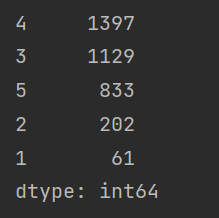
**USING SVM TO PREDICT VIEWER’S FEELING TOWARDS A YOUTUBER’S STYLE THROUGH FOOD DESCRIPTION**

**Answer to required questions (detailed report below):**

* Kernel used: SVM linear kernel, since the number of features is significantly smaller than the number of examples (15 to 3622 ratio).
* Result:
  + SVM with random hyperameter initialization achieves accuracy only at 39.4%.
  + Grid search is used to choose the best C and max iterations in SVM.
  + The most stable set of hyperparameters and supposedly the best ones are with C = 0.001 and max iterations = 1000.
  + The final accuracy of SVM kernel sits at around 40%. (lower than previous Neural Network models, which was at 44%)
* Conclusion:
  + C = 0.001 produce the best result probably since how noisy our data is and to consider how most of our data is skewedly labeled.



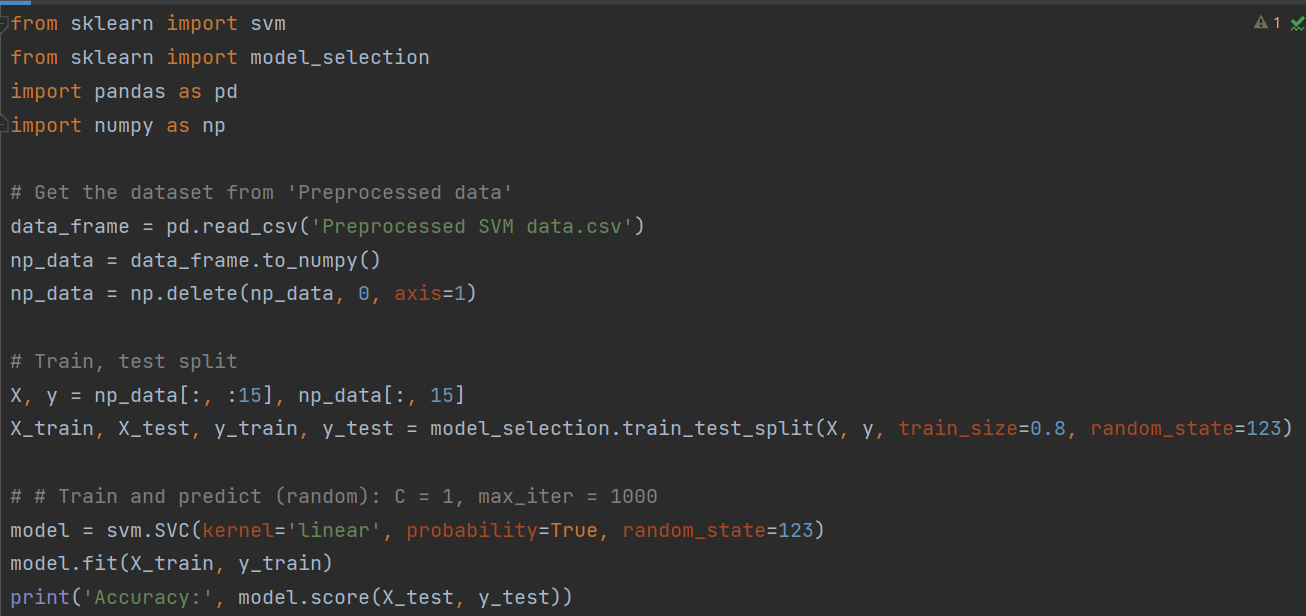
* + SVM is normally used in datasets where the number of features is larger than the number of examples in our data. Low performance is to be expected if we don’t increase our number of features in our data.
  + Using data with additional squared features yields better results in terms of accuracy, but not in ROC curve score.

Link to the codes: [Google Colab](https://colab.research.google.com/drive/1BEIKwMLnlFywDd7W9xJtutDKm6lREWzJ?usp=sharing)

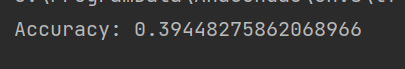
**Detailed report**

**SVM Model with random hyperparameters:**

* Kernel used: linear, since the number of features is significantly smaller than the number of examples (15 to 3622 ratio).
* Implementation:
  + The data used is the original data with 15 features and 1 output which is the rating, scoring from 1 to 5.
  + Random\_state is set to 123.



