What features do you use in your classifier?
There are seven features in my project.
Features: the number of the vowel, prefix, suffix, the last three letters, the combination of the vowel, the last combination of the two vowels, and the type of the word.
Why are they important and what information do you expect them to capture?
The number of the vowel: I find that the words with different number of the vowel has different characteristic, so the first step, I set a feature that indicate the number of the vowel of that word.
The last three letters: Firstly, if the last letter is "S" or "D", I will delete that. Sometimes the last three letters perform like the suffix, and I delete the suffix that appear less than 30 times.
The combination of the vowel: I record the combination of the vowels in the word, and by recoding to the position of each vowel, we can know that some vowel will become stress after some vowels.
The last combination of the two vowels: this feature mostly contributes to find the last vowel which should be stressed.

How do you experiment and improve your classifier?
I using decision tree classifier, and put 80% of the data as training data, and put 20% of the data as the test data, and then by comparing the prediction of the stress position of the test data with the true stress position of the test data, I could know that how many correct stress position I have found.
When I experiment, I write some code:

```
prediction = list(clf.predict(x_test))
ground_truth = list(y_test)
precision = [0,0,0,0]
count = [0,0,0,0]
for i in range(len(ground_truth)):
    if ground_truth[i] == prediction[i]:
        precision[ground_truth[i] - 1] +=1
    count[ground_truth[i] - 1] += 1
for j in range(len(count)):
    precision[j] = precision[j]/count[j]
print(count)
print(precision)
print(f1_score(ground_truth,prediction,average = 'macro'))
```

By using the above code, I could know the percentage of the correct stress I have found in four different vowel position. And I find that when the stress at the forth vowel of the word, the correct rate is extremely low, so if I want to increase the F1 score, I need to find method to increase the probability that I could predict the forth vowel correct. And I found that when I use the encode method like old = old*100+new, the performance is better.
At first, I use the features like below:
The prefix: prefixes=('COUNTER', 'INTER', 'NITRO', 'UNDER', 'AERO', 'DEMO', 'IDIO', 'OVER', 'SOCI', 'TELE', 'BIO', 'COM', 'LAV', 'LEG', 'MIS', 'NAI', 'NAT')
I find that these prefixes will change the stress position.
The suffix: suffixes=('THELESS', 'TATIVES', 'MINATE', 'UNDERREPORT', 'COMEDIENNE', 'DULATE', 'JAHIDEEN', 'NERATE', 'NASIONAL', 'SCRIBE', 'ACKED', 'ANCED', 'MINES', 'NEERS', 'NOSED', 'POSED', 'SENTS', 'TANDS', 'TIANE', 'UILLE', 'VERSE', 'ADOR', 'ELLE', 'ENDS', 'ETTE', 'EURS', 'EVAN', 'IBED', 'LIED',

'MAIN', 'NECT', 'NEER', 'NOSE', 'PATH', 'PPLY', 'TECT', 'TUNE', 'WEES', 'DAD', 'ERU', 'ETE', 'EUR', 'JAN', 'LET', 'MAR', 'TIF', 'TIK', 'YOR')
Also, I find these suffixes will change the stress position.
But after several experiment, I use a new feature that I firstly scan all training data and put the last three letter without the last letter if the last letter is “S” or “D” into a dictionary, and delete the keys that appear less than 30 times. And this feature is better.
I also use the nltk at the early version because I think that most of the verb will stress at the second position, but after the experiment I find this may overfit so I also delete this feature.
The type of the word: I find that most of the noun word stress at the first position, but most of the verb are stressed at the second position.