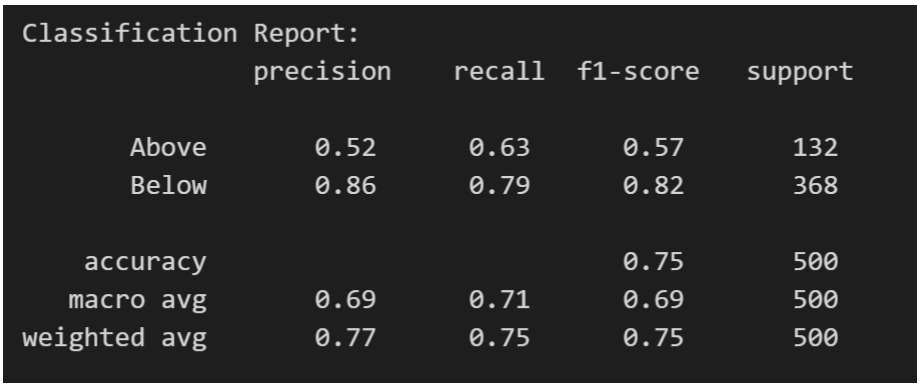
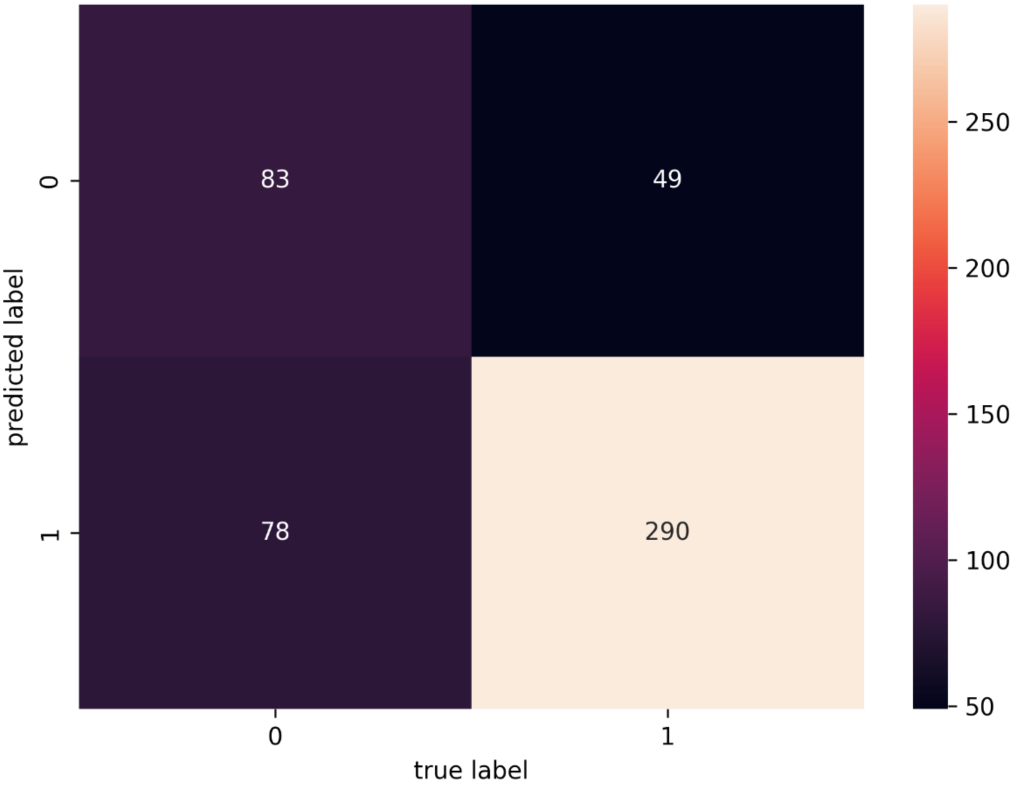
The classification report is:



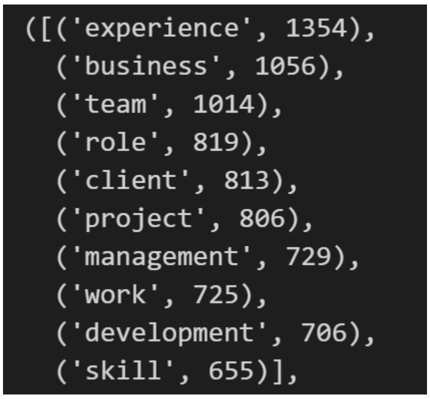
```
Classification Report:
              precision    recall  f1-score   support

       Above       0.52      0.63      0.57       132
       Below       0.86      0.79      0.82       368

    accuracy                           0.75       500
   macro avg       0.69      0.71      0.69       500
weighted avg       0.77      0.75      0.75       500
```