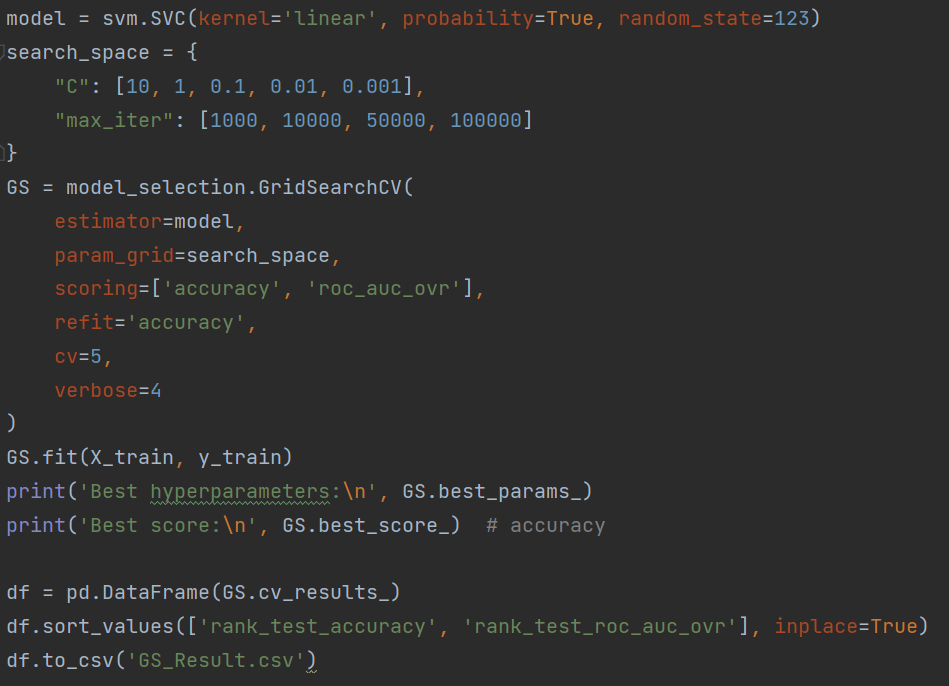
* The result:



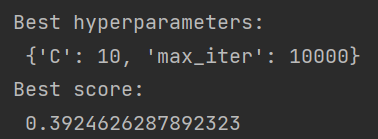
* + The performance is a bit lower than the previous Neural Networks. (43%).
  + This happens since this is just a random initialization of hyperparameter C = 1 and max iteration is set at 1000.

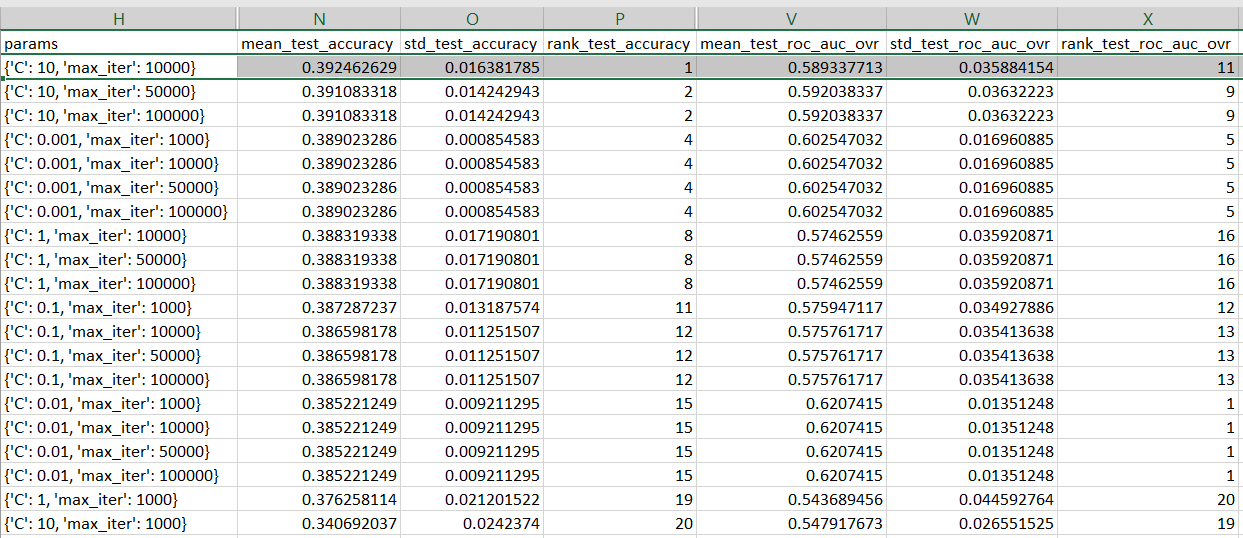
**Comparing SVM Models with grid search CV:**

