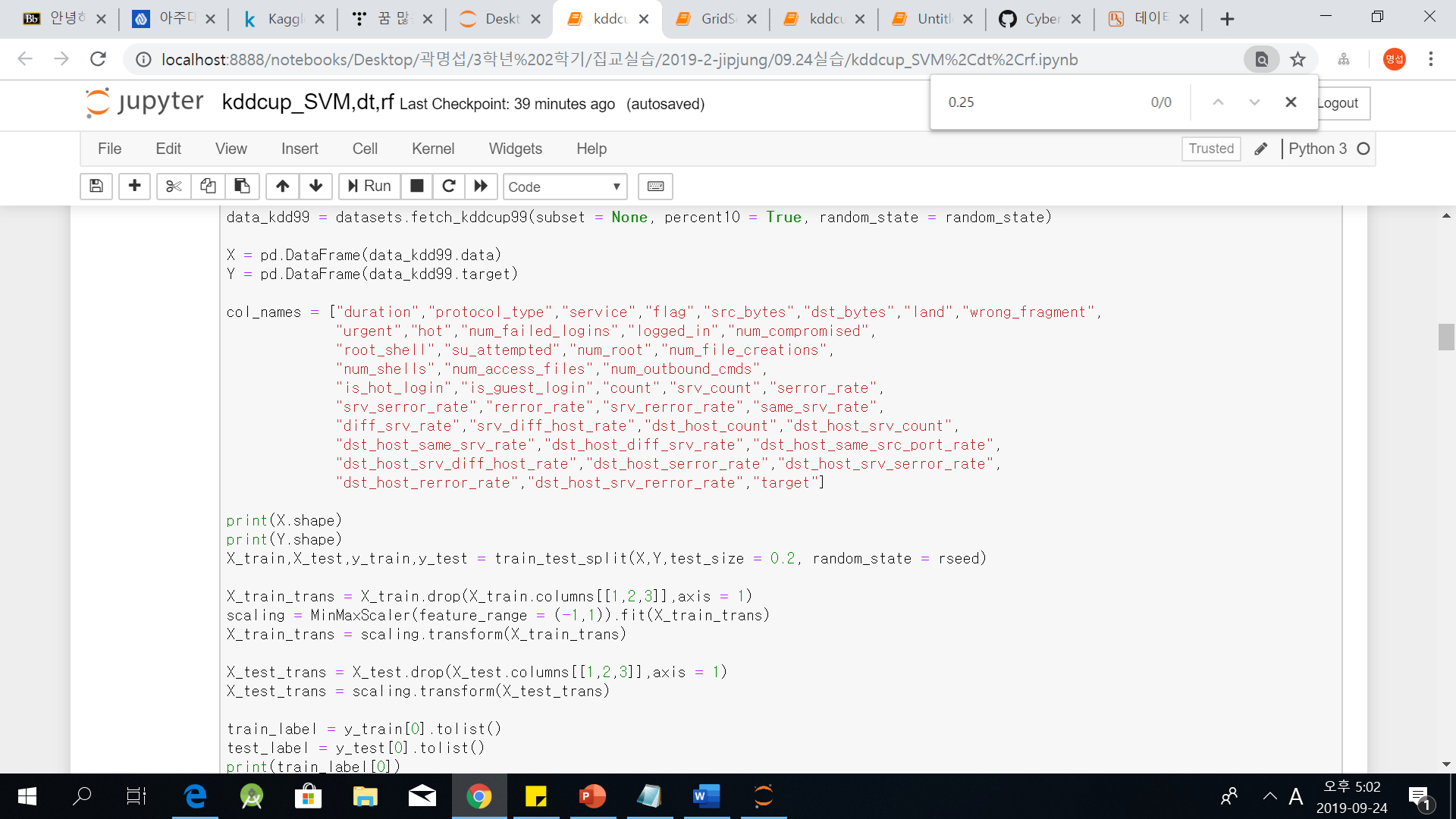
KDDCUP 실습 보고서

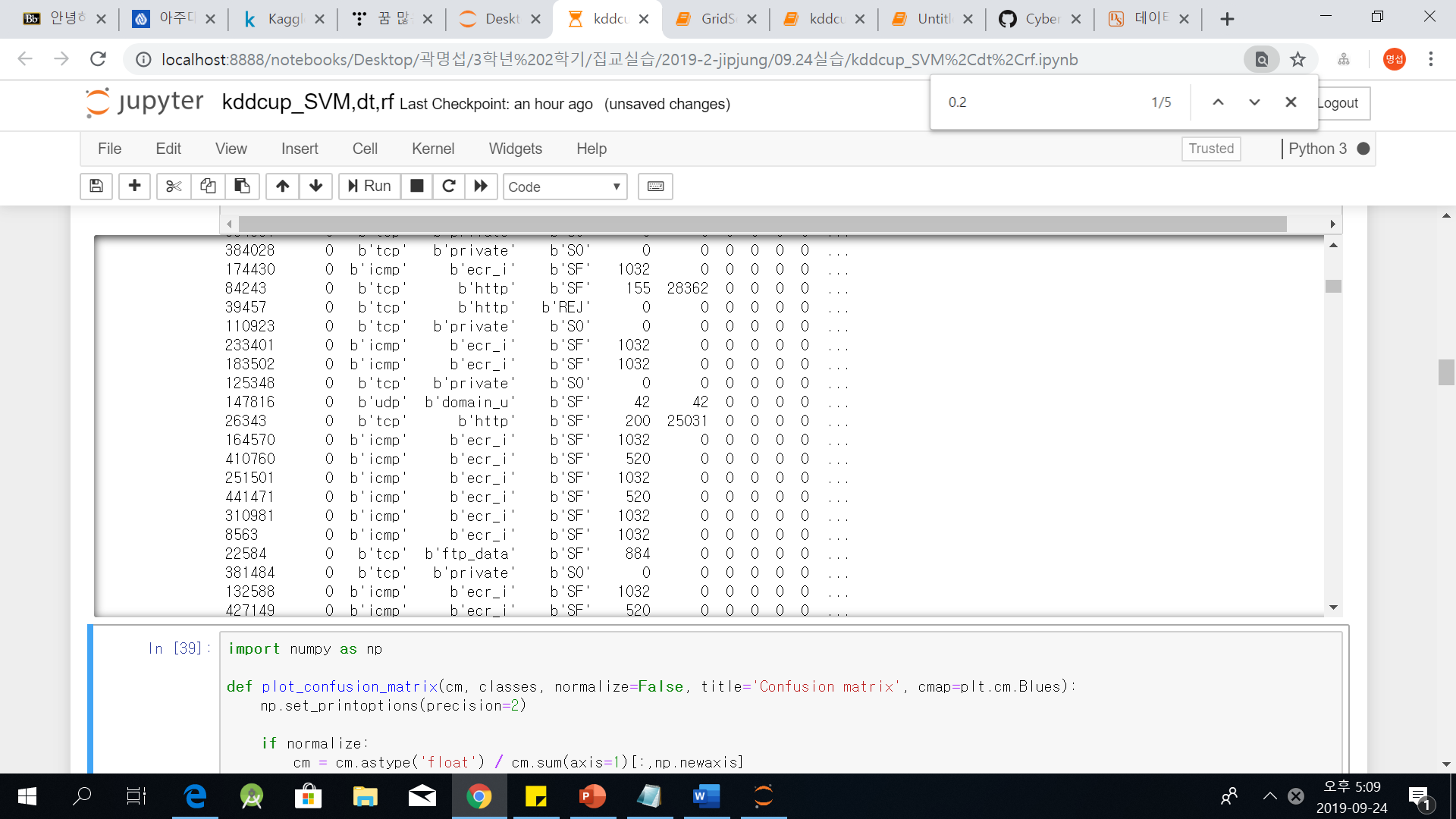
201520991

곽명섭

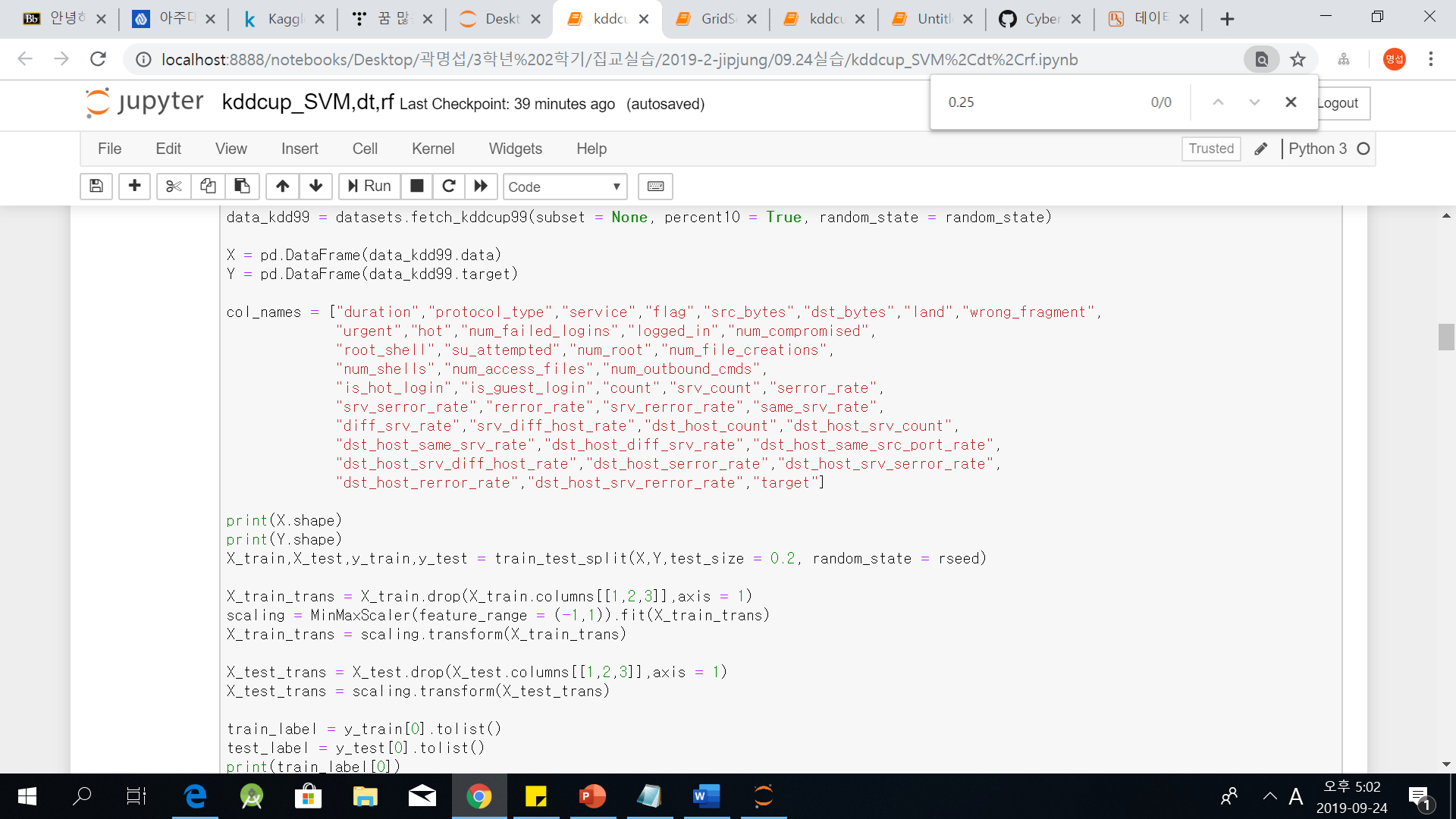
오늘 실습시간에는 KDDCUP의 데이터를 활용하여 classification을 하는 실습을 하였다. KDD 99 dataset을 Train set과 Test set으로 split하고 test\_size =0.2로 맞춰준 이후로 3가지 svm, decision tree, random forest를 이용한 classification을 한 후 Train과 Test accuracy를 도출해보는 실습이었다.



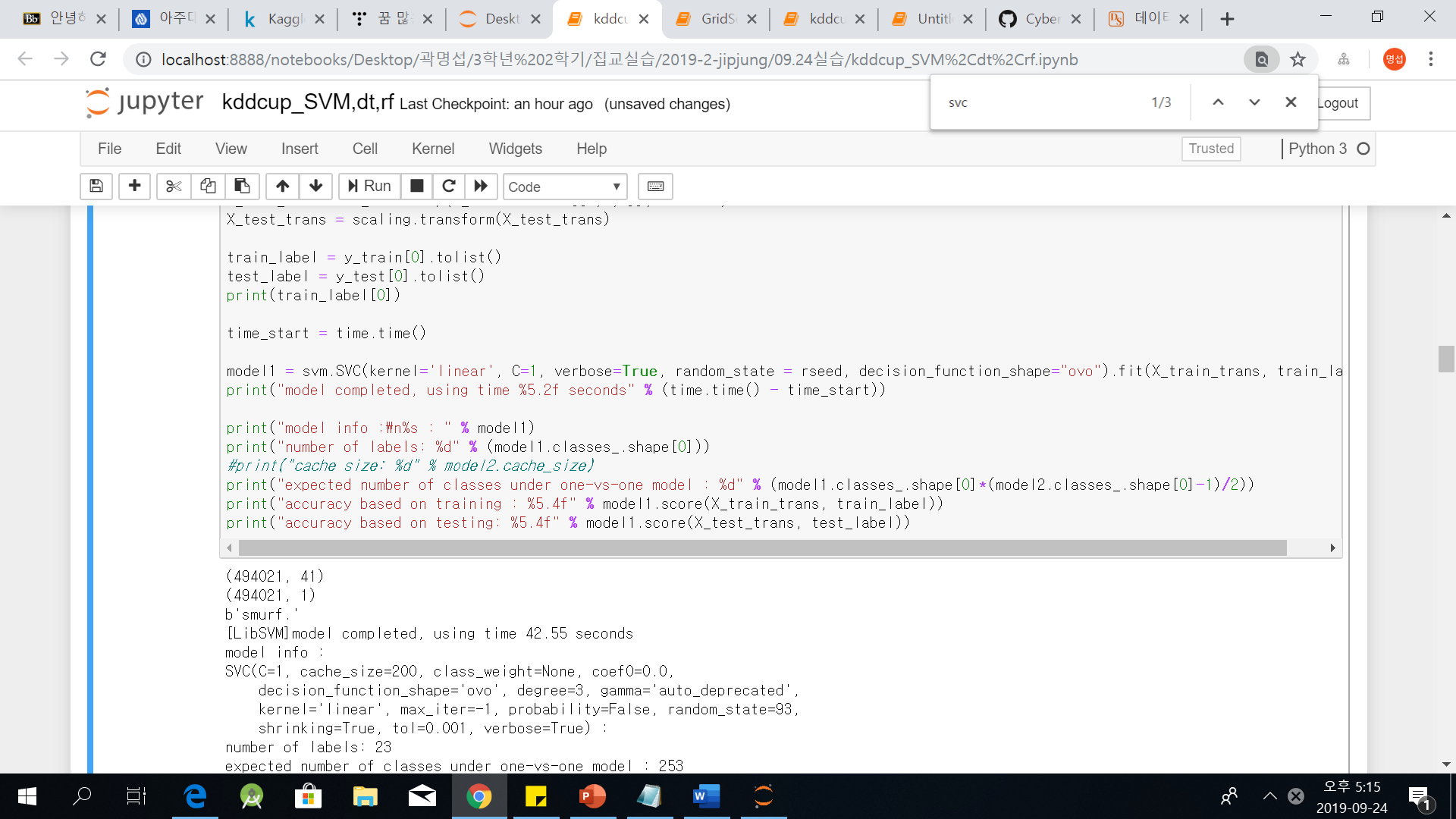
Kdd99 데이터를 불러오는 과정인데 처음에 막상 타이핑만을 하고 나니 이 데이터가 무엇인지 조차도 이해가 되지 않아 일단은 데이터가 어떤 식으로 되어 있는 건지를 알아보기 위해 print를 통해서 한번 데이터를 출력을 해보았다.



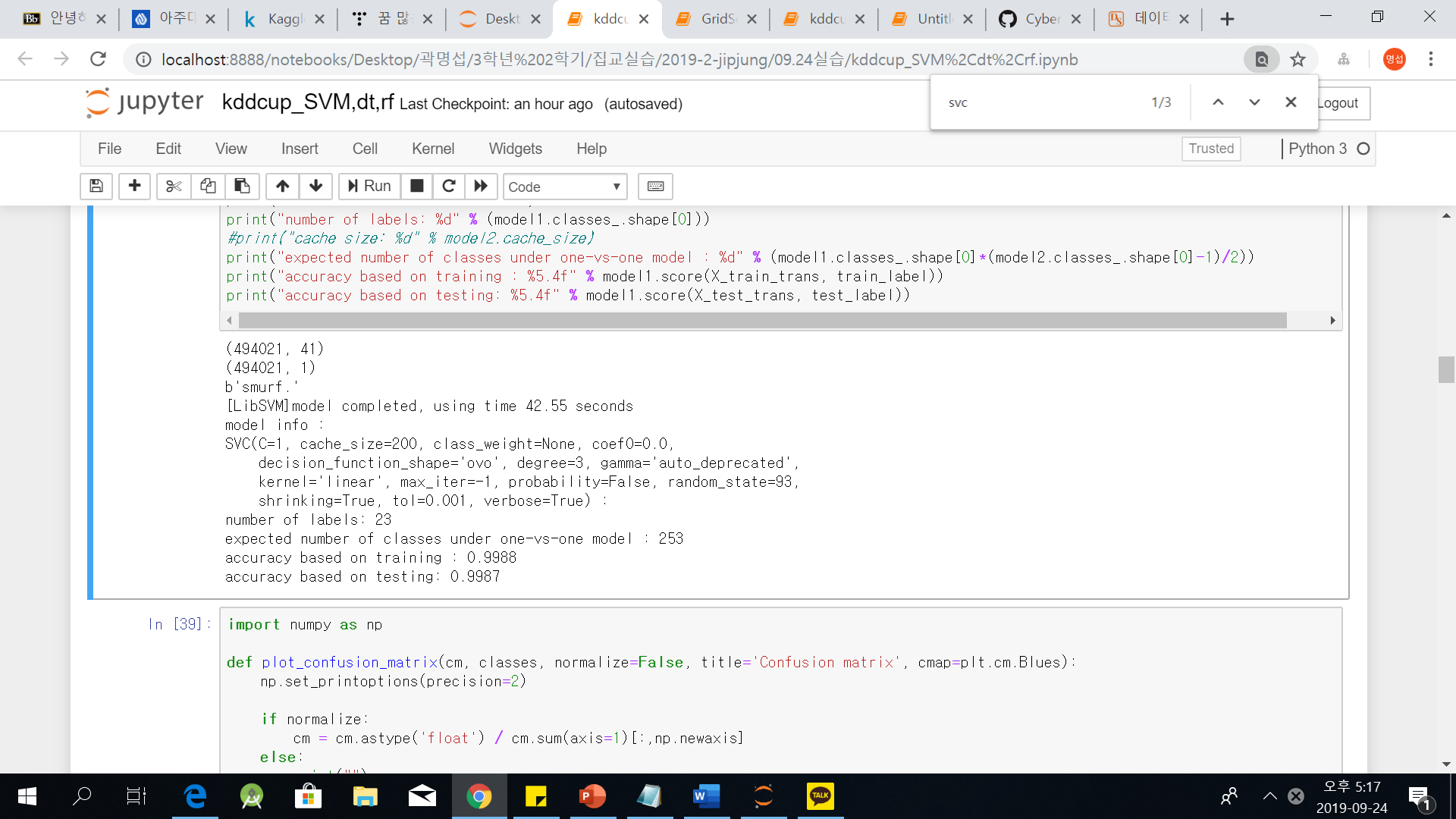
결과는 대략 알 수 없는 형식의 데이터의 포맷으로 이런식으로 출력이 되었다.



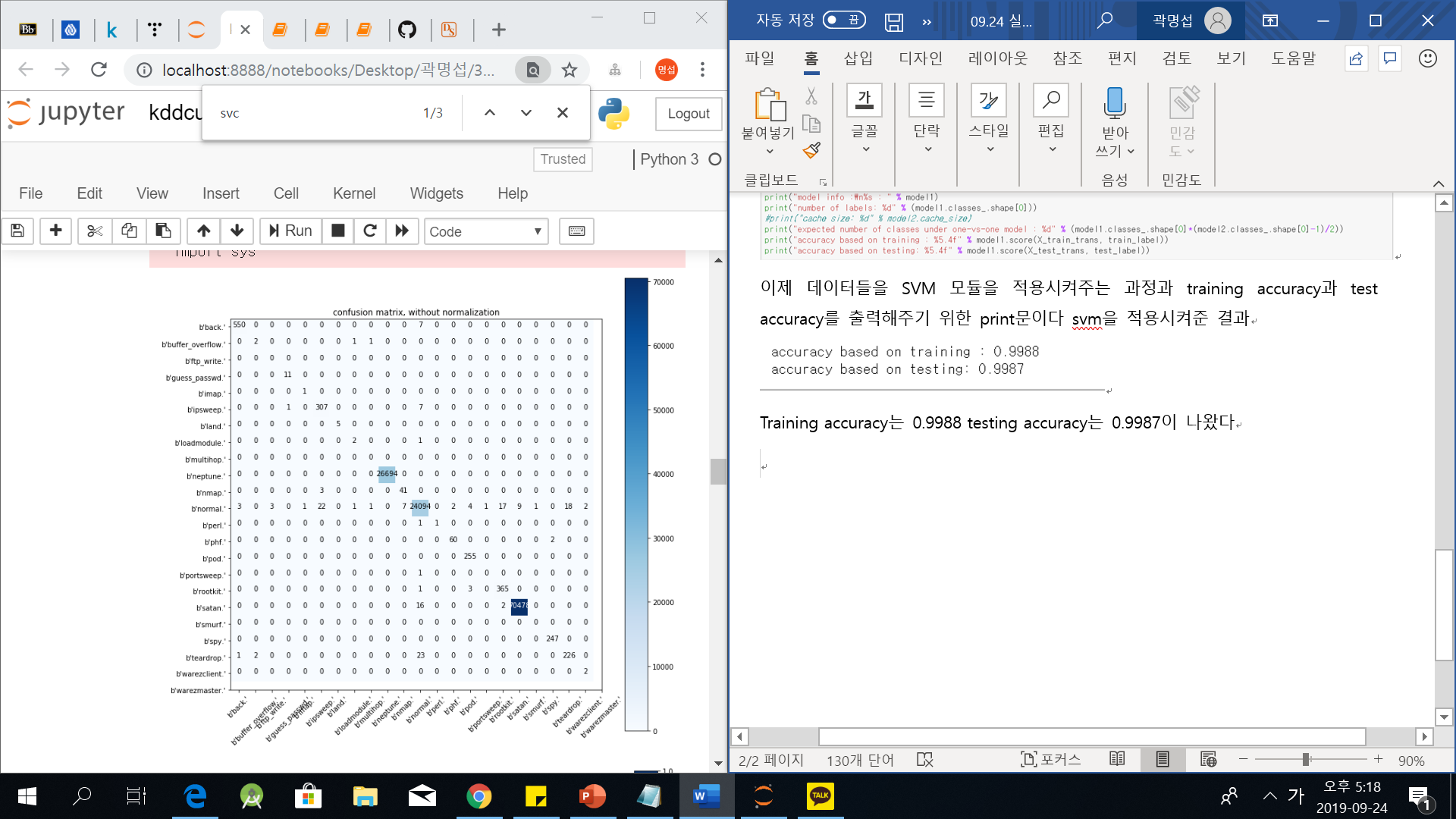
여기서 보면 그래서 exercise에 써있는데로 test split을 0.2로 두고 split을 한후에 1,2,3열에 있는 데이터들은 전부 drop을 해버리고 나머지 남은 숫자 데이터들을 MinMaxScaler로 변형시켜주는 과정이다



