Note On Similarities

Dear Dr. Ayman,

In this document, I want to highlight the differences between my project and a similar one found online which you could check through this link: <https://towardsdatascience.com/predicting-future-stock-market-trends-with-python-machine-learning-2bf3f1633b3c>.

The linked project above is the one that inspired me to tackle the problem of predicting the stock market from a classification view point. It was the first spark for me and in it I saw the perfect setup to build something bigger and better (which I was successful in doing considering the superior accuracy scores).

**Classifiers:**

In the linked project, 2 different classification algorithms, namely kNN and Random Forest, are used and a single ensemble was built using them.

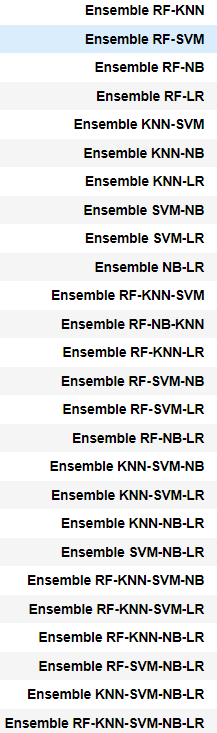
In my project, I added 3 other different classification algorithms and hence the total classification algorithms now would be 5 and they are:

1. kNN
2. Random Forest
3. Support Vector Machine
4. Naïve Bayes
5. Logistic Regression

I have not even used the same kNN and Random Forest methods he had setup as I have tuned the hyperparameters with different inputs. However, I only have so much room to change as the way you train a kNN or a Random Forest model is mostly similar in most implementations. The other 3 new classification algorithms are all my own implementations along with their hyperparameter tuning.

**Ensembles:**

In the linked project’s code only one Ensemble was built and that was the ensemble that used the classifications of the kNN and Random Forest classifiers. In my code I have implemented 26 ensemble’s and they are shown below in Figure 1.



Figure

All combinations of the 5 models, which total 26 combinations, were implemented. Not only that, but I added to the ensemble models themselves a very powerful feature. That feature is that the voting classifier’s method takes in as parameter the accuracy of the individual models and assigns them as weights to the vote of each classifier in the voting classifier. Please check Figure 3 for the code and clarification on how that was done.

Figure 2 shows the ensemble function built in the linked project and Figure 3 shows my own implementation. This highlights that even the only ensemble we have in common (The RF-KNN ensemble); I have not built it the same way and it takes more parameters and has an extra feature which is adding the accuracies of the individual models as weights to the voting classifier.

