**CS5590 APS - Python Programming**

**Lab 2.**

Fall 2018

UMKC

Kim-Ndor Djimadoumngar

1. **Introduction**

This is a report by Kim-Ndor Djimadoumngar for Assignment 2 of the Special Topic in Applied Programming Learning (APL) series: CS5590-0001 python and Deep Learning. My student ID is 6; my Lab ID is 7. The course is taught by Dr. Yugyung Lee and instructed by Saria Goudarzvand.

1. **Objectives**

The purposes of this lab are triple:

* To make categorical plots seaborn library or matplotlib
* To use the algorithms such as Naïve Bayes, Support Vector Machine with linear, rbf, and poly kernel to make predictive and classification models and report their accuracies
* To apply the techniques such as lemmatization, bigram, bigram-frequency, concatenation, and summarization on the text.

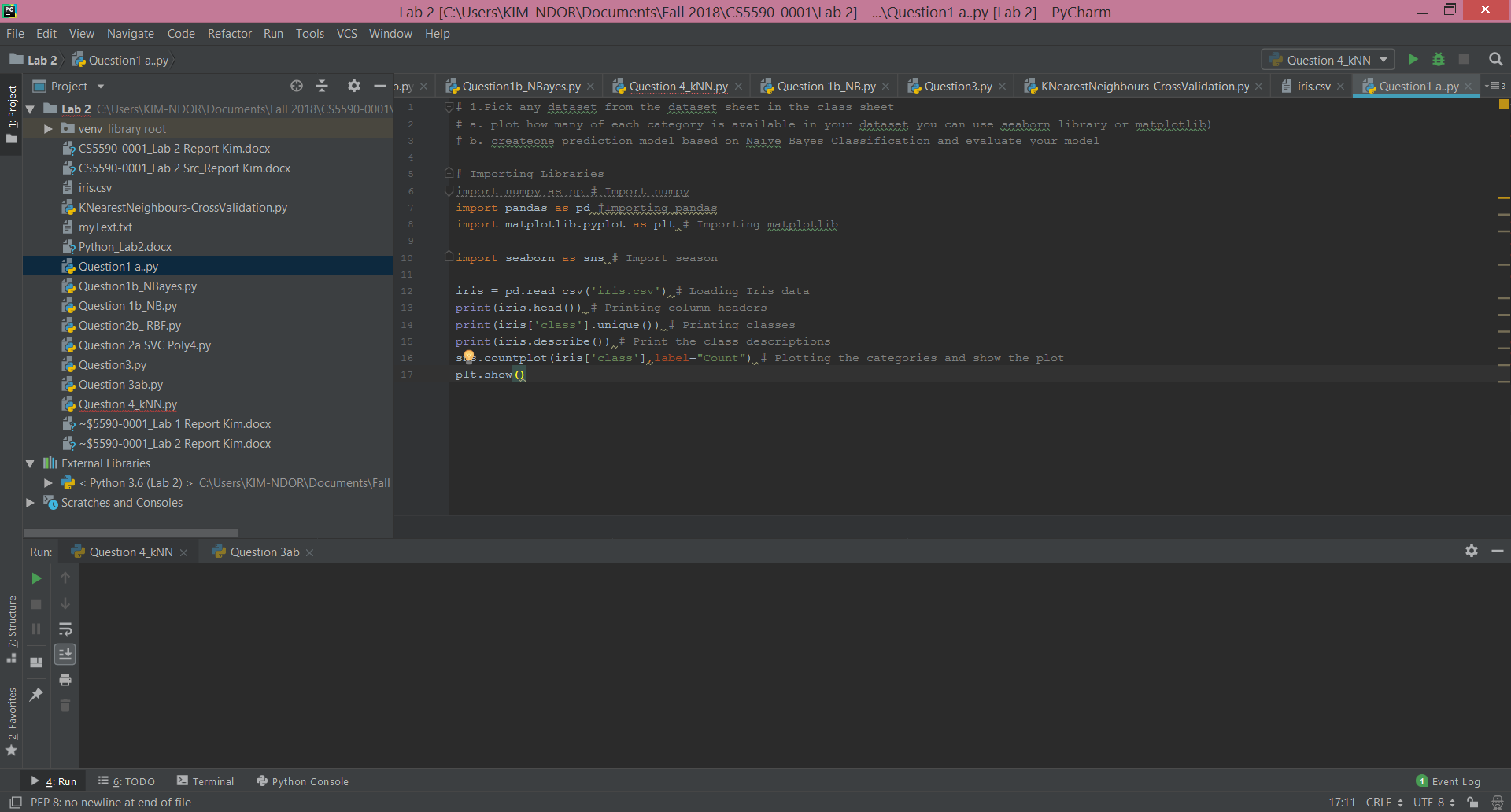
1. **Approaches/Methods**

Pycharm Community Edition 2018 was used to create Python in order to implement the exercises. Notepad was used to create txt files.

