**Customized Backward Elimination for Classification**

Backward elimination is used essentially with regression and ultimately finds the subset which gives the best “accuracy” with the regressive model.

Customization:

The main idea of backward elimination is kept constant, that is all features are taken up first, and then one by one features are eliminated depending on the criteria.

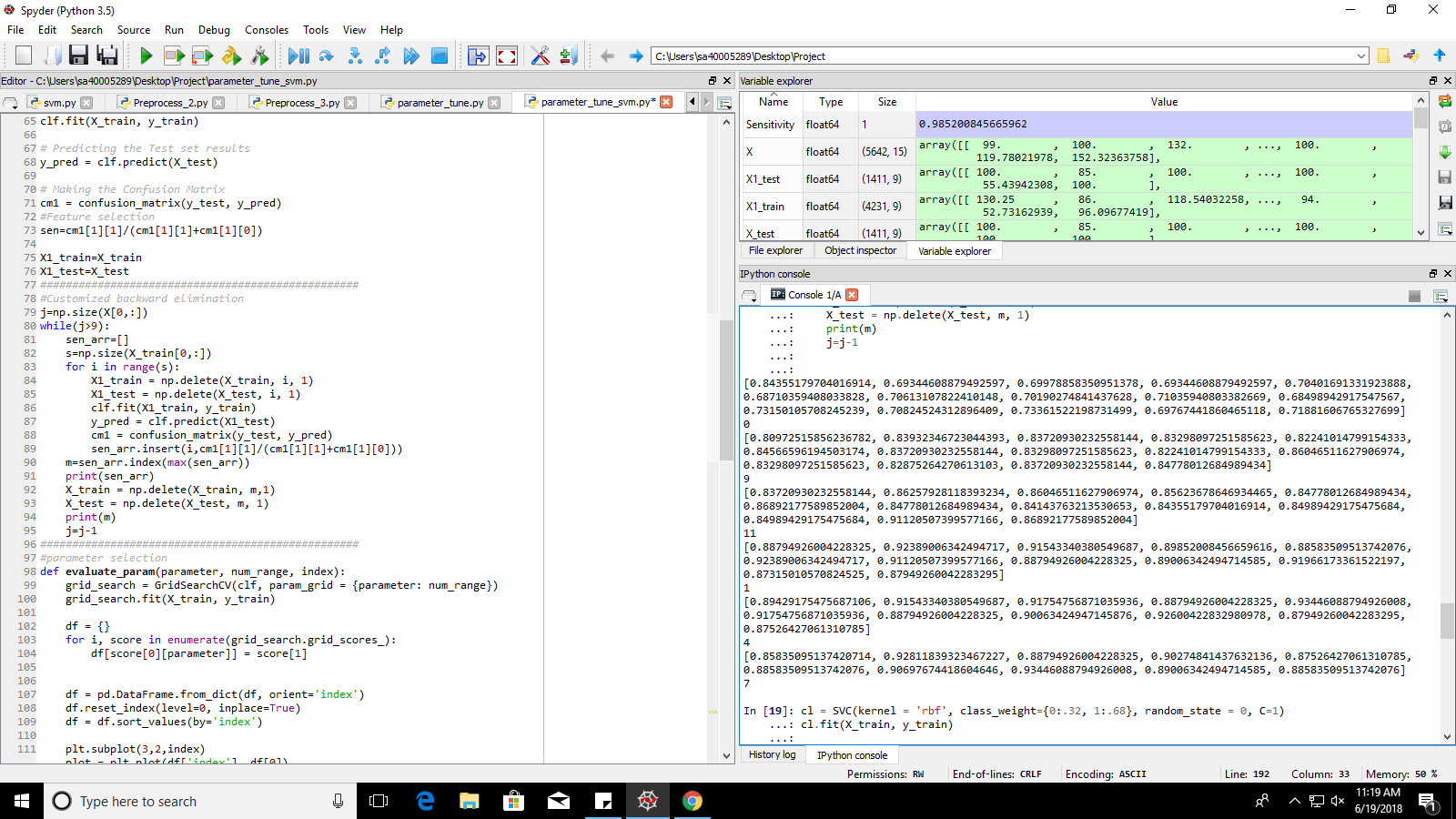
In the following code, the following steps have been implemented:

**Initial sensitivity: 71%**

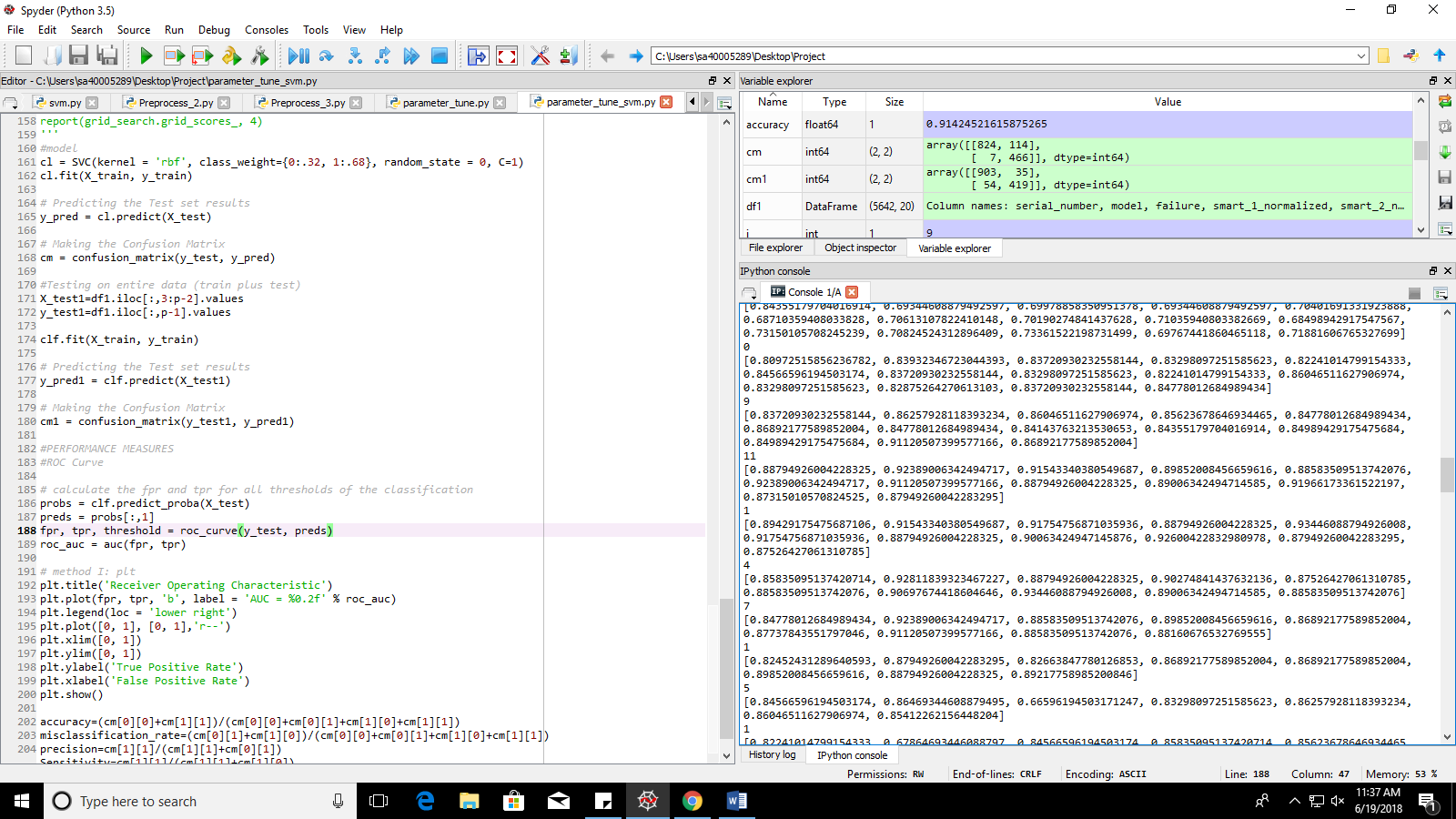
1. Model fitted with all features (15 features in this case)
2. While loop created (Selects top 5 features)
3. Since our final model must have a very high sensitivity (least misclassification of actual failure alarm as not a failure alarm), we select “sensitivity” as the criterion for feature elimination instead of “accuracy”
4. Inside while loop, another for loop is created which eliminates one feature at a time and checks sensitivity if that particular feature is eliminated.
5. An array (sen\_arr) is created which stores the sensitivity if ith column is deleted in the ith index.
6. We look for the index of the maximum value from this array since we need to eliminate a column only if it improves sensitivity and gives a better sensitivity than when another column is deleted instead.
7. We continue elimination until 5 features are reached.
8. Then we check all the arrays which were printed in the process and spot the maximum value among all. (0.93446088794926008 in this case, which is obtained after eliminating the 7th column). Thus, if we delete more columns after this value is reached, we will only be decreasing our sensitivity.
9. There improvement is made in the while loop and features are increased from 5 features to 9 best features.
10. Code is re-run.

**After fitting the SVC model on the final features, we get a sensitivity of 98.5%**

**Accuracy of 91.42% and precision of 80.34%**



Set of arrays (one for every iteration):



Final Confusion Matrix:

