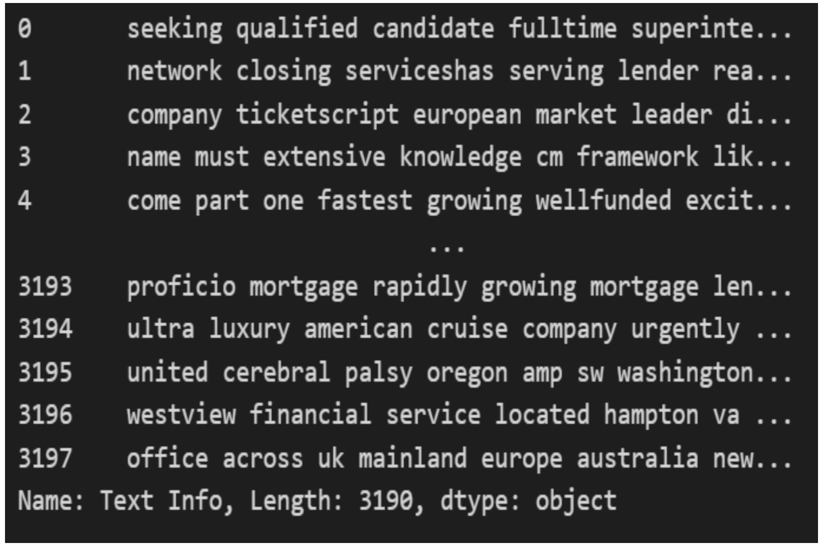


```
# Remove mentions (@username)
# Remove URLs (w.. ://..)
# Remove "rt" character
# Remove URLs starting with "http"
# Remove any non-alphanumeric characters except whitespace and period
# Replace a specific characters
# Remove "url" starting with "url"
# Remove "url" with any non-alphanumeric characters except whitespace and period
# Remove any digits
```

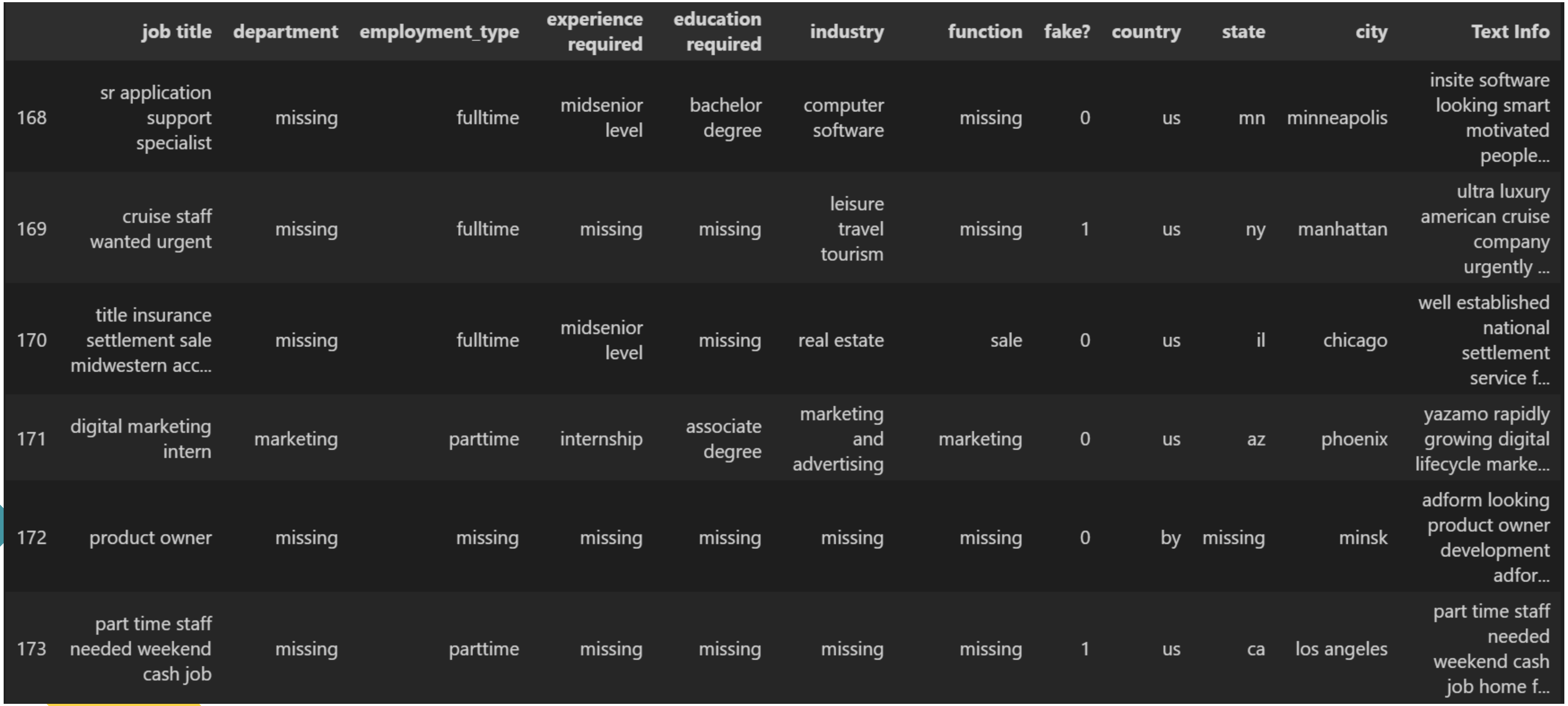
# Remove stop words :

Stop words are commonly used words in a language that are removed from text data during natural language processing as they do not add much meaning to the overall context of the text.