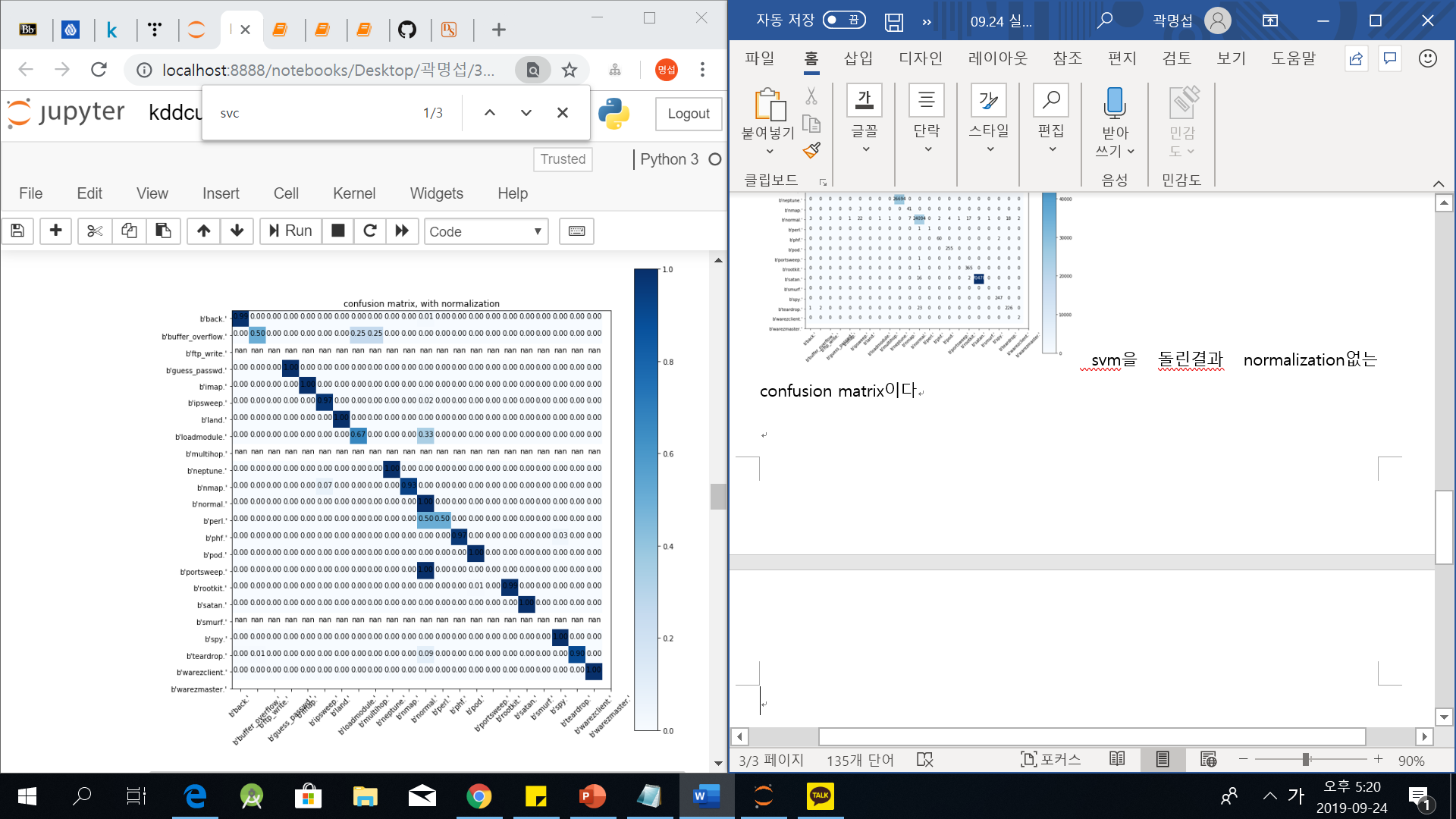
이제 데이터들을 SVM classifier 모듈을 적용시켜주는 과정과 training accuracy과 test accuracy를 출력해주기 위한 print문이다 svm을 적용시켜준 결과