1. **Results/Discussion: Input/output**

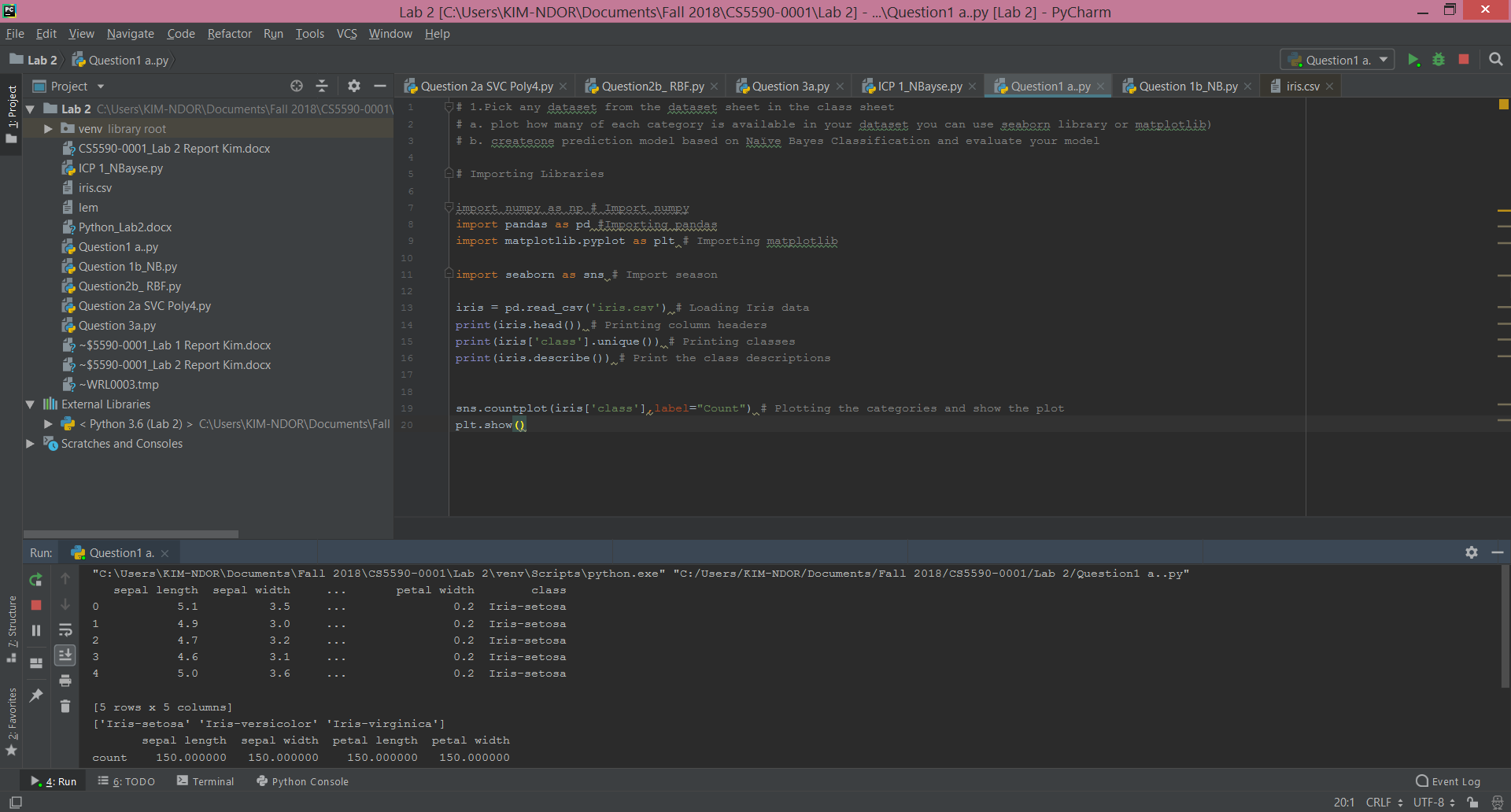
**Answers to Question 1**

Question 1.a: Plotting the classes in Iris dataset. The codes are shown below.

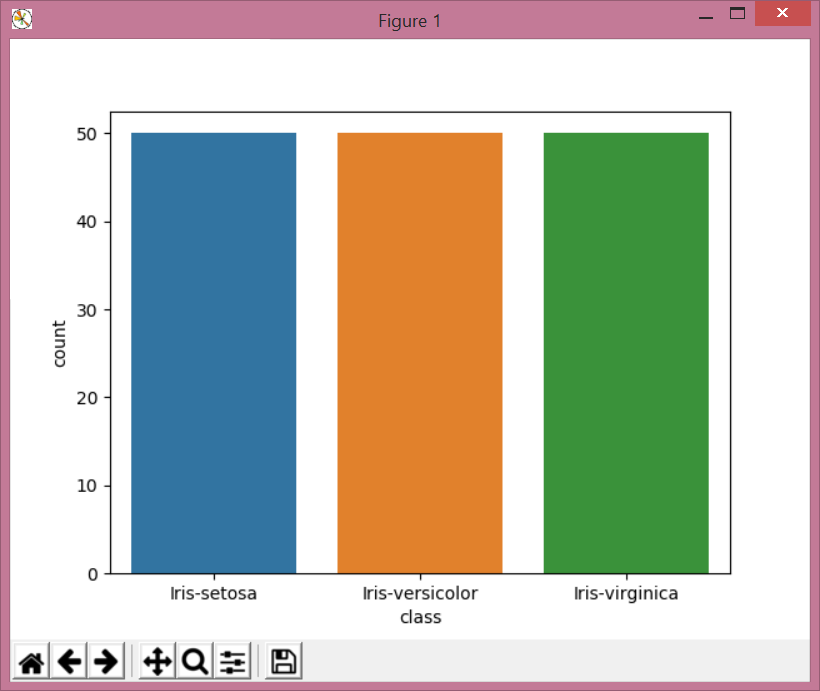


**Results**

The top section of the results is shown in the console.



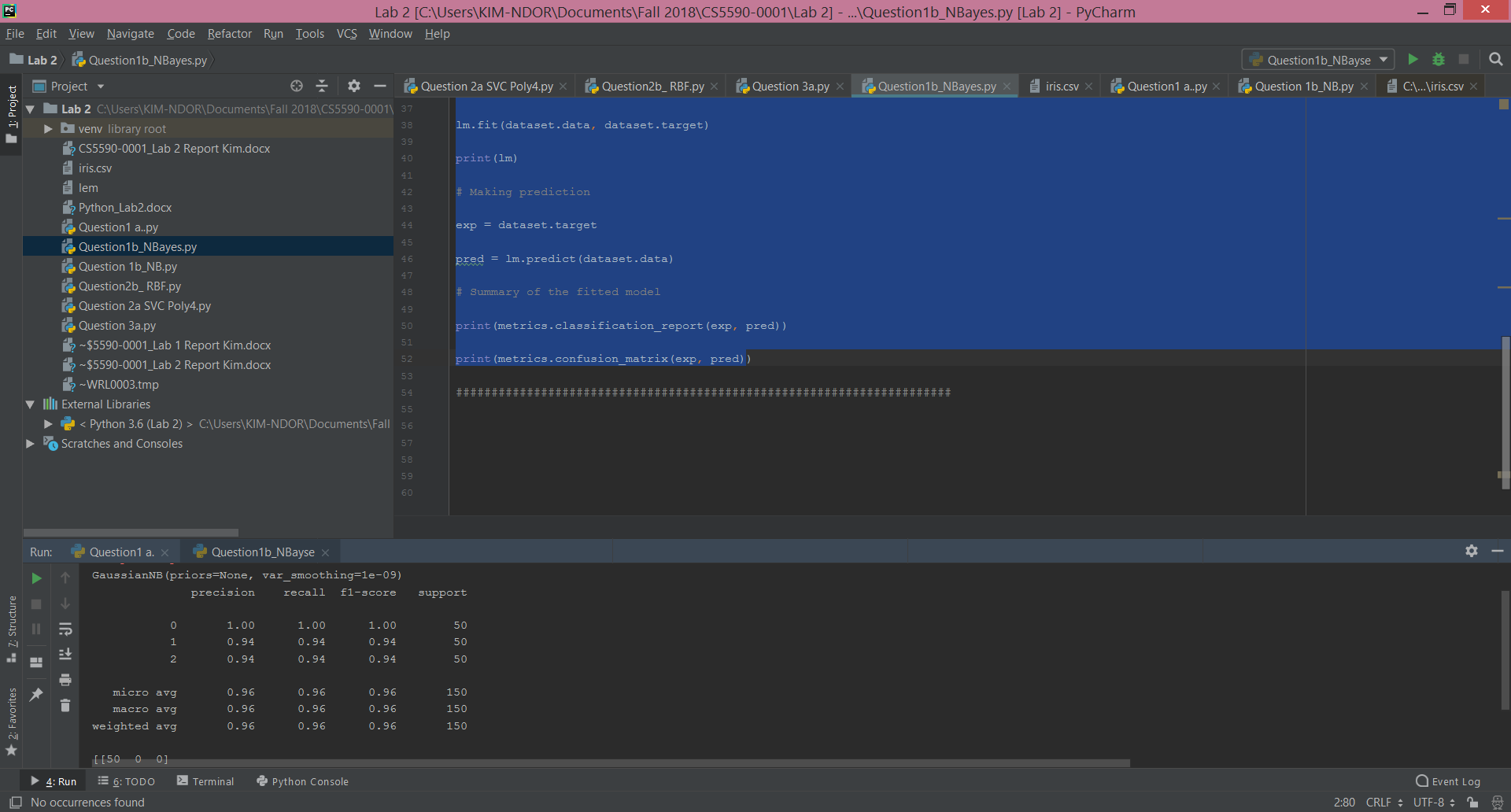
The categorical plot of the Iris dataset is below:



question 1.b: creating a predictive model using Naïve Bayes classifier. create one prediction model based on Naïve Bayes Classification and evaluate your model

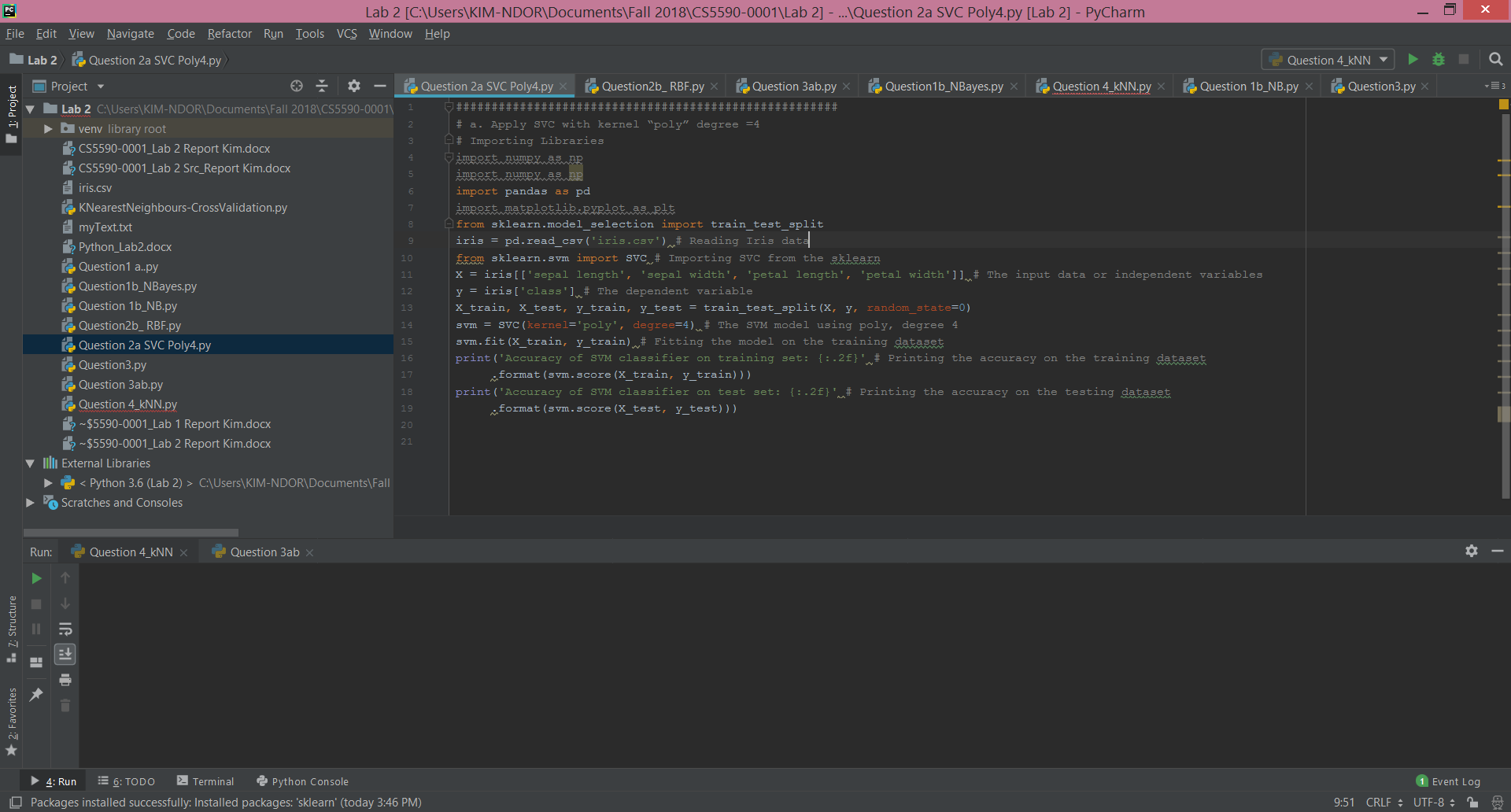
1. #############################################################  
   # #Naive Bayse (Gaussian)  
   #  
   from sklearn import datasets  
     
   from sklearn import metrics  
     
   from sklearn.naive\_bayes import GaussianNB  
     
   #Loading iris data  
     
   dataset = datasets.load\_iris()  
     
     
   # Fitting NB Model to the data  
     
   lm = GaussianNB()  
     
   lm.fit(dataset.data, dataset.target)  
     
   print(lm)  
     