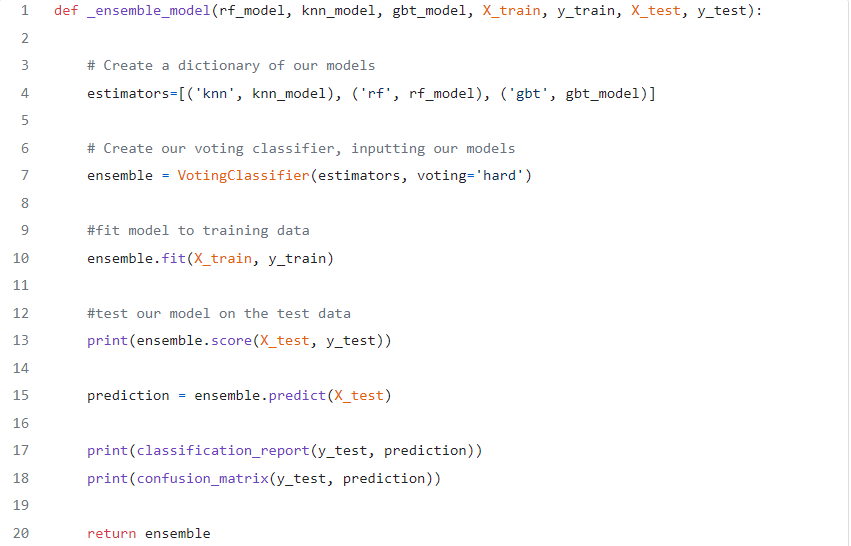


Figure : Ensemble Function in Linked Project

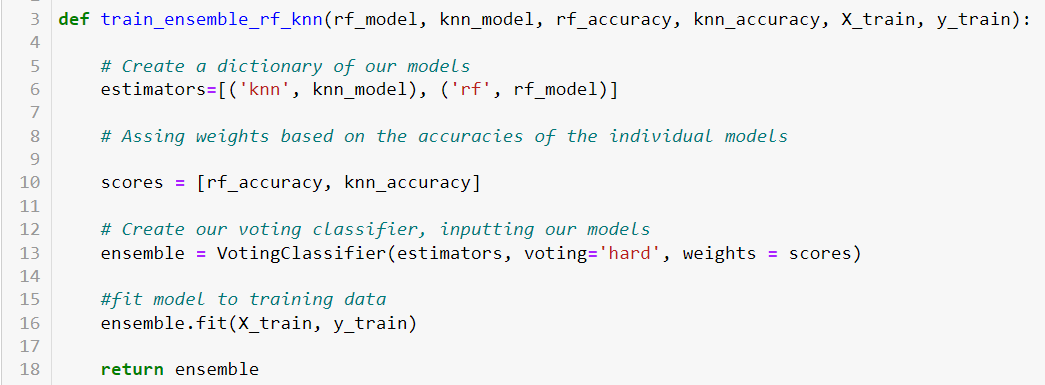


Figure : My Implementation of the Ensemble

**Cross-Validation:**

The cross-validation method implemented in the linked project would work well with my code and I understand its implementation very well. However, and following the lecture we had with you where you mentioned the importance of it being our own work even if it is not the best, I have implemented my own cross-validation method that builds up all the individual models and their ensembles. This cross-validation method is unique to me and is my own work. My cross-validation method is over 380 lines long and hence I will not include it here but you can certainly check it out through the link to my GitHub repository which includes the whole code.

**Results:**

His cross-validation method does not save and return results and only prints them and hence makes it hard to investigate the data further. It also prints the intermediate results of each fold in the 10-fold cross validation, as shown in Figure 4, in a haphazard manner that is hard to make sense of, and again it also printed and not returned from the function and hence it is hard to even investigate the intermediate results of every fold in the cross validation.

On the other hand, my cross-validation method returns a zipped object that zips together the name of every model along with a list that has the model’s accuracy in each fold of the 10-fold cross validation. This can be seen in the last few lines of my cross-validation method as shown in Figure 5.