Training accuracy는 0.9988 testing accuracy는 0.9987이 나왔다

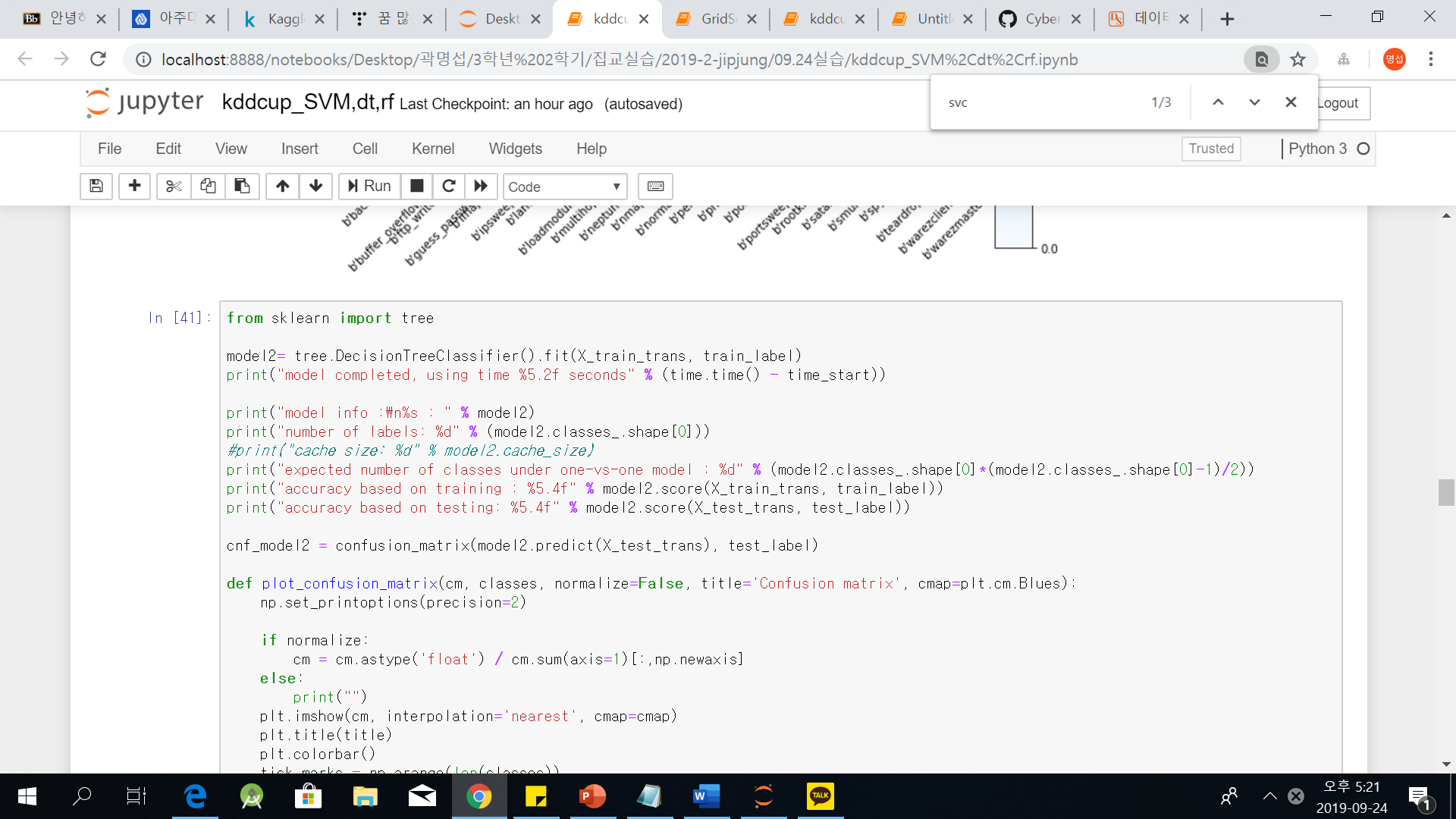


svm을 돌린결과 normalization없는 confusion matrix이다

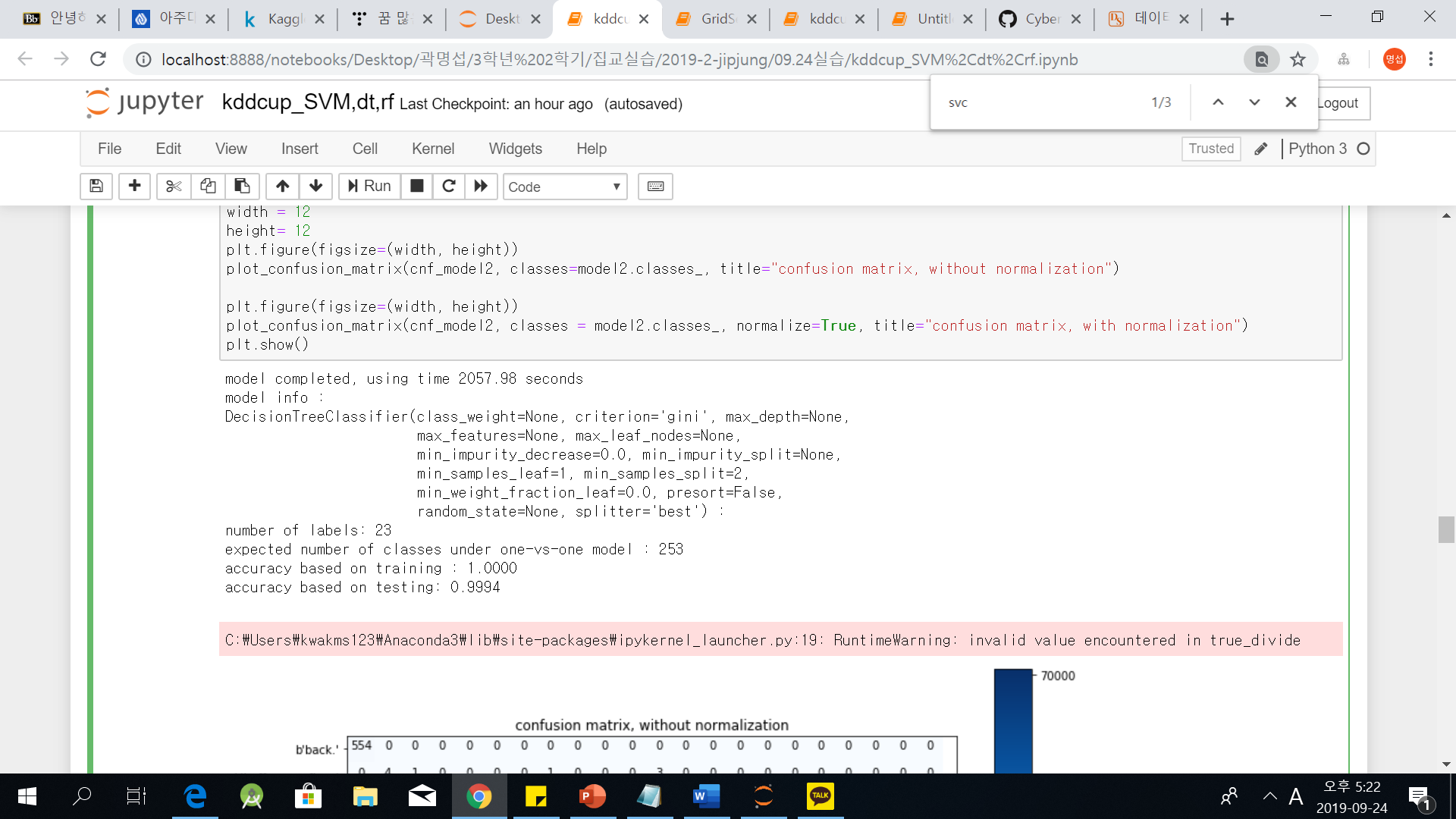


svm을 사용한결과 normalization을 한 confusion matrix이다.