   # Making prediction  
     
   exp = dataset.target  
     
   pred = lm.predict(dataset.data)  
     
   # Summary of the fitted model  
     
   print(metrics.classification\_report(exp, pred))  
     
   print(metrics.confusion\_matrix(exp, pred)

**The results are as below:**



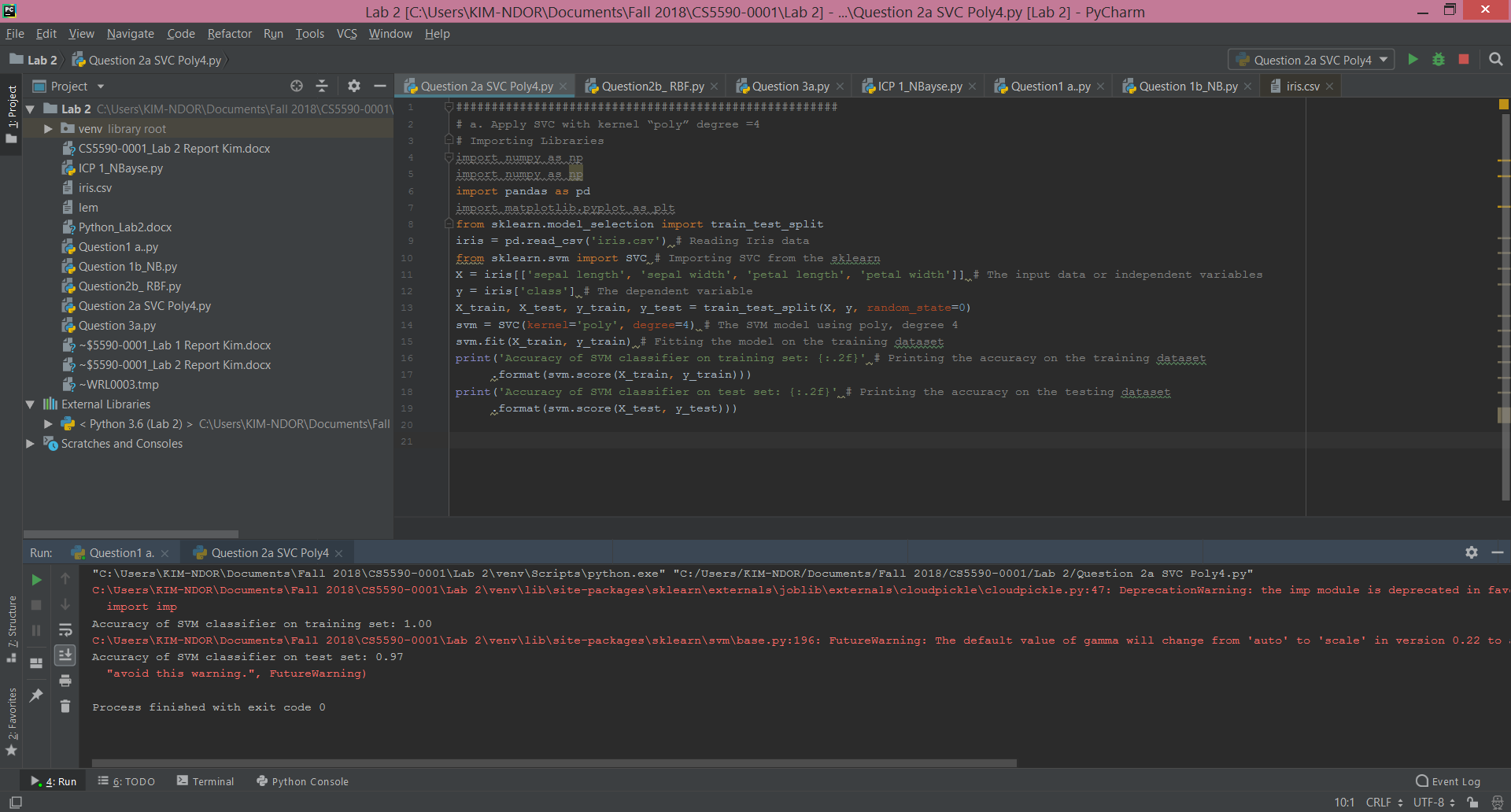
**Answers to question 2**

**Question 2.a: Application of SVC with poly, degree a kernel**



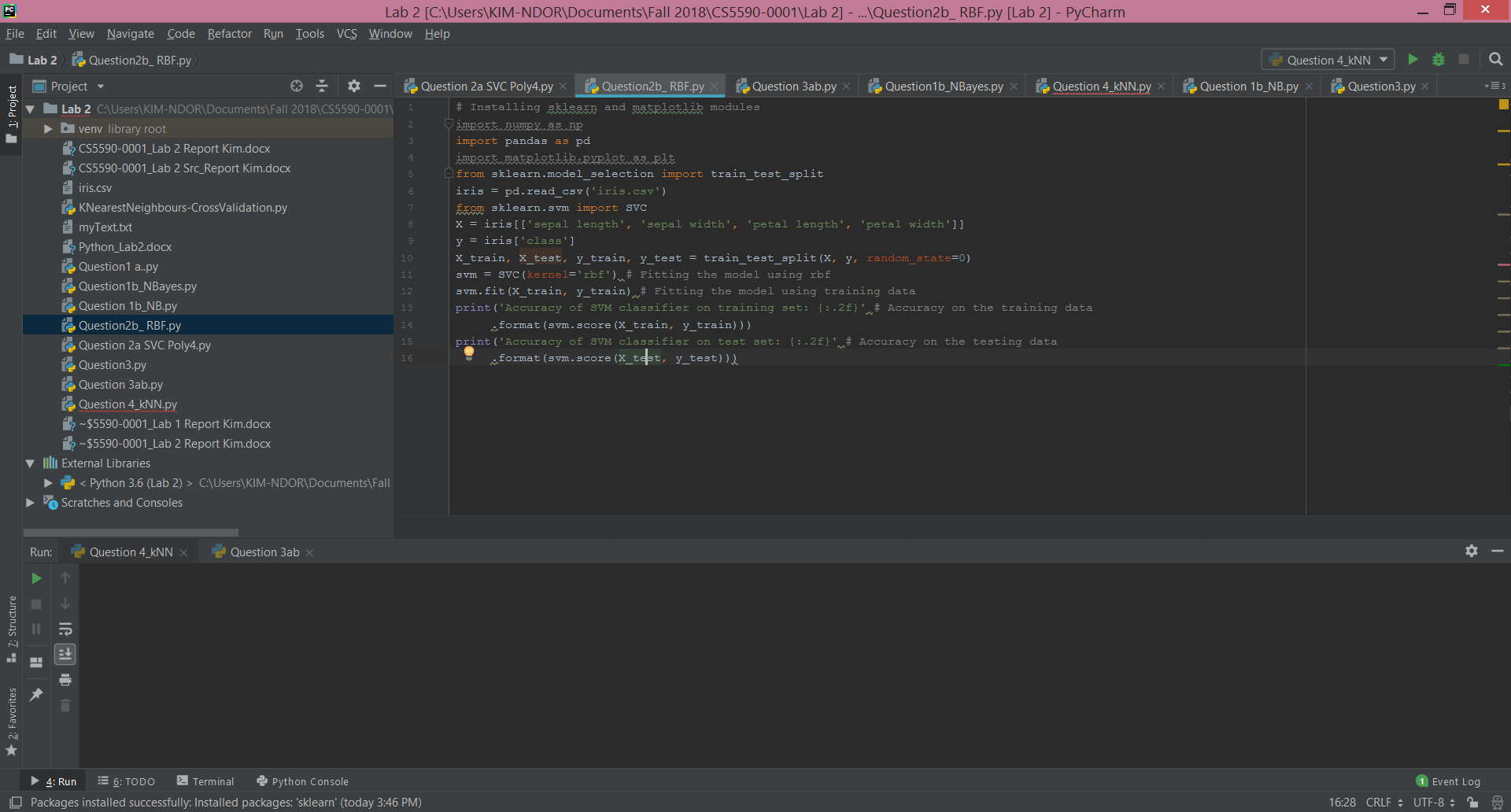
