

[Return to Classroom](#)

# Combine Signals for Enhanced Alpha

## REVIEW

## CODE REVIEW

## HISTORY

### Requires Changes

#### 1 specification requires changes

Great job, you are almost there! 🎉 Clearly, you have acquired all the important concepts from this project. You only need to make some modifications and then you are ready to go. Wish you all the best for the upcoming projects! 🙌

Tip: If you are interested in this topic, I would suggest that you could read [this post](#) to get more ideas about how to combine different alpha signals. Also [this notebook](#) to learn how to determine the risk exposure of alpha factors. Moreover, you can take a look at [this](#) curated list to learn more about how other people are using ML to do the financial prediction.

### Features and Labels

Describe the relationship between the shifted labels.

Great observation! 🙌 The auto-correlation is indeed highly correlated between two consecutive days.

Correctly implement the `train_valid_test_split` function.

Good job on splitting the data sets based on the index `all_x.index.levels[0]` 🏆

## Random Forests

Describe why `dispersion_20d` has the highest feature importance, when the first split is on the `Momentum_1YR` feature.

You're slightly off from the question. The idea is that the feature importance will be **accumulated** and every split we only consider using the maximum important one to do the split. In other words, the `Momentum_1YR` feature has the largest information gain for one split. The `dispersion_20d` feature has more information gain when dealing with more splits.

The reason why `dispersion_20d` has the highest feature importance **is** that the greatest normalized total reduction of the criterion it the one brought by `dispersion_20d`, which **is** used **in 3** nodes, **while** the others are used **in** less than **3** nodes.

Describe how the accuracy changes over time and what indicates the model is overfitting or underfitting.

Great answer 👍 The model is indeed overfitting because the accuracy of training is much better than the validation set

## Overlapping Samples

Correctly implement the `non_overlapping_samples` function.

Good job, you correctly implement the `non_overlapping_samples` function. 👍

Correctly implement the `bagging_classifier` function.

Fantastic, `bagging_classifier` function is correctly implemented. 🏆

Correctly implement the `calculate_oob_score` function.

Well done, you successfully complete this `calculate_oob_score` function with `oob_score_` attribute. 🏆  
Tip: Here is another way to it.

```
np.mean([clf.oob_score_ for clf in classifiers])
```

Correctly implement the `non_overlapping_estimators` function.

Good, `non_overlapping_estimators` function successfully passed the test by correctly using `non_overlapping_samples` to split the data for each classifier. 🎉

```
n_x, n_y = non_overlapping_samples(x, y, n_skip_samples, start_i=i)
```

👍 RESUBMIT

📄 DOWNLOAD PROJECT

Learn the [best practices for revising and resubmitting your project](#).

RETURN TO PATH

Rate this review

START