And the confusion matrix is:



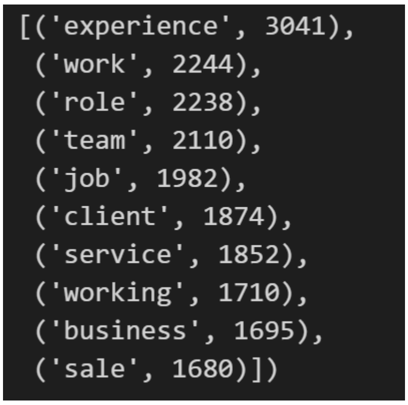
And the model accuracy is : 0.746

Top 10 words of high salary is:



```
([('experience', 1354),
  ('business', 1056),
  ('team', 1014),
  ('role', 819),
  ('client', 813),
  ('project', 806),
  ('management', 729),
  ('work', 725),
  ('development', 706),
  ('skill', 655)],
```

And top 10 words of low salary is:



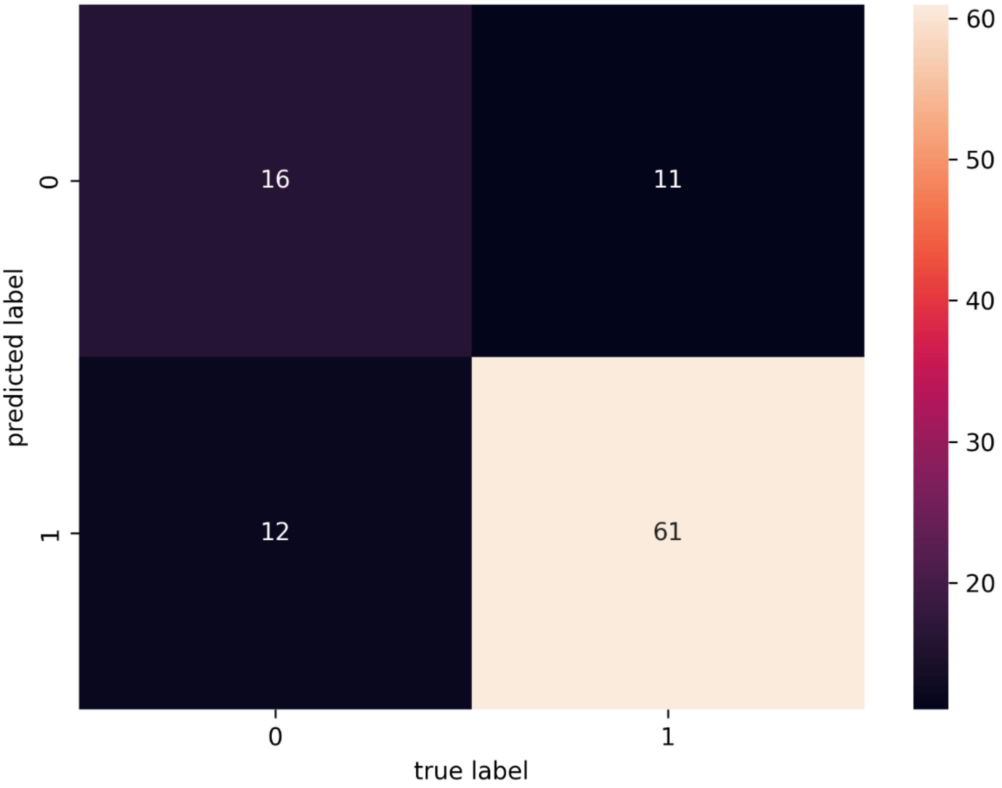
```
[('experience', 3041),
 ('work', 2244),
 ('role', 2238),
 ('team', 2110),
 ('job', 1982),
 ('client', 1874),
 ('service', 1852),
 ('working', 1710),
 ('business', 1695),
 ('sale', 1680)])
```