# Lemmatize : **Lemmatization is the process of reducing words to their base or dictionary form, which helps in reducing the complexity of text data and improving text analysis.**

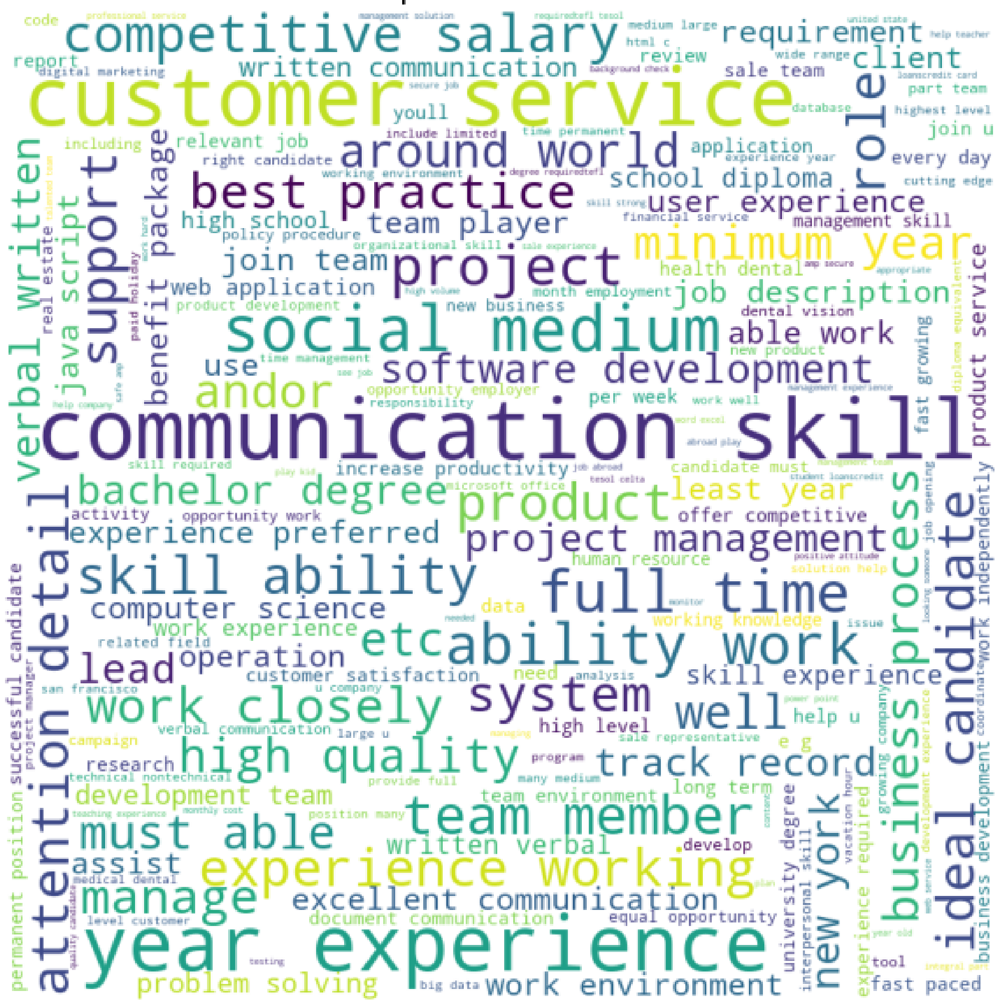


```
0       seeking qualified candidate fulltime superinte...
1       network closing serviceshas serving lender rea...
2       company ticketscript european market leader di...
3       name must extensive knowledge cm framework lik...
4       come part one fastest growing wellfunded excit...
                              ...
3193    proficio mortgage rapidly growing mortgage len...
3194    ultra luxury american cruise company urgently ...
3195    united cerebral palsy oregon amp sw washington...
3196    westview financial service located hampton va ...
3197    office across uk mainland europe australia new...
Name: Text Info, Length: 3190, dtype: object
```