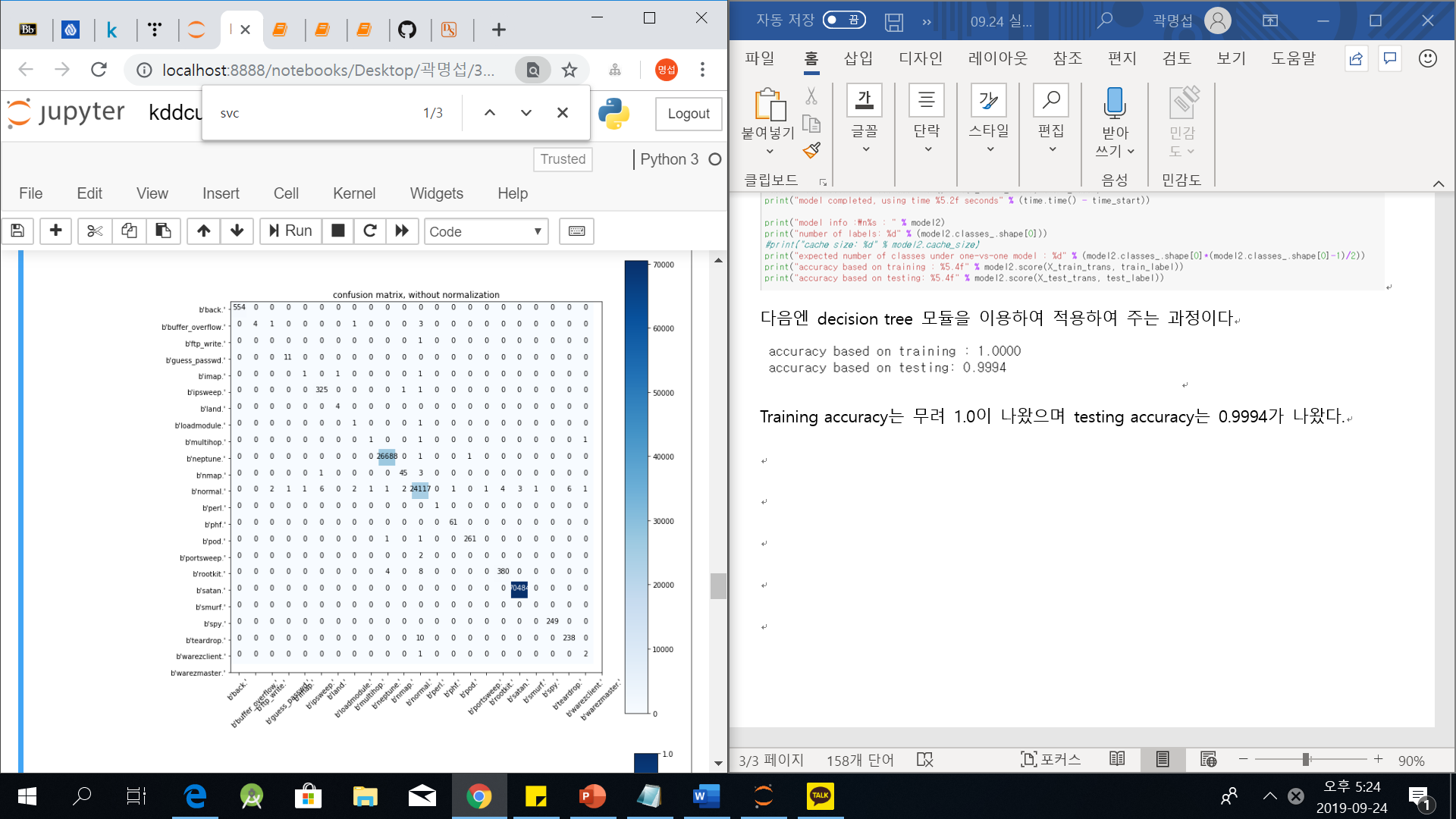
x축과 y축은 예측결과와 실제 결과인데 대각선에 치중 되어 있는 것으로 보아 잘 예측 했다는것을 알 수 있다



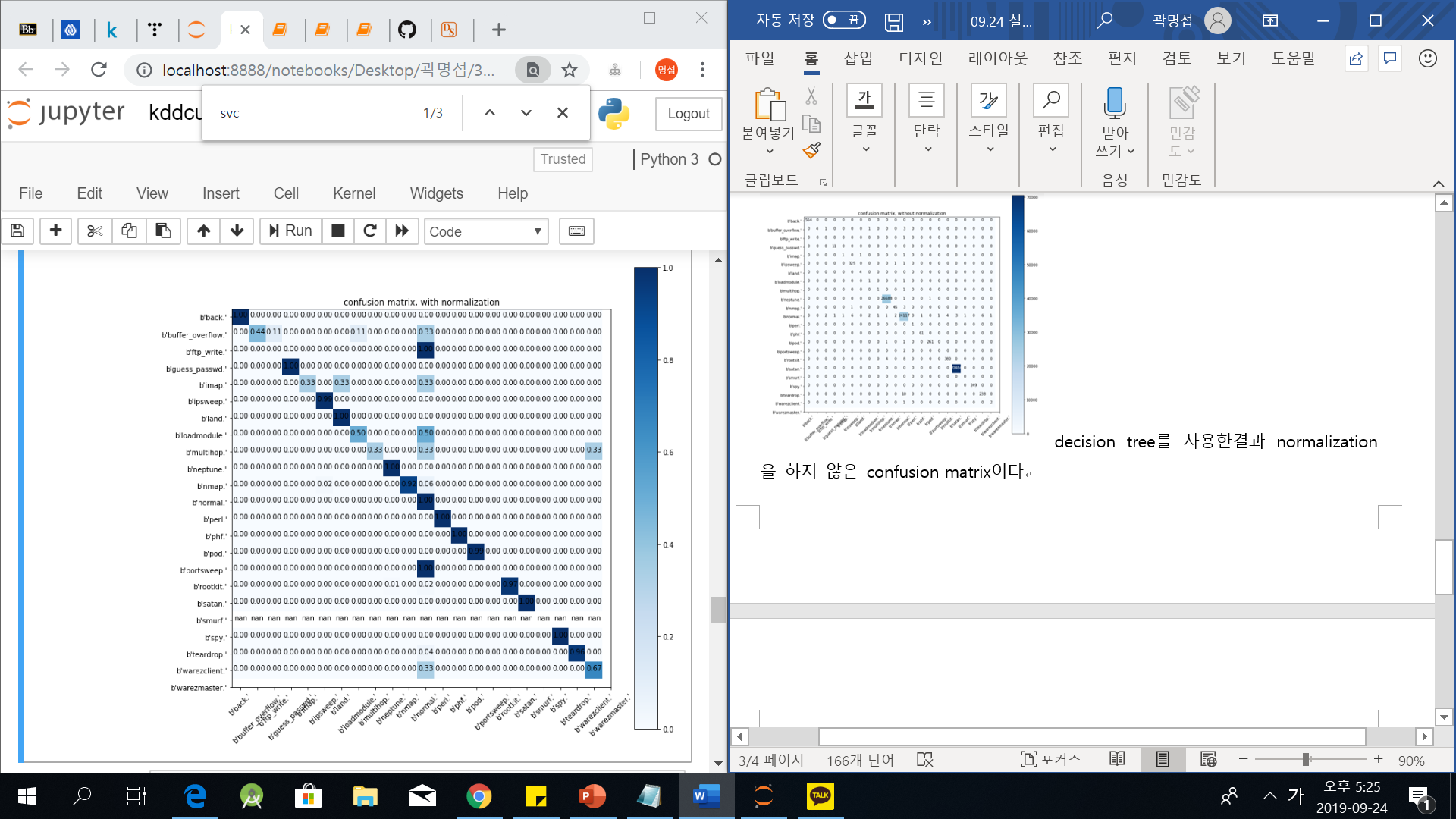
이제 데이터들을 DecisionTreeClassifier 모듈을 적용시켜주는 과정과 training accuracy과 test accuracy를 출력해주기 위한 print문이다 svm을 적용시켜준 결과



Training accuracy는 무려 1.0이 나왔으며 testing accuracy는 0.9994가 나왔다.