**Results**

The screenshot below show the output result of the above codes.

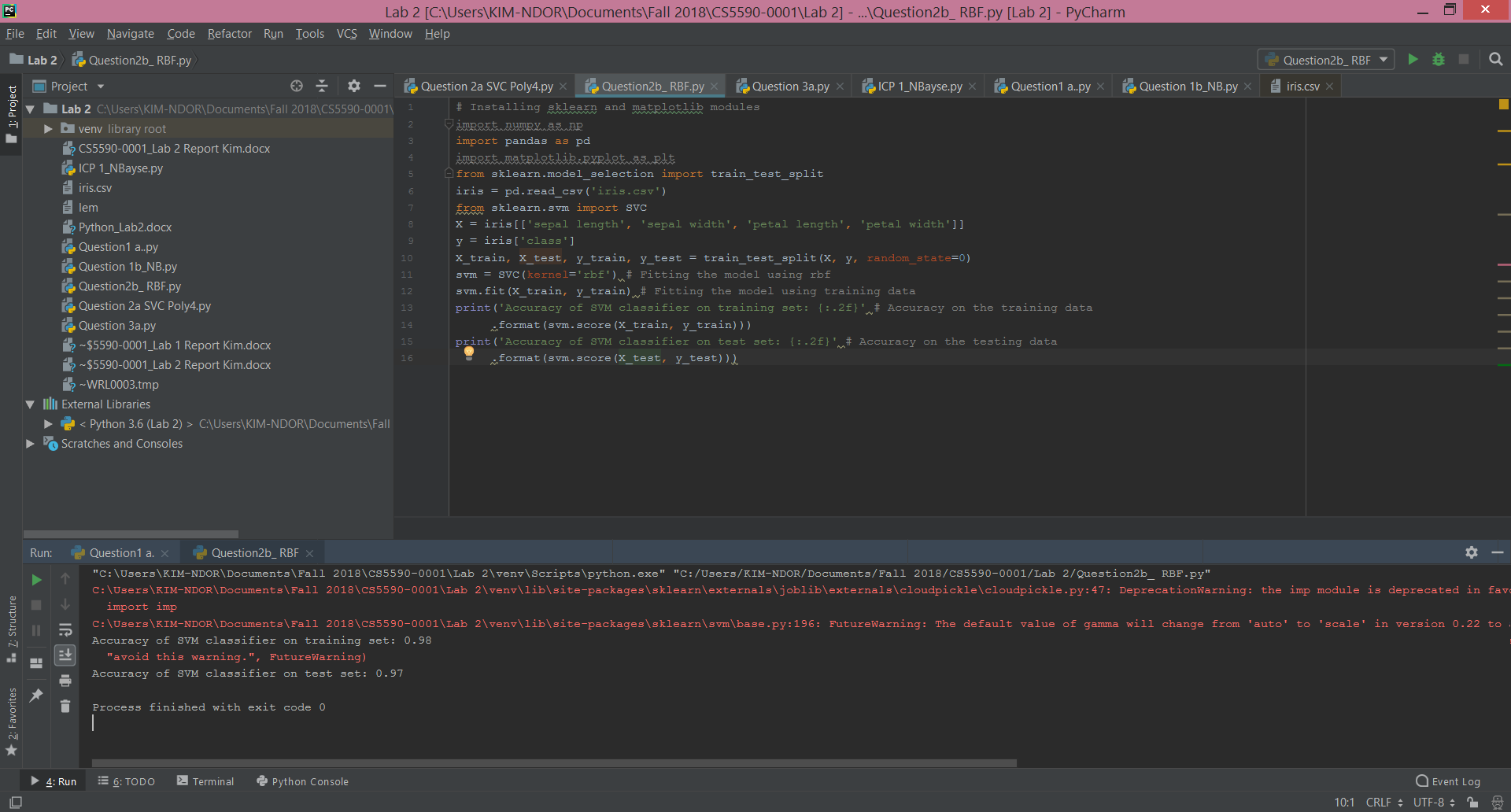


The accuracy on the training set is 100% while the testing accuracy is 97%.

**Question 2.b: Application of SVC with the rbf method**



**The results to question 2.b are as follows:**



The training set accuracy is 98% while the accuracy of testing dataset is 97%.