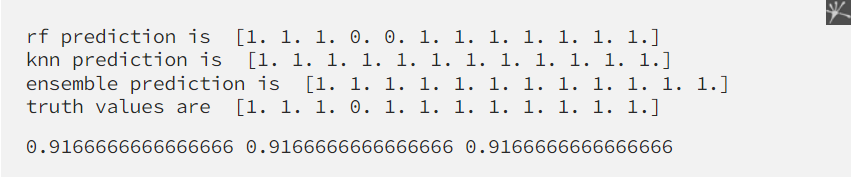


Figure : Intermediate results of linked project

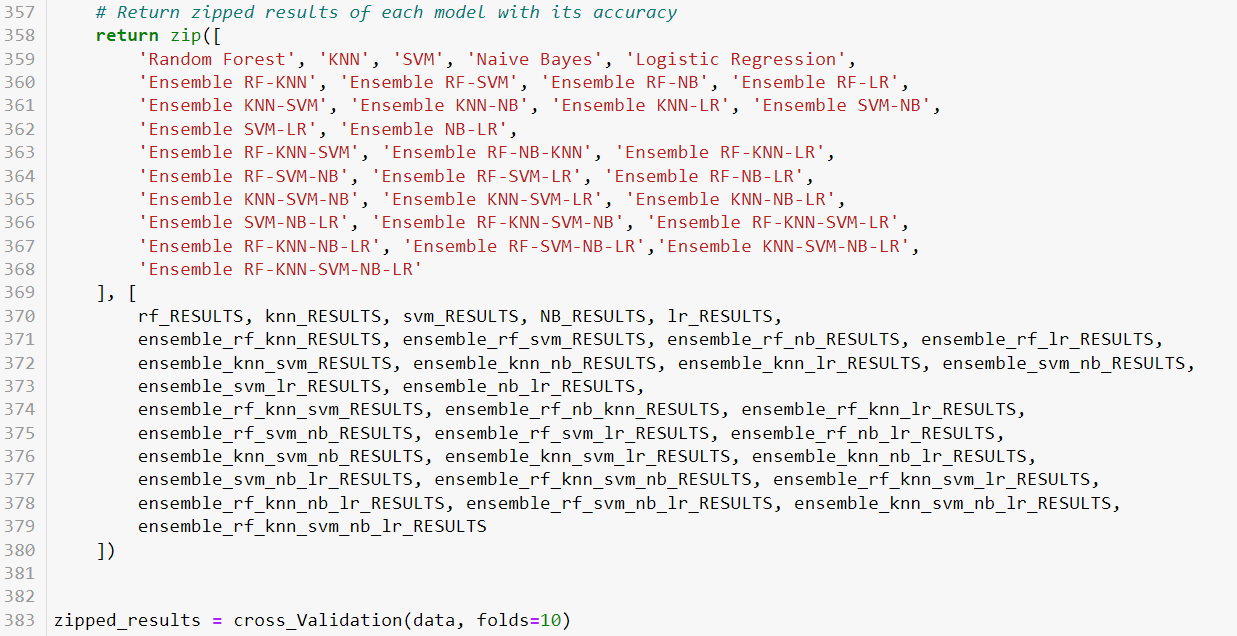


Figure 5: Return statement of my cross-validation method

For my final output for results, I wrote a function that takes this zipped object that was outputted from the cross-validation as a parameter and returns a Data Frame. The function I wrote is shown in Figure 6. When I pass a zipped object that was returned from a 10-fold cross validation we get a Data Frame like the one in Figure 7.