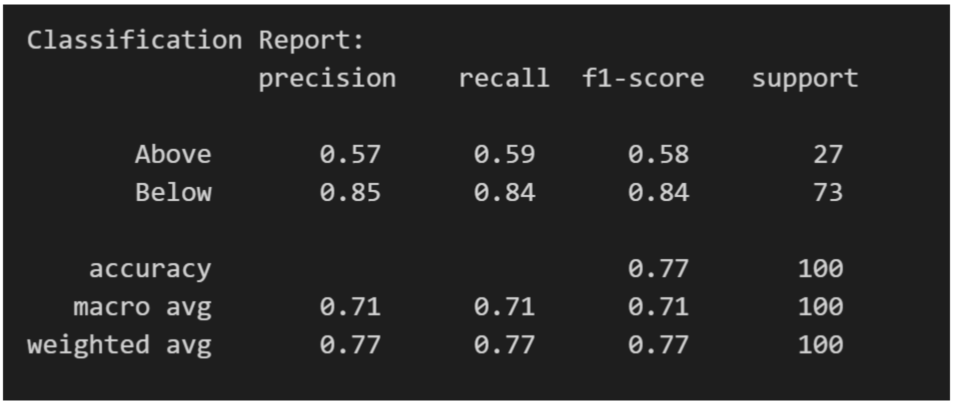
To improve the model accuracy, we could use:

N-gram: Trigrams are based on three words, whereas bigrams are tokens of words based on two words each. These can be used in place of our unigram method to test whether accuracy is improved.

Extra Features: In addition to our earlier method, features like as Job Title, Location, Contract Type, and Category may be provided for every job posting. Since our salary projections are only dependent on text data, our model's performance may be greatly enhanced by include details like seniority, department, location, and full-time/part-time employment.

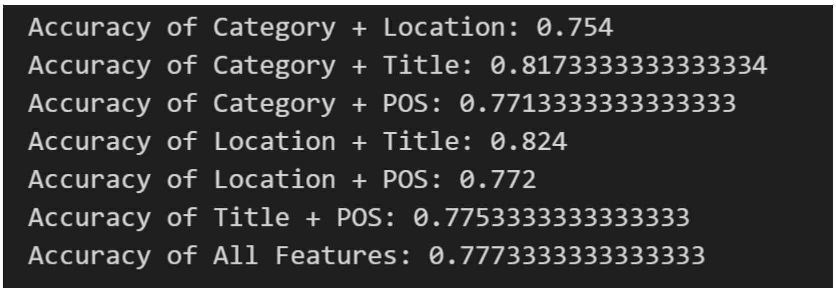
Therefore, I used the trigrams and bigrams combination, the accuracy improved from 0.746 to 0.77





```
Classification Report:
               precision    recall  f1-score   support

       Above       0.57      0.59      0.58        27
       Below       0.85      0.84      0.84        73

    accuracy                           0.77       100
   macro avg       0.71      0.71      0.71       100
weighted avg       0.77      0.77      0.77       100
```

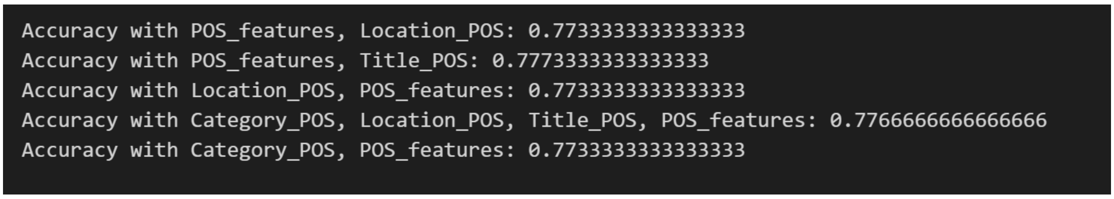
Then I incorporate the variables “Title”,”Location” and “Category” with larger dataset (7500 rows) into the classification model for accuracy improvement, the result is like:



```
Accuracy of Category + Location: 0.754
Accuracy of Category + Title: 0.8173333333333334
Accuracy of Category + POS: 0.7713333333333333
Accuracy of Location + Title: 0.824
Accuracy of Location + POS: 0.772
Accuracy of Title + POS: 0.7753333333333333
Accuracy of All Features: 0.7773333333333333
```

which POS means the FullDescription after tagging.

Then, I tried using POS tagging for those extra variables, but the accuracy wasn’t improved to great extent.



```
Accuracy with POS_features, Location_POS: 0.7733333333333333
Accuracy with POS_features, Title_POS: 0.7773333333333333
Accuracy with Location_POS, POS_features: 0.7733333333333333
Accuracy with Category_POS, Location_POS, Title_POS, POS_features: 0.7766666666666666
Accuracy with Category_POS, POS_features: 0.7733333333333333
```