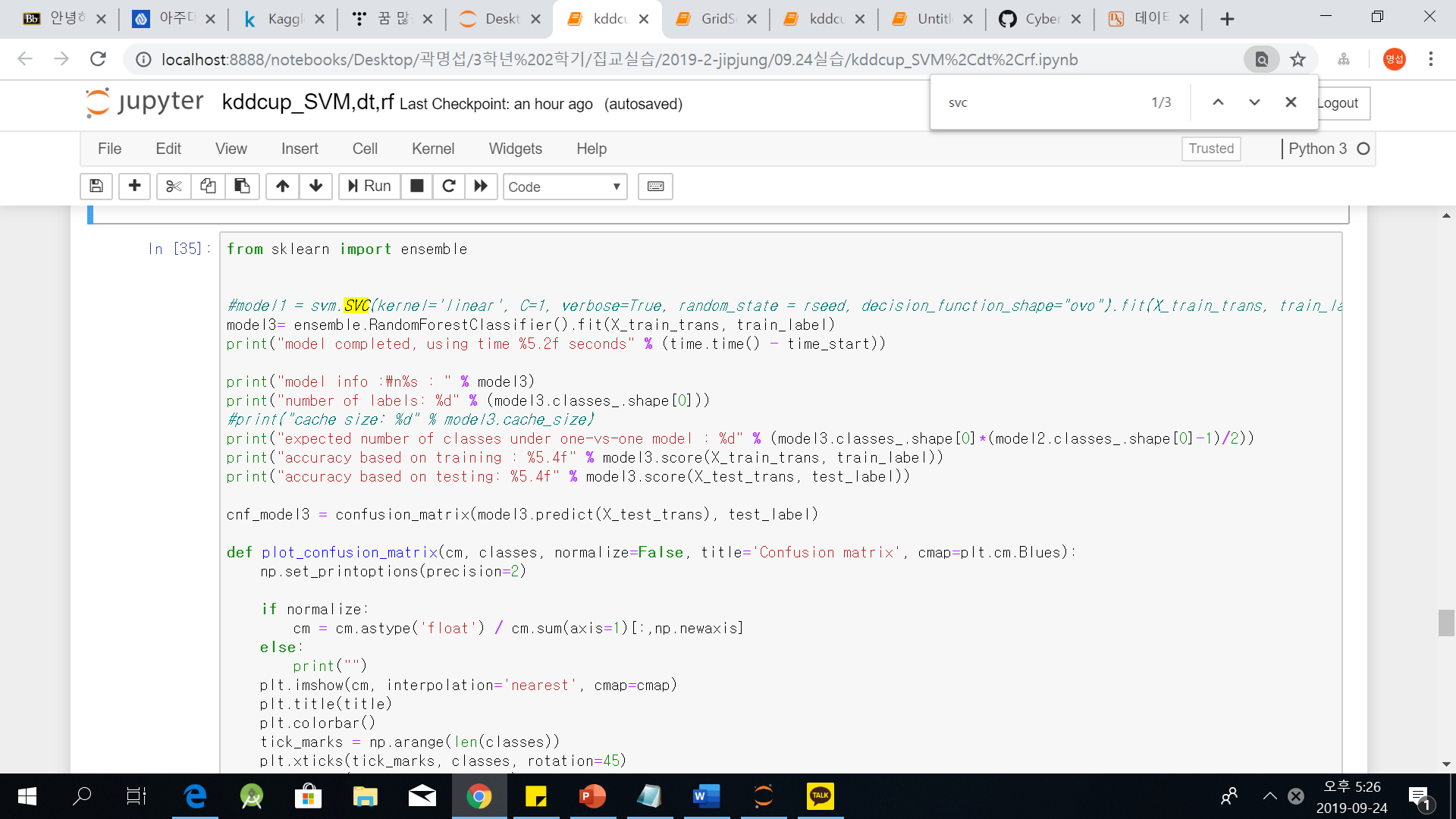


decision tree를 사용한결과 normalization을 하지 않은 confusion matrix이다

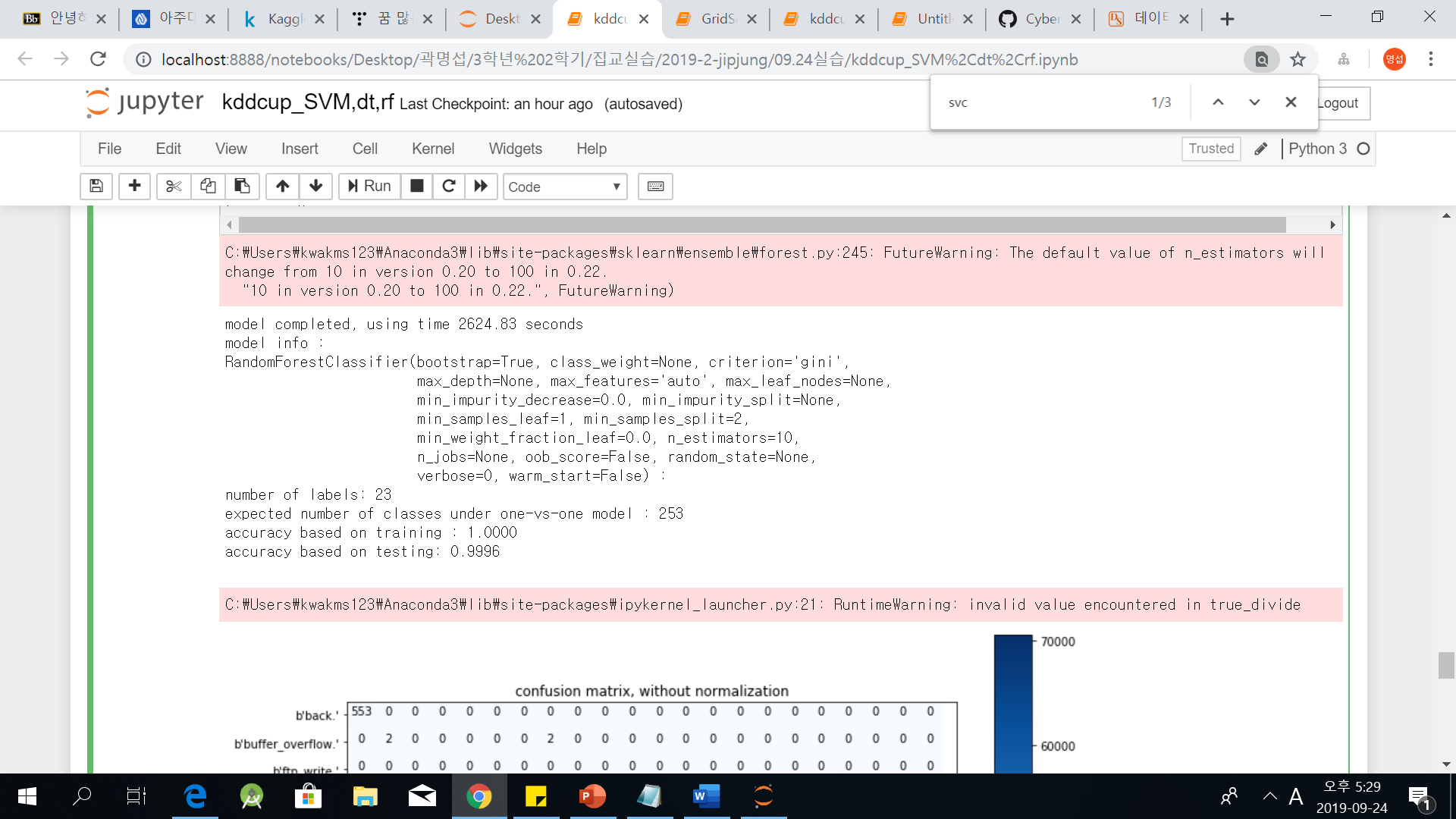


decision tree를 사용한결과 normalization을 한 confusion matrix이다.