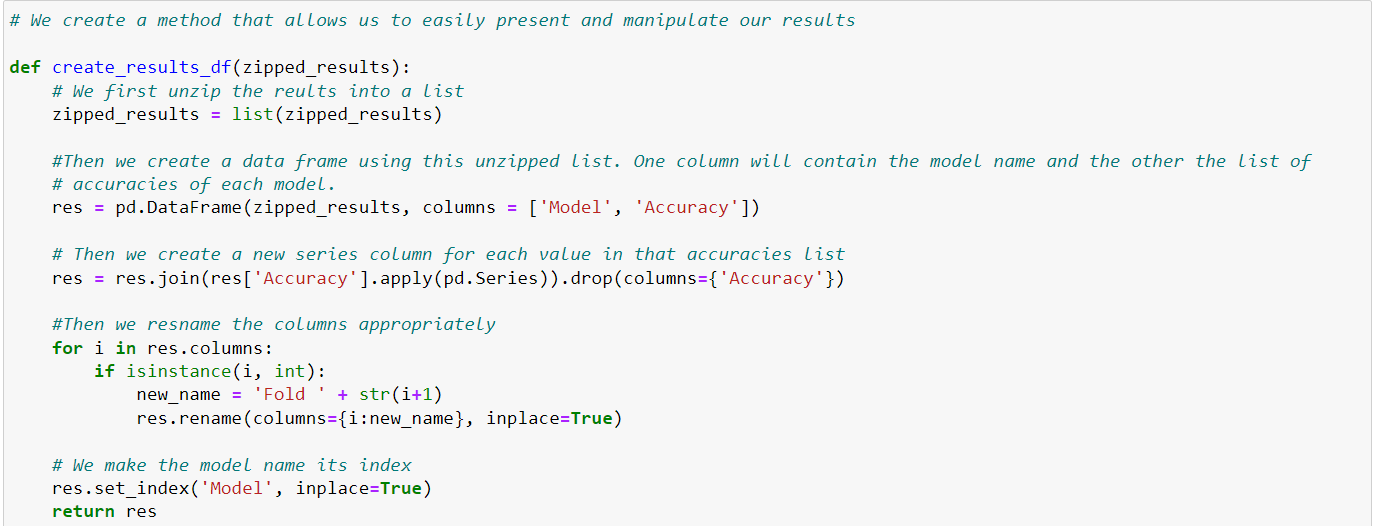


Figure 6: Function to return zipped results as a Data Frame

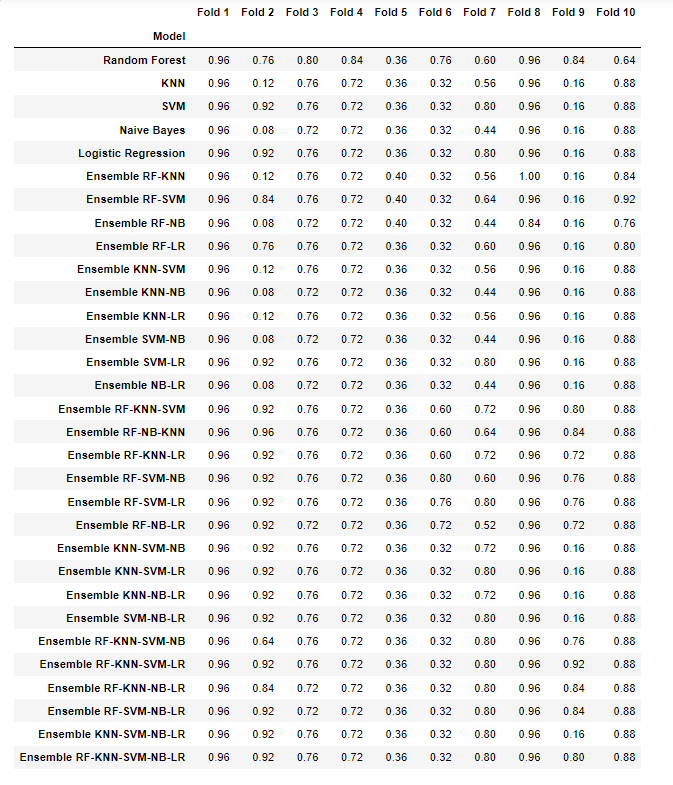


Figure 7: Results of all models in every fold of the cross-validation

These Data Frames give you the option to further investigate the data by, for example, examining several descriptive statistics for the whole Data Frame. It also allows you to view the performance of every fold on average across all models as show in Figure 8 and not only look at the accuracies of the models.

The point here is that with the output presented like that in Figure 7, I was able to uncover more details in the data, such as Fold 5 being the worst-performing fold on average, and made me investigate why that is and come up with interesting conclusions that really aided my results (for reasons why Fold 5 is the worst performing fold please refer to the Thesis Paper).

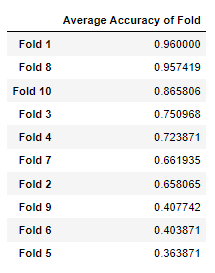


Figure 8: Average Accuracy of every fold

**Similarities:**

After discussing the differences this last section will discuss similarities.

The only functions that have been copied as is from this code are the following:

1. get\_indicator\_data (which has a similar name in my code)
2. \_produce\_prediction (which is called get\_truth\_values in my code)

Other than these two methods, any resemblance between the two codes is out of mere necessity since we are tackling the same issue from the same angle and hence will have some resemblance. Nevertheless, I have written over 30 functions in my code, and other than the 2 I just mentioned, nothing has been copied or taken as is.

I will be happy to answer any further questions you have during our appointment or later on in my mini-defense.

Thank you for your support and understanding throughout this thesis and thank you for your time.