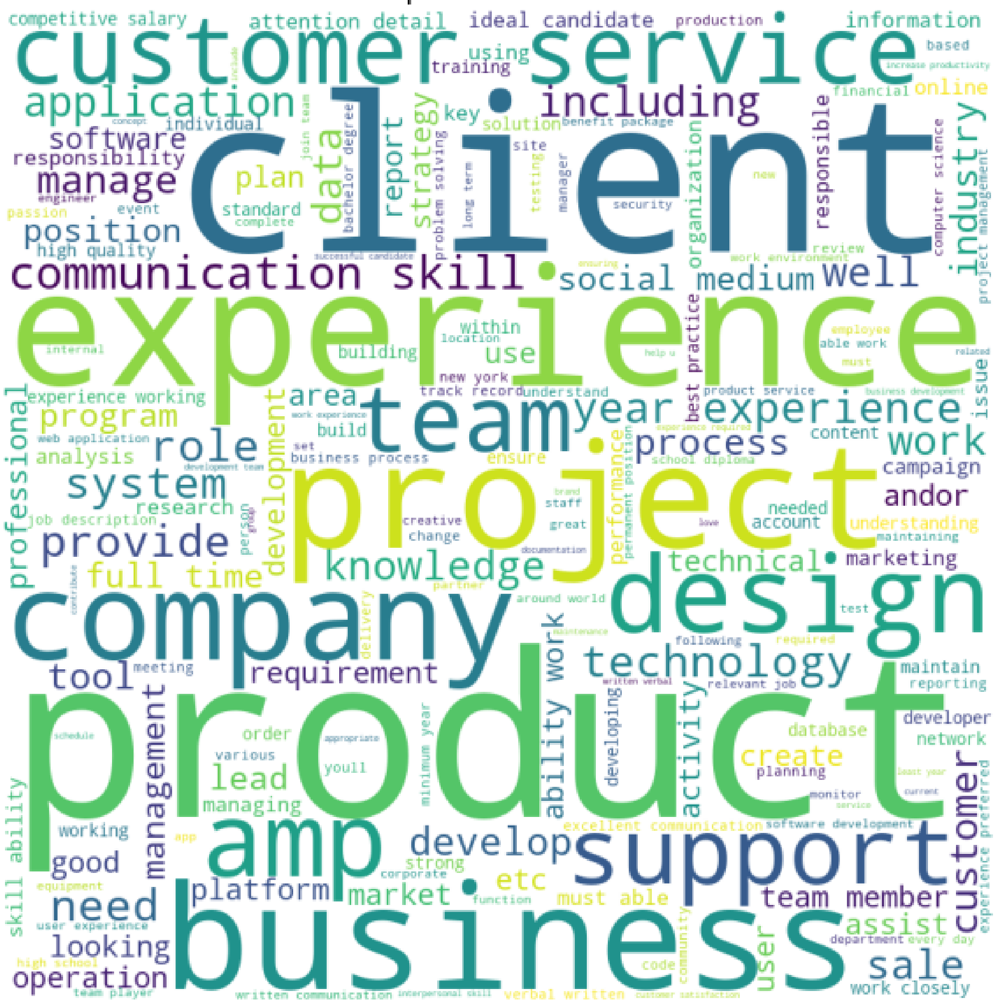


	job title	department	employment_type	experience required	education required	industry	function	fake?	country	state	city	Text Info
168	sr application support specialist	missing	fulltime	midsenior level	bachelor degree	computer software	missing	0	us	mn	minneapolis	insite software looking smart motivated people...
169	cruise staff wanted urgent	missing	fulltime	missing	missing	leisure travel tourism	missing	1	us	ny	manhattan	ultra luxury american cruise company urgently ...
170	title insurance settlement sale midwestern acc...	missing	fulltime	midsenior level	missing	real estate	sale	0	us	il	chicago	well established national settlement service f...
171	digital marketing intern	marketing	parttime	internship	associate degree	marketing and advertising	marketing	0	us	az	phoenix	yazamo rapidly growing digital lifecycle marke...
172	product owner	missing	missing	missing	missing	missing	missing	0	by	missing	minsk	adform looking product owner development adfor...
173	part time staff needed weekend cash job	missing	parttime	missing	missing	missing	missing	1	us	ca	los angeles	part time staff needed weekend cash job home f...