* Grid search CV: is basically a way to evaluate multiple models based on different hyperparameters. Let’s say we have a list of 3 C values and 5 max iterations to consider. Grid search CV will train all these 15 models and generate results. Then we can use pandas to get all these results onto a csv file and decide which set of hyperparamters do best.
* Implementation:

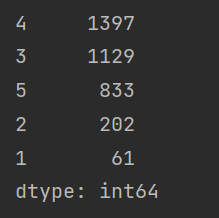


* + In this implementation, we will consider 5 C values and 4 max iteration settings. A total of 20 models will be trained and compared.
  + Our scoring method is going to be accuracy first, then ROC Curve for multiclass classification.
* The result:



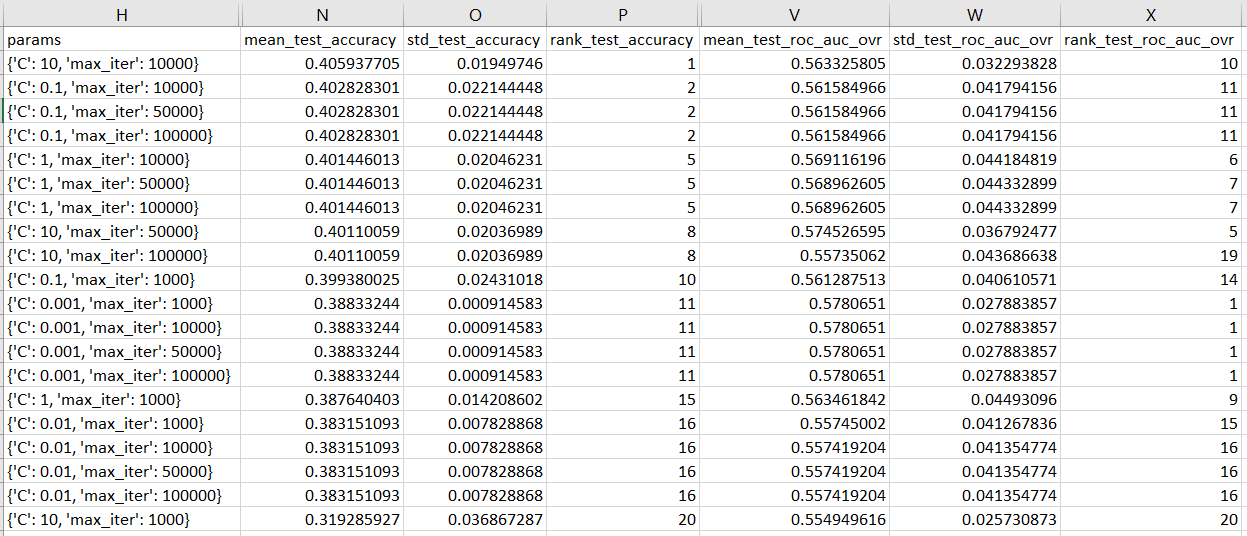


* + In the results, we can see how C = 10 and max iterations = 10000 produce the best result in terms of accuracy.
  + However, if we look at results of C = 0.001, the accuracy is not too far down. Yet, the standard deviation in both accuracy and ROC curve is fairly low. That means the models themselves are very stable, comparing to the top accuracy score.
  + Trading a little bit of accuracy with stability can be a good decision when it comes to choosing our models.
  + Why does C = 0.001 produce such good result? The reason behind this is probably how noisy our data is. Considering how most of our data is skewedly labeled.



**Testing out the squared data:**

* Previously, we made a dataset with all the features squared. Now, let’s use that dataset and see if we can get a better model.
* The result:



* + Averagely, all the accuracy scores from all models improved. (1% improvement)
  + C = 10 and max iterations = 10000 are still the best hyperparameters accuracy wise.
  + C = 0.001 still holds the record in stability (low standard deviation).