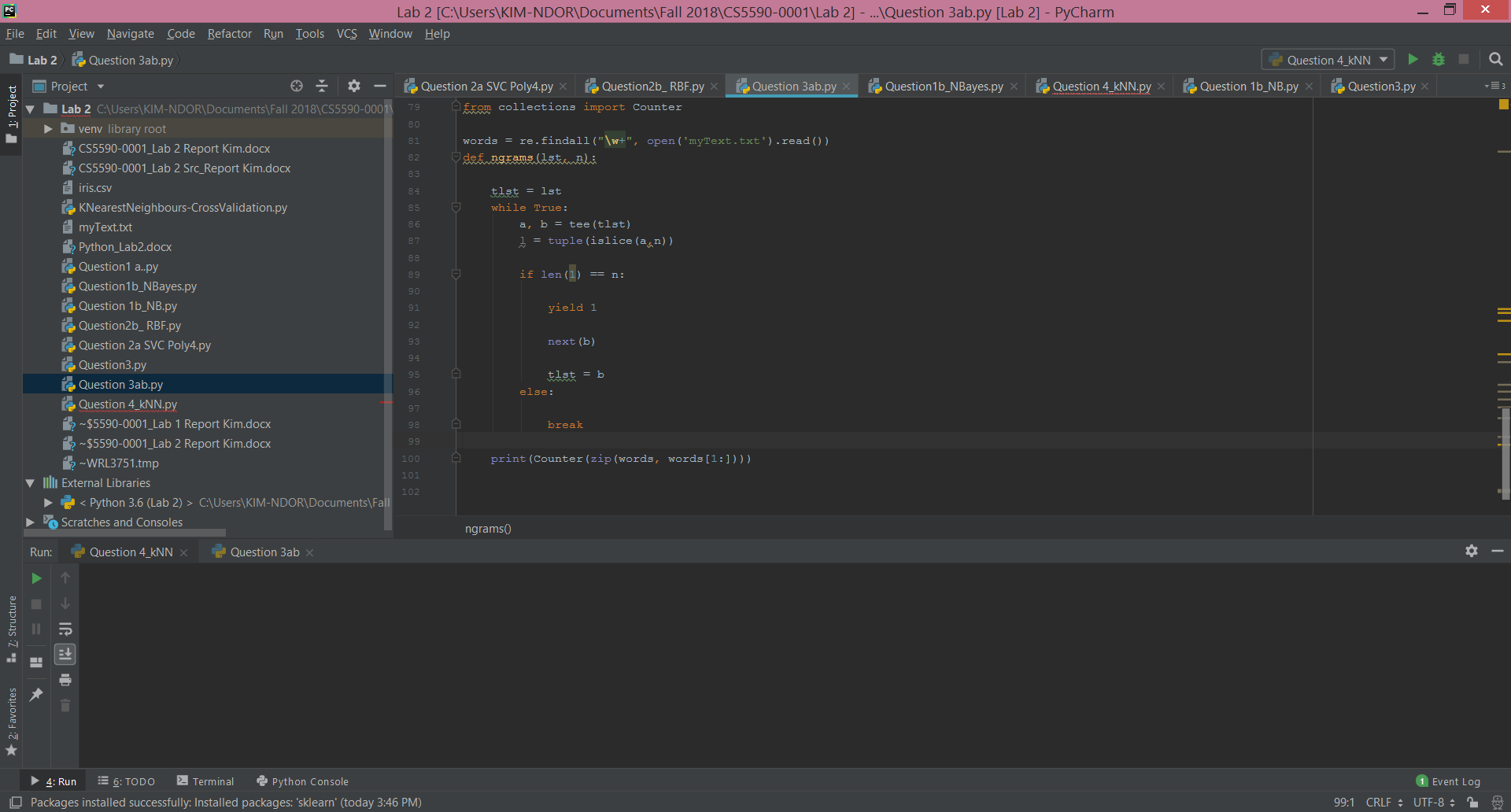
**Question 2.c and 2.d**

The accuracy is a little bit higher on training than on testing data.

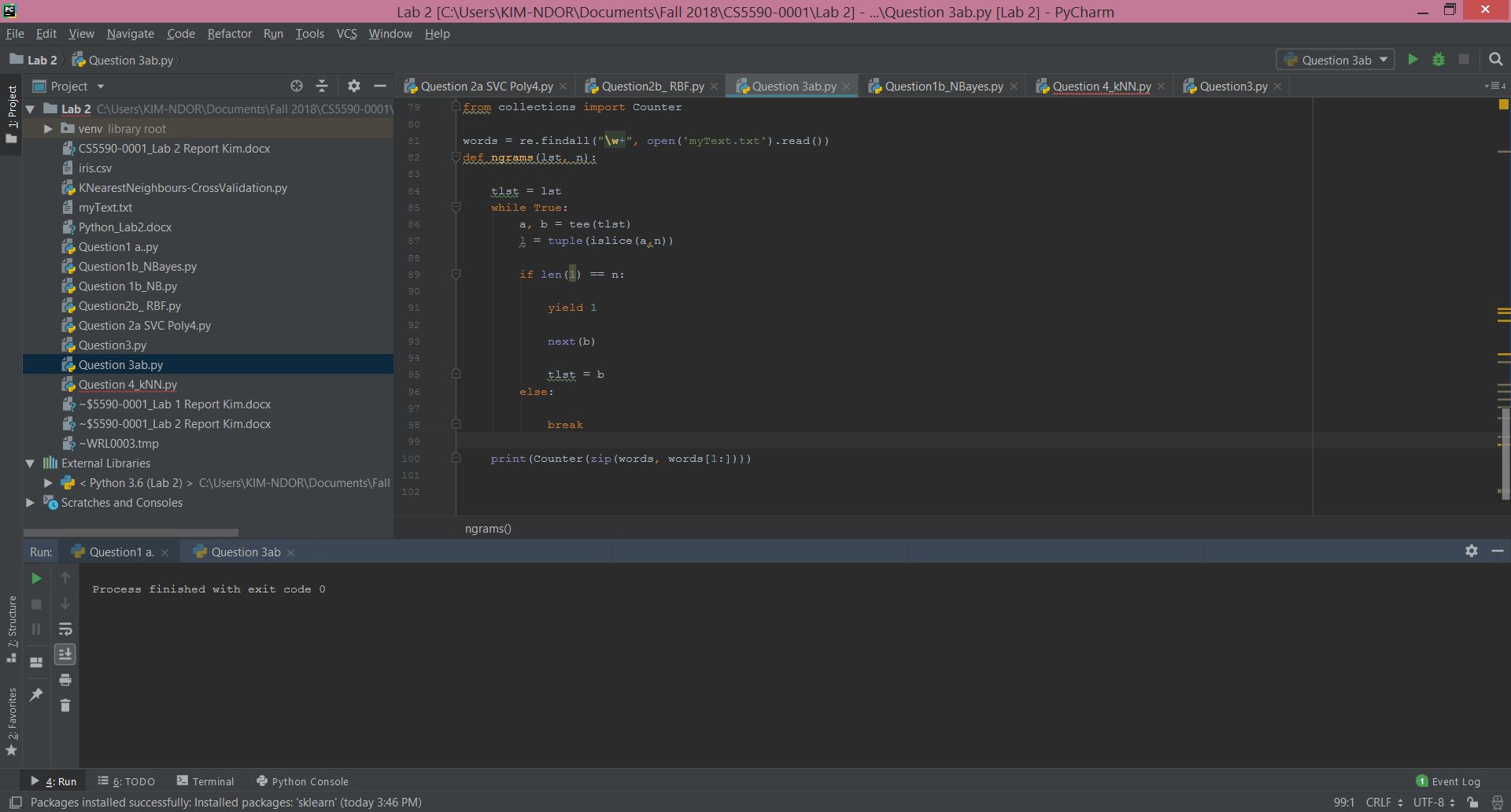
The model using poly, degree 4 performs better than the model with rbf