World Cloud



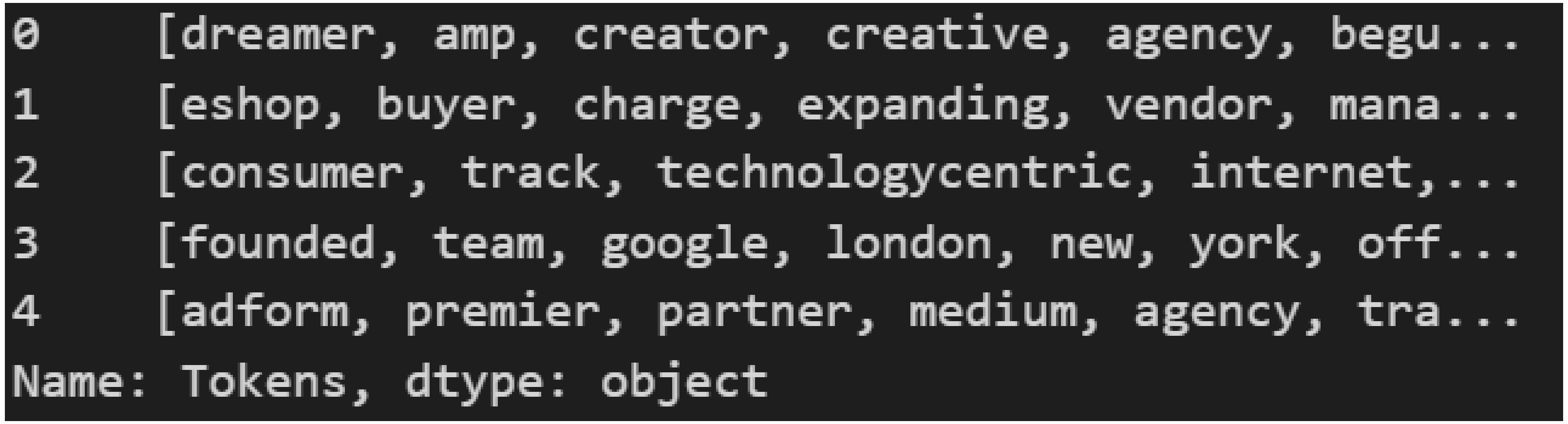
Top Words in Train



Top Words in Test

Feature Extraction

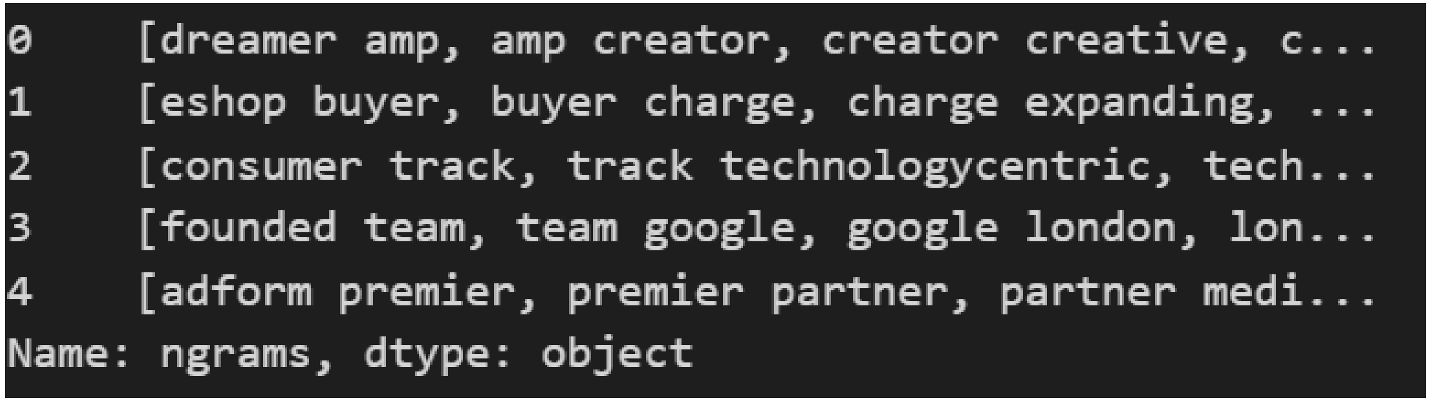
1-Tokenization: **Tokenization is the process of splitting text data into individual units called tokens, which can be words, phrases, or other meaningful elements for natural language processing.**



```
0    [dreamer, amp, creator, creative, agency, begu...
1    [eshop, buyer, charge, expanding, vendor, mana...
2    [consumer, track, technologycentric, internet,...
3    [founded, team, google, london, new, york, off...
4    [adform, premier, partner, medium, agency, tra...
Name: Tokens, dtype: object
```

Feature Extraction

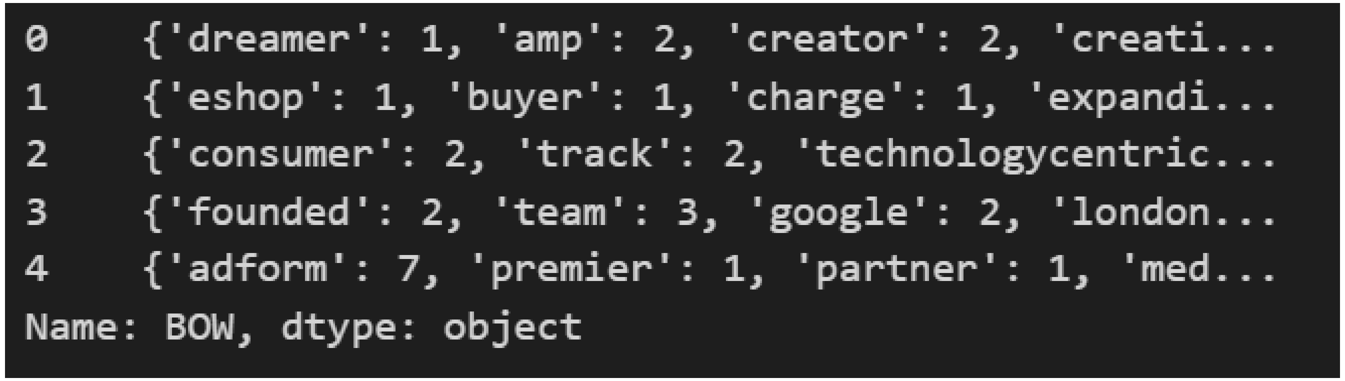
2- ngrams: **N-grams are contiguous sequences of n items (usually words) from a given text, used in natural language processing and text analysis to analyze the frequency and co-occurrence of specific phrases or patterns.**



```
0    [dreamer amp, amp creator, creator creative, c...
1    [eshop buyer, buyer charge, charge expanding, ...
2    [consumer track, track technologycentric, tech...
3    [founded team, team google, google london, lon...
4    [adform premier, premier partner, partner medi...
Name: ngrams, dtype: object
```