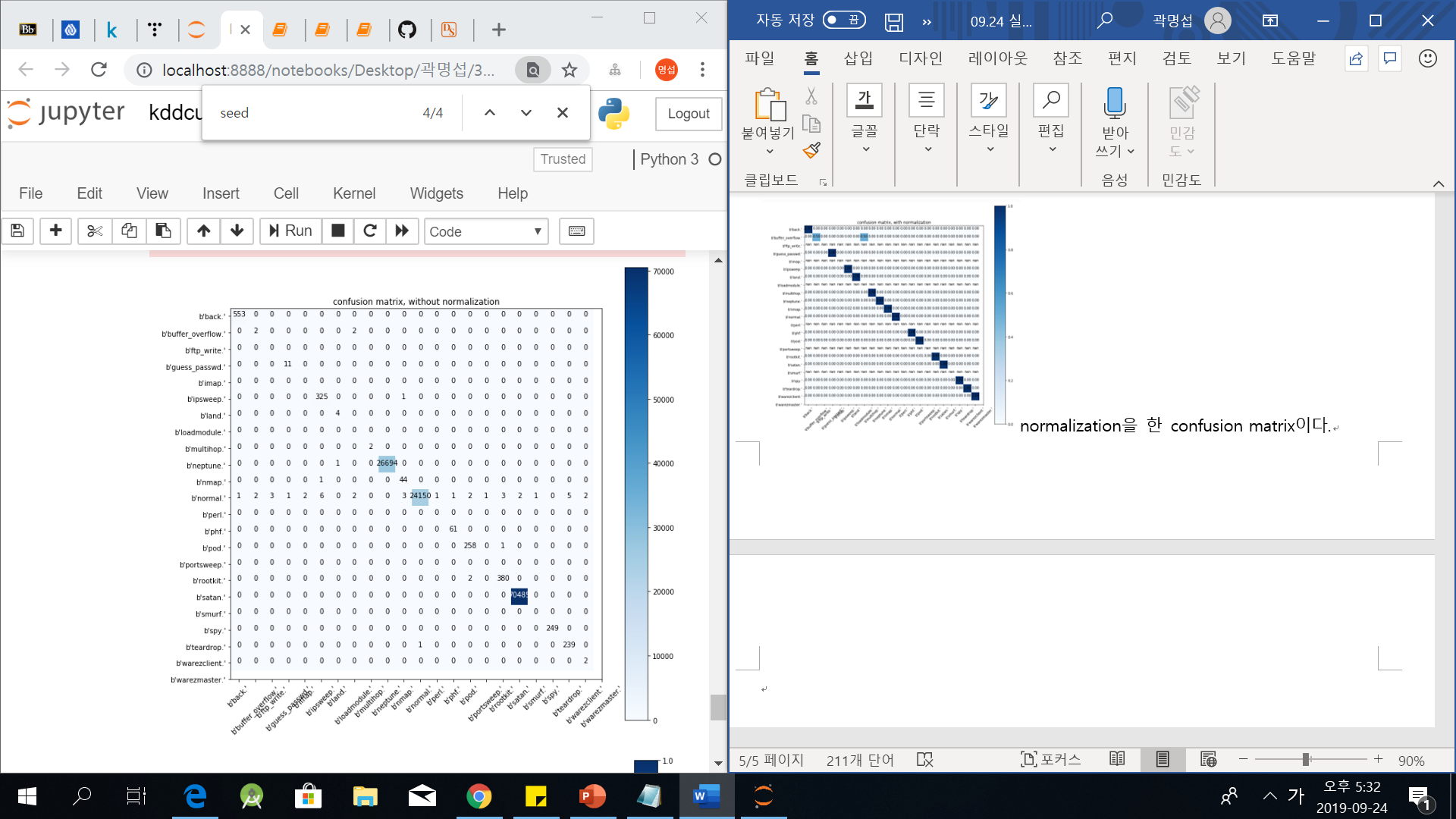
x축과 y축은 예측결과와 실제 결과인데 대각선에 치중되어 있는 것으로 보아 잘 예측했다는것을 알 수 있다

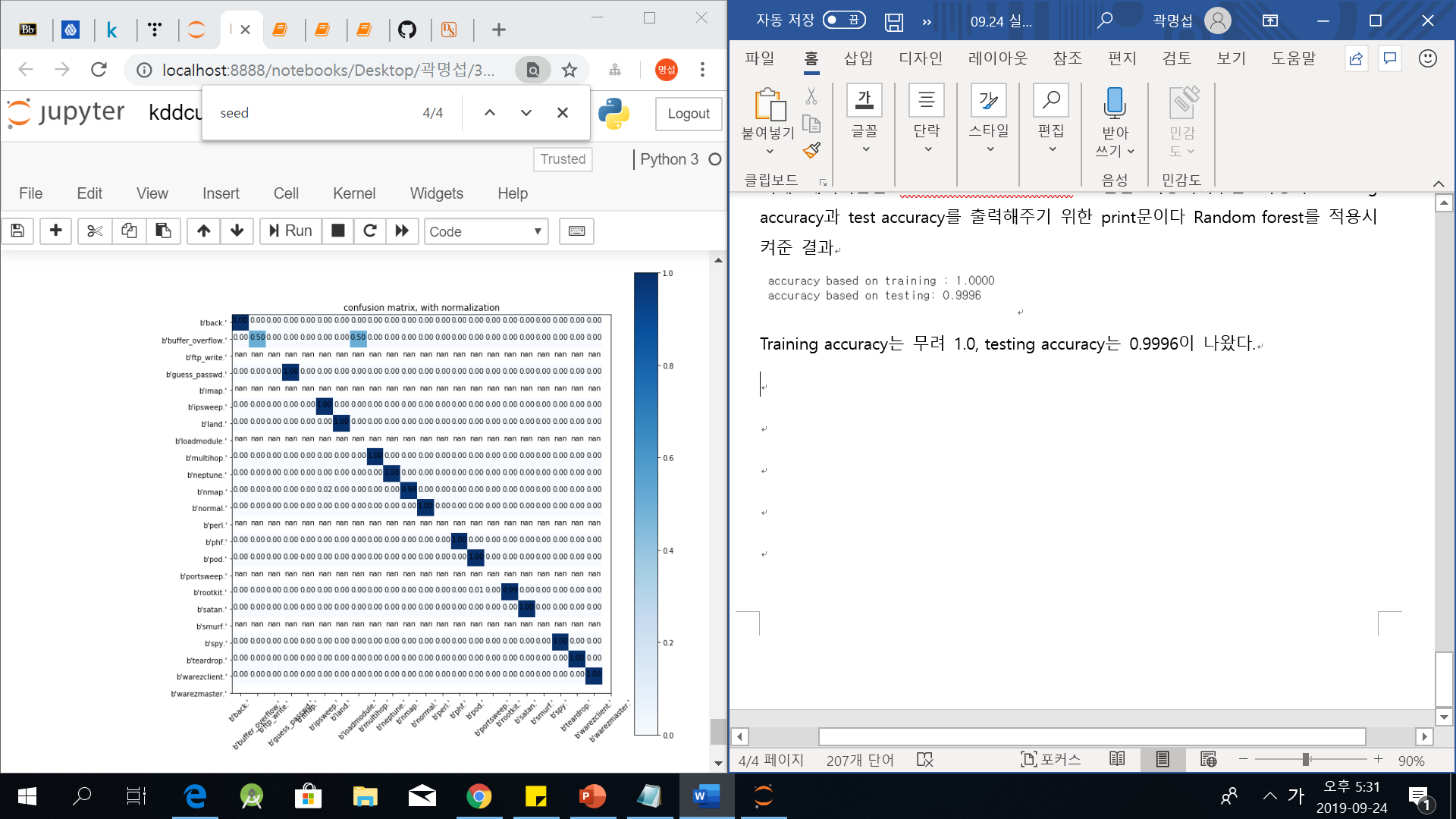


이제 데이터들을 RandomForestClassifier 모듈을 적용시켜주는 과정과 training accuracy과 test accuracy를 출력해주기 위한 print문이다 Random forest를 적용시켜준 결과



Training accuracy는 무려 1.0, testing accuracy는 0.9996이 나왔다.

normalization을 하지 않은 confusion matrix이다.

normalization을 한 confusion matrix이다.

x축과 y축은 예측결과와 실제 결과인데 대각선에 치중되어 있는 것으로 보아 잘 예측 했다는 것을 알 수 있다