**Answers to question 3**

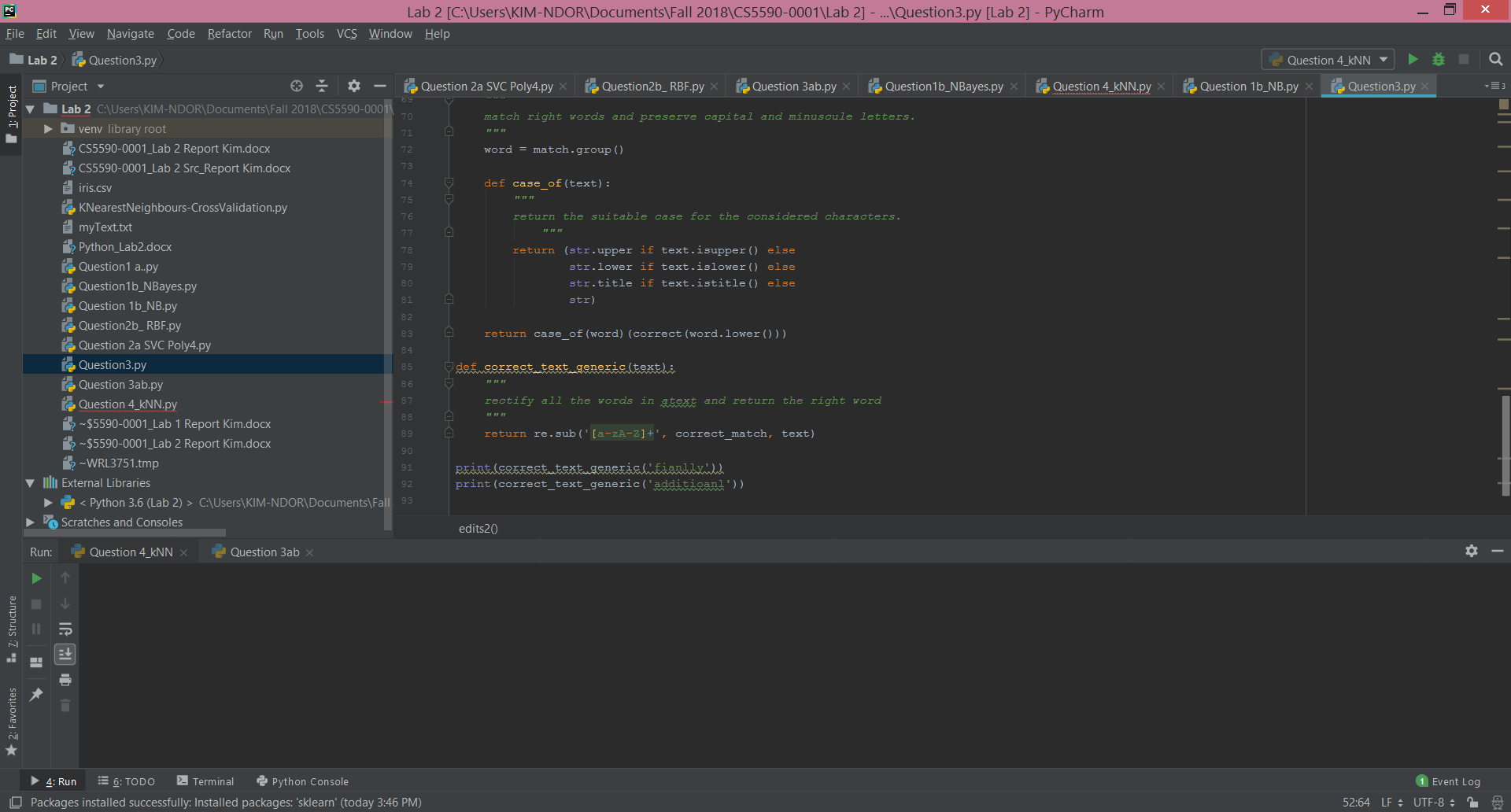
**questions 3.a & 3.b are combined.** The are codes are displayed below.