Feature Extraction

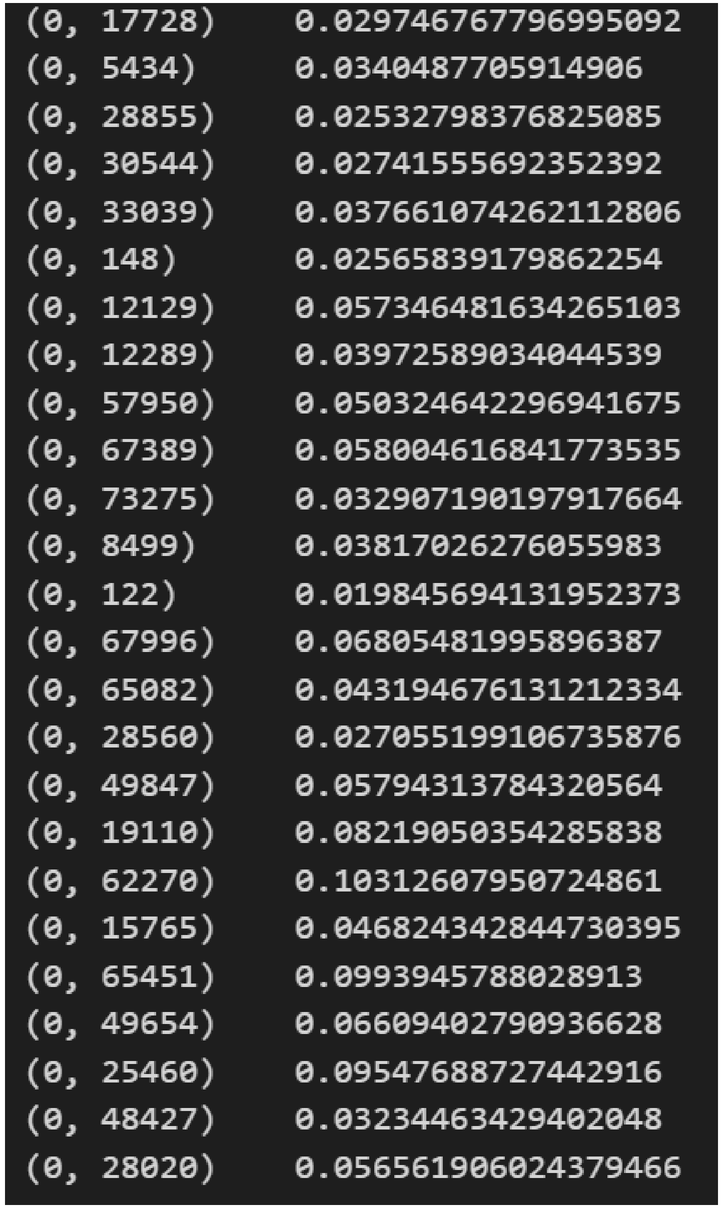
3- Bag of words : **The bag-of-words model is a natural language processing technique that represents text data as a bag (multiset) of its words, disregarding grammar and word order but keeping track of the frequency of each word.**



```
0    {'dreamer': 1, 'amp': 2, 'creator': 2, 'creati...
1    {'eshop': 1, 'buyer': 1, 'charge': 1, 'expandi...
2    {'consumer': 2, 'track': 2, 'technologycentric...
3    {'founded': 2, 'team': 3, 'google': 2, 'london...
4    {'adform': 7, 'premier': 1, 'partner': 1, 'med...
Name: BOW, dtype: object
```

Feature Extraction

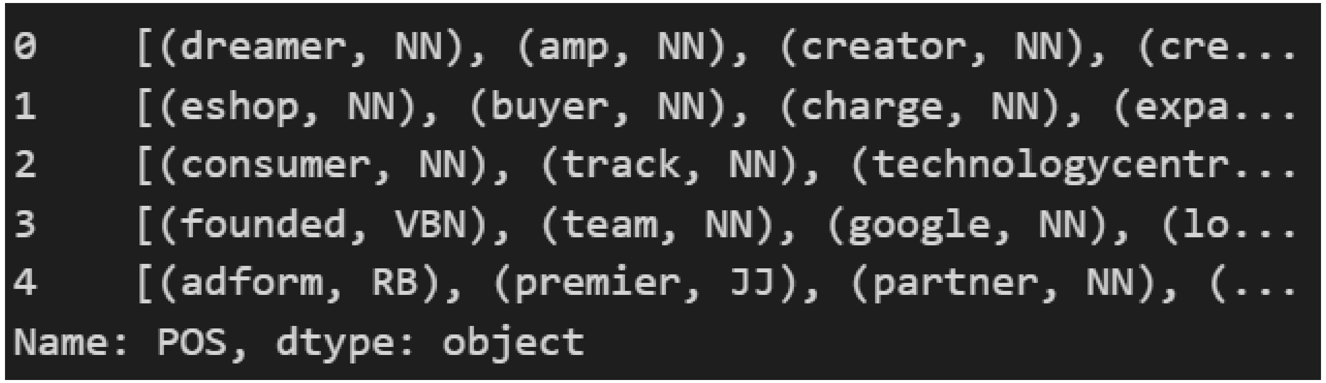
4- TF-IDF: TF-IDF (term frequency-inverse document frequency) is a statistical technique used to evaluate the importance of each word in a document, relative to its occurrence in a corpus of documents.



```
(0, 17728)    0.029746767796995092
(0, 5434)     0.0340487705914906
(0, 28855)    0.02532798376825085
(0, 30544)    0.02741555692352392
(0, 33039)    0.037661074262112806
(0, 148)      0.02565839179862254
(0, 12129)    0.057346481634265103
(0, 12289)    0.03972589034044539
(0, 57950)    0.050324642296941675
(0, 67389)    0.058004616841773535
(0, 73275)    0.032907190197917664
(0, 8499)     0.03817026276055983
(0, 122)      0.019845694131952373
(0, 67996)    0.06805481995896387
(0, 65082)    0.043194676131212334
(0, 28560)    0.027055199106735876
(0, 49847)    0.05794313784320564
(0, 19110)    0.08219050354285838
(0, 62270)    0.10312607950724861
(0, 15765)    0.046824342844730395
(0, 65451)    0.0993945788028913
(0, 49654)    0.06609402790936628
(0, 25460)    0.09547688727442916
(0, 48427)    0.03234463429402048
(0, 28020)    0.056561906024379466
```

