

< [Back to Artificial Intelligence Nanodegree and Specializations](#)

# Build a Sign Language Recognizer

审阅

代码审阅

HISTORY

Requires Changes

还需满足 2 个要求 变化

Hello Student,

The first submission here is not bad, however, there were something required to amend in the `my_model_selectors.py` evaluation functions. I am sure you could easily get it done soon.

Have a Good Day and nice AI Learning!

## PART 1: Data

1. Student provides correct alternate feature sets: delta, polar, normalized, and custom.
2. Student passes unit tests.
3. Student provides a reasonable explanation for what custom set was chosen and why (Q1).

Bravo! You did a fantastic job to preprocess the data and it explained why the data preprocessing technique was chosen. I am really impressed since all of the features are declared by using `for loop`, I rarely see students would use it to achieve code simplicity. Nice!

## PART 2: Model Selection

1. Student correctly implements CV, BIC, and DIC model selection techniques in

"my\_model\_selectors.py".

2. Student code runs error-free in notebook, passes unit tests and code review of the algorithms.
3. Student provides a brief but thoughtful comparison of the selectors (Q2).

It is fabulous that all of the unit tests can be passed in the first submission, however, it is an interface only. And I found out the BIC and DIC functions set up have some errors.

## Requires Amendments

1). The BIC evaluation function ( $BIC = -2 \log L + p \log N$ ) is not quite correct as the parameter  $p$ 's value is a bit deviated from what we expected and here is the explanation.

"Free parameters" are parameters that are learned by the model and it is a sum of:

1. The free transition probability parameters, which is the size of the transition matrix less one row because they add up to 1 and therefore the final row is deterministic, so  $n*(n-1)$
2. The free starting probabilities, which is the size of startprob minus 1 because it adds to 1.0 and last one can be calculated so  $n-1$
3. Number of means, which is  $n*f$
4. Number of covariances which is the size of the covars matrix, which for "diag" is  $n*f$

So it is equivalent to  $n^2 + 2*n*f - 1$

And here is a [linkage](#) for a further study. And the free parameters were discussed on page 165.

- Please amend `SelectorBIC` function.

2). In the DIC function, the anti\_log\_likelihood score should not be calculated within the for loop:

```
for word in all_words:
    sum_scores = 0.0
    if word != self.this_word:
        X, lengths = self.hwords[word]
        sum_scores += hmm_model.score(X, lengths)
        DIC = logI + 1.0/(M-1)*sum_scores #<< ---- Please
take it out from the for loop.
```

- Please correct the DIC anti\_log\_likelihood score algorithm.

## PART 3: Recognizer

1. Student implements a recognizer in "my\_recognizer.py" which runs error-free in the notebook and passes all unit tests

2. Student provides three examples of feature-selector combinations in the submission cells of the notebook.
3. Student code provides the correct words within <60% WER for at least one of the three examples student provided.
4. Student provides a summary of results and speculates on how to improve the WER.

It is awesome that the `Recognizer Unit Tests` are passed, however, we are looking forward to a new result after the `BIC & DIC` evaluation function amendment.

## Requires Amendments

- Please update this section after the evaluation function `BIC & DIC` amended.

 重新提交

 下载项目



## 重新提交项目的最佳做法

Ben 与你分享修改和重新提交的 5 个有益的小贴士。

 [观看视频 \(3:01\)](#)

[返回 PATH](#)

---

[学员 FAQ](#)