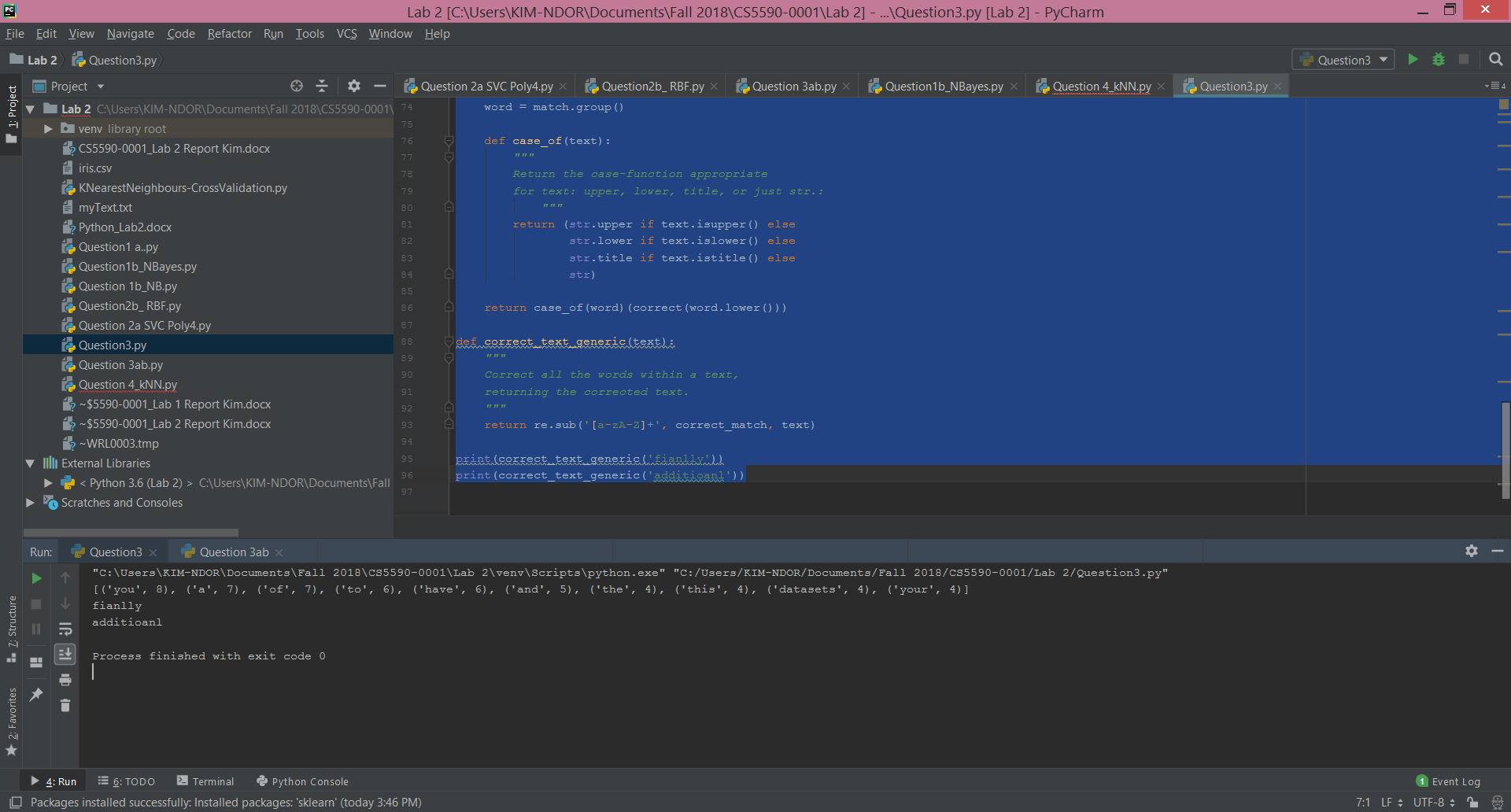
**Results to questions 3.a and 3.b**



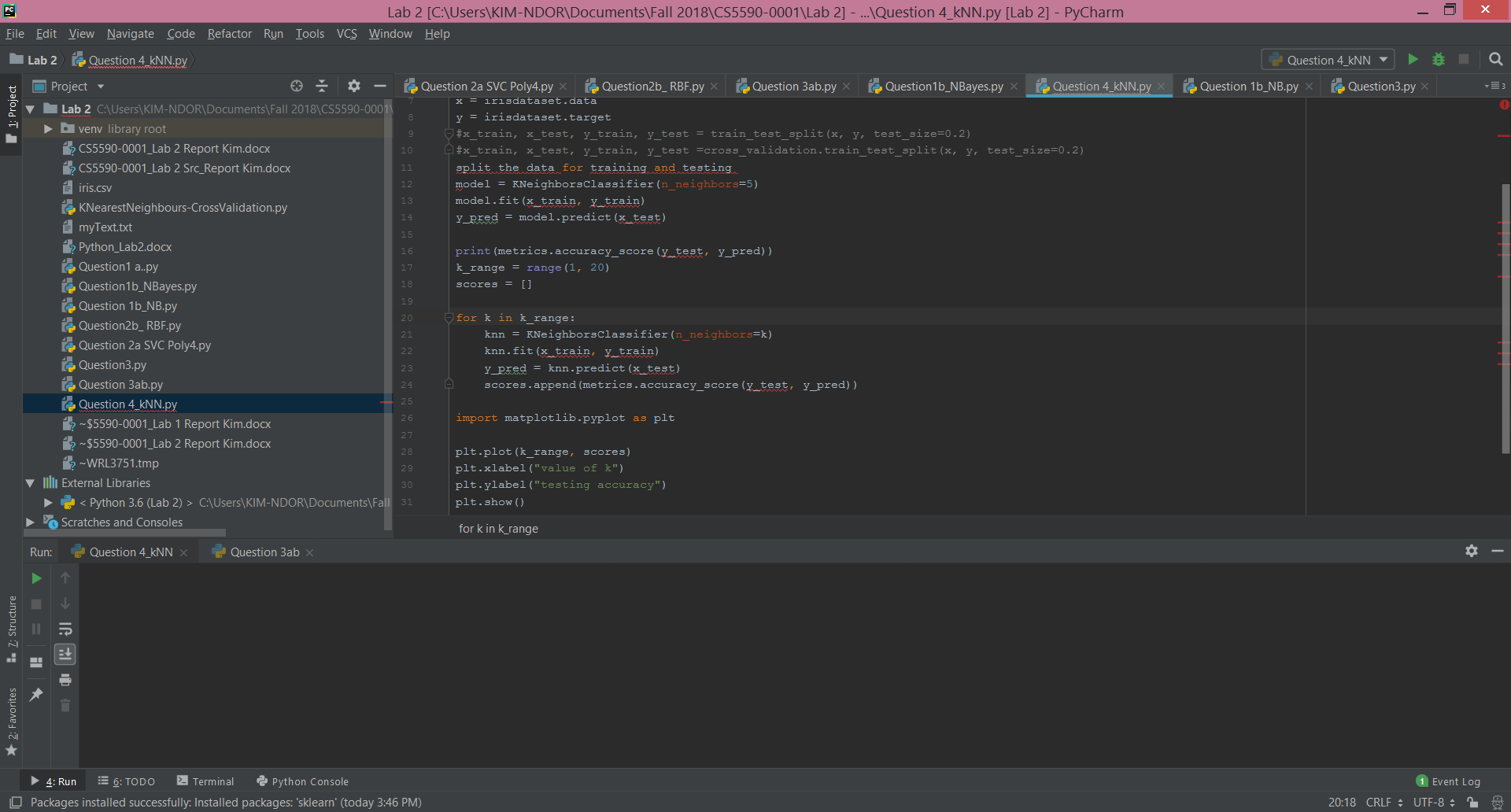
**Answers to questions 3.c through 3.1**



**Results obtained are as below.**