Feature Extraction

5- part of speech: **Part of speech refers to the grammatical category that a word belongs to, based on its function and relationship to other words in a sentence, such as nouns, verbs, adjectives, adverbs, prepositions, and conjunctions.**



```
0    [(dreamer, NN), (amp, NN), (creator, NN), (cre...
1    [(eshop, NN), (buyer, NN), (charge, NN), (expa...
2    [(consumer, NN), (track, NN), (technologycentr...
3    [(founded, VBN), (team, NN), (google, NN), (lo...
4    [(adform, RB), (premier, JJ), (partner, NN), (...
Name: POS, dtype: object
```

Models

Models

1-Logistic Regression: a simple and efficient linear model that is widely used for binary classification tasks and can be extended to multiclass problems.

2-Random Forest: an ensemble learning method that combines multiple decision trees to improve the accuracy and robustness of the model for both binary and multiclass classification tasks.

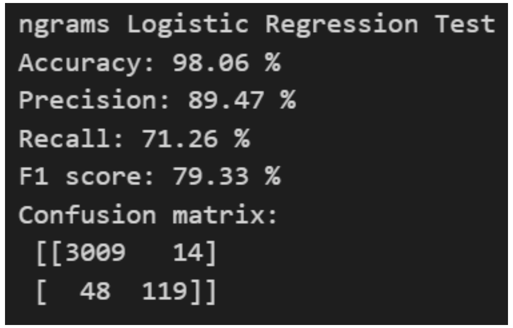
3-CNN (Convolutional Neural Network): a deep learning model that uses convolutional layers to extract features from images or other multidimensional data and learn patterns for classification or regression tasks, and is especially effective for image classification.

4-SVM (Support Vector Machine): a versatile and powerful model that finds a hyperplane to separate data into different classes by maximizing the margin between them, and can be used for both binary and multiclass classification tasks.

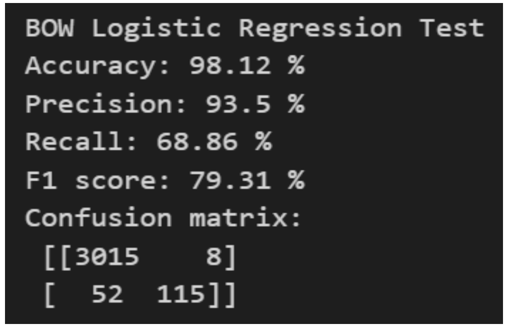
Hyperparameters

- n : (n = 2 *# best parameter for n-grams),* the hyperparameter n is set to 2, which means that the model will consider pairs of adjacent words (bigrams) when creating n-grams. This is a common choice for text classification tasks, as bigrams can capture some of the contextual information and the relationship between adjacent words that unigrams (single words) may miss.
- Batch size : When comparing two CNN models with different hyperparameters, specifically batch size and filter size, it was observed that a model with a larger batch size of 100 and filter size of 64 achieved a lower training time compared to a model with a smaller batch size of 60 and filter size of 32. Furthermore, the model with the larger batch size also achieved higher accuracy results.
- *n_estimator: It* has been observed that setting the 'n_estimators' hyperparameter to 100 in a Random Forest model resulted in higher accuracy compared to other values of this hyperparameter, specifically n_estimators=60 and n_estimators=150.
- Number of epochs: After experimenting with different values of the 'epochs' hyperparameter in a CNN model, specifically with values of 15 and 12 epochs, it was observed that the accuracy remained unchanged after exceeding 10 epochs.

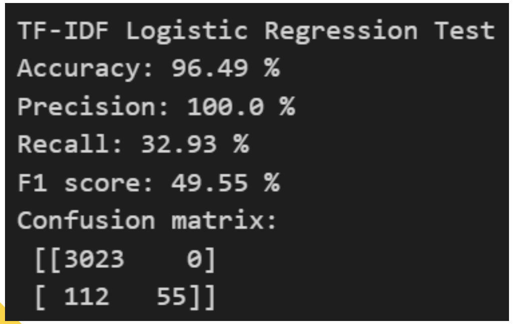
Logistic Regression



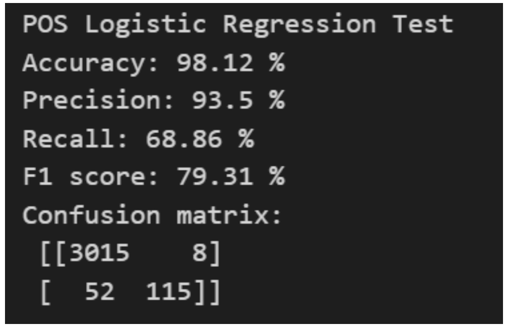
```
ngrams Logistic Regression Test
Accuracy: 98.06 %
Precision: 89.47 %
Recall: 71.26 %
F1 score: 79.33 %
Confusion matrix:
 [[3009   14]
 [  48  119]]
```



```
BOW Logistic Regression Test
Accuracy: 98.12 %
Precision: 93.5 %
Recall: 68.86 %
F1 score: 79.31 %
Confusion matrix:
 [[3015    8]
 [  52  115]]
```

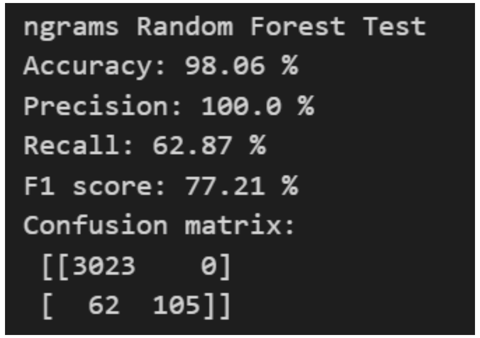


```
TF-IDF Logistic Regression Test
Accuracy: 96.49 %
Precision: 100.0 %
Recall: 32.93 %
F1 score: 49.55 %
Confusion matrix:
 [[3023    0]
 [ 112   55]]
```

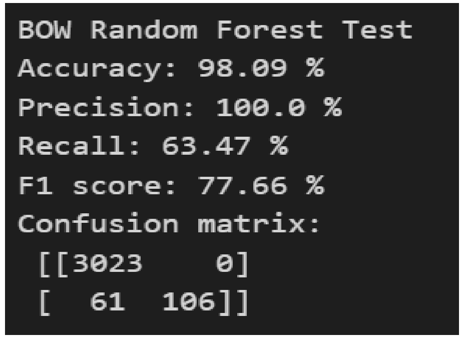


```
POS Logistic Regression Test
Accuracy: 98.12 %
Precision: 93.5 %
Recall: 68.86 %
F1 score: 79.31 %
Confusion matrix:
 [[3015    8]
 [  52  115]]
```

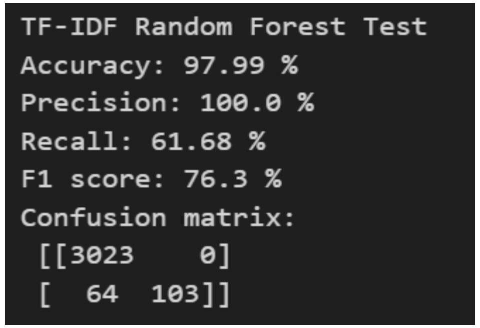
Random Forest



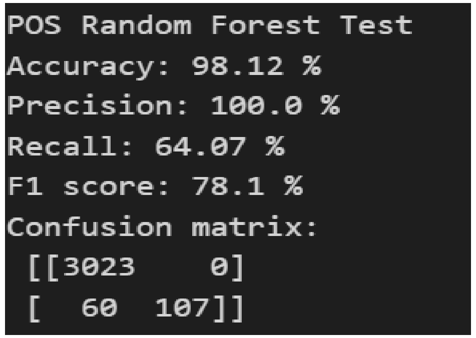
```
ngrams Random Forest Test
Accuracy: 98.06 %
Precision: 100.0 %
Recall: 62.87 %
F1 score: 77.21 %
Confusion matrix:
 [[3023    0]
 [  62  105]]
```



```
BOW Random Forest Test
Accuracy: 98.09 %
Precision: 100.0 %
Recall: 63.47 %
F1 score: 77.66 %
Confusion matrix:
 [[3023    0]
 [  61  106]]
```

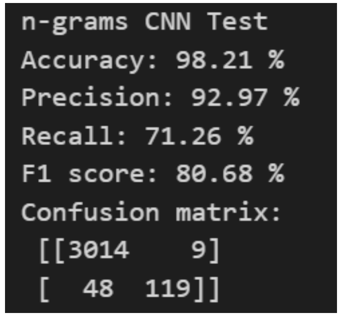


```
TF-IDF Random Forest Test
Accuracy: 97.99 %
Precision: 100.0 %
Recall: 61.68 %
F1 score: 76.3 %
Confusion matrix:
 [[3023    0]
 [  64  103]]
```

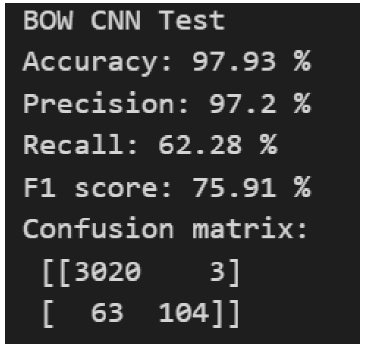


```
POS Random Forest Test
Accuracy: 98.12 %
Precision: 100.0 %
Recall: 64.07 %
F1 score: 78.1 %
Confusion matrix:
 [[3023    0]
 [  60  107]]
```

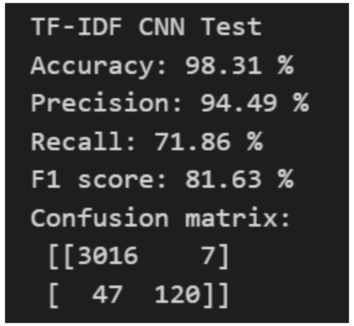
CNN



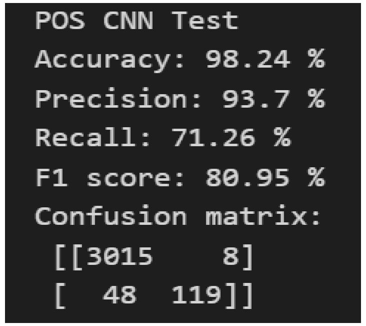
```
n-grams CNN Test
Accuracy: 98.21 %
Precision: 92.97 %
Recall: 71.26 %
F1 score: 80.68 %
Confusion matrix:
 [[3014    9]
 [  48  119]]
```



```
BOW CNN Test
Accuracy: 97.93 %
Precision: 97.2 %
Recall: 62.28 %
F1 score: 75.91 %
Confusion matrix:
 [[3020    3]
 [  63  104]]
```

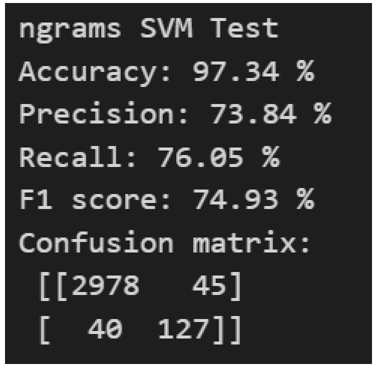


```
TF-IDF CNN Test
Accuracy: 98.31 %
Precision: 94.49 %
Recall: 71.86 %
F1 score: 81.63 %
Confusion matrix:
 [[3016    7]
 [  47  120]]
```

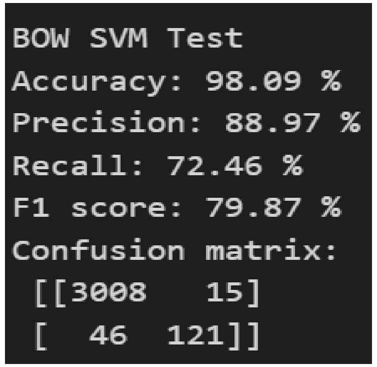


```
POS CNN Test
Accuracy: 98.24 %
Precision: 93.7 %
Recall: 71.26 %
F1 score: 80.95 %
Confusion matrix:
 [[3015    8]
 [  48  119]]
```

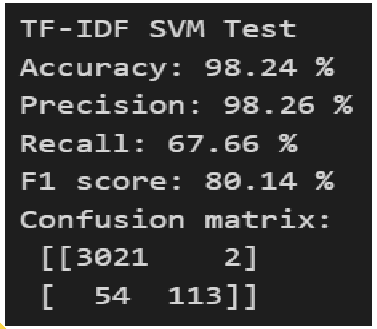
SVM



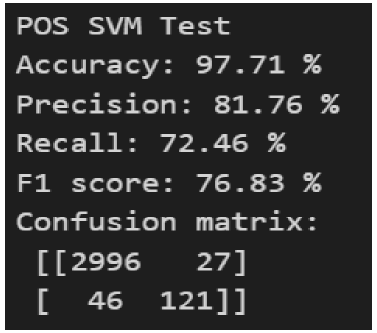
```
ngrams SVM Test
Accuracy: 97.34 %
Precision: 73.84 %
Recall: 76.05 %
F1 score: 74.93 %
Confusion matrix:
 [[2978   45]
 [  40  127]]
```



```
BOW SVM Test
Accuracy: 98.09 %
Precision: 88.97 %
Recall: 72.46 %
F1 score: 79.87 %
Confusion matrix:
 [[3008   15]
 [  46  121]]
```



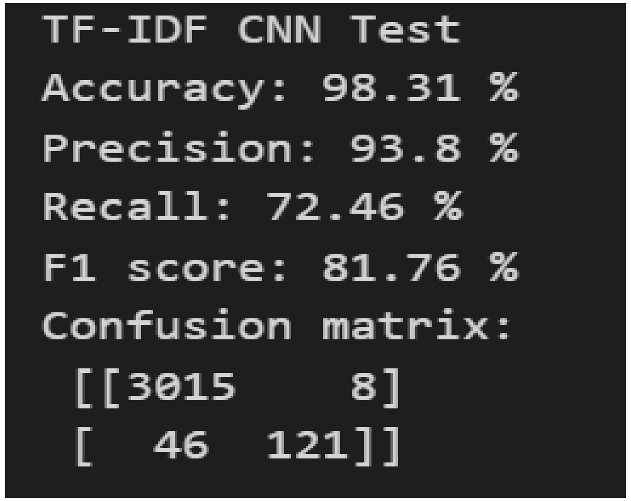
```
TF-IDF SVM Test
Accuracy: 98.24 %
Precision: 98.26 %
Recall: 67.66 %
F1 score: 80.14 %
Confusion matrix:
 [[3021    2]
 [  54  113]]
```



```
POS SVM Test
Accuracy: 97.71 %
Precision: 81.76 %
Recall: 72.46 %
F1 score: 76.83 %
Confusion matrix:
 [[2996   27]
 [  46  121]]
```

Models Evaluation

- **Best Model is " CNN " with combination with " TF-IDF "**



```
TF-IDF CNN Test
Accuracy: 98.31 %
Precision: 93.8 %
Recall: 72.46 %
F1 score: 81.76 %
Confusion matrix:
 [[3015    8]
 [  46  121]]
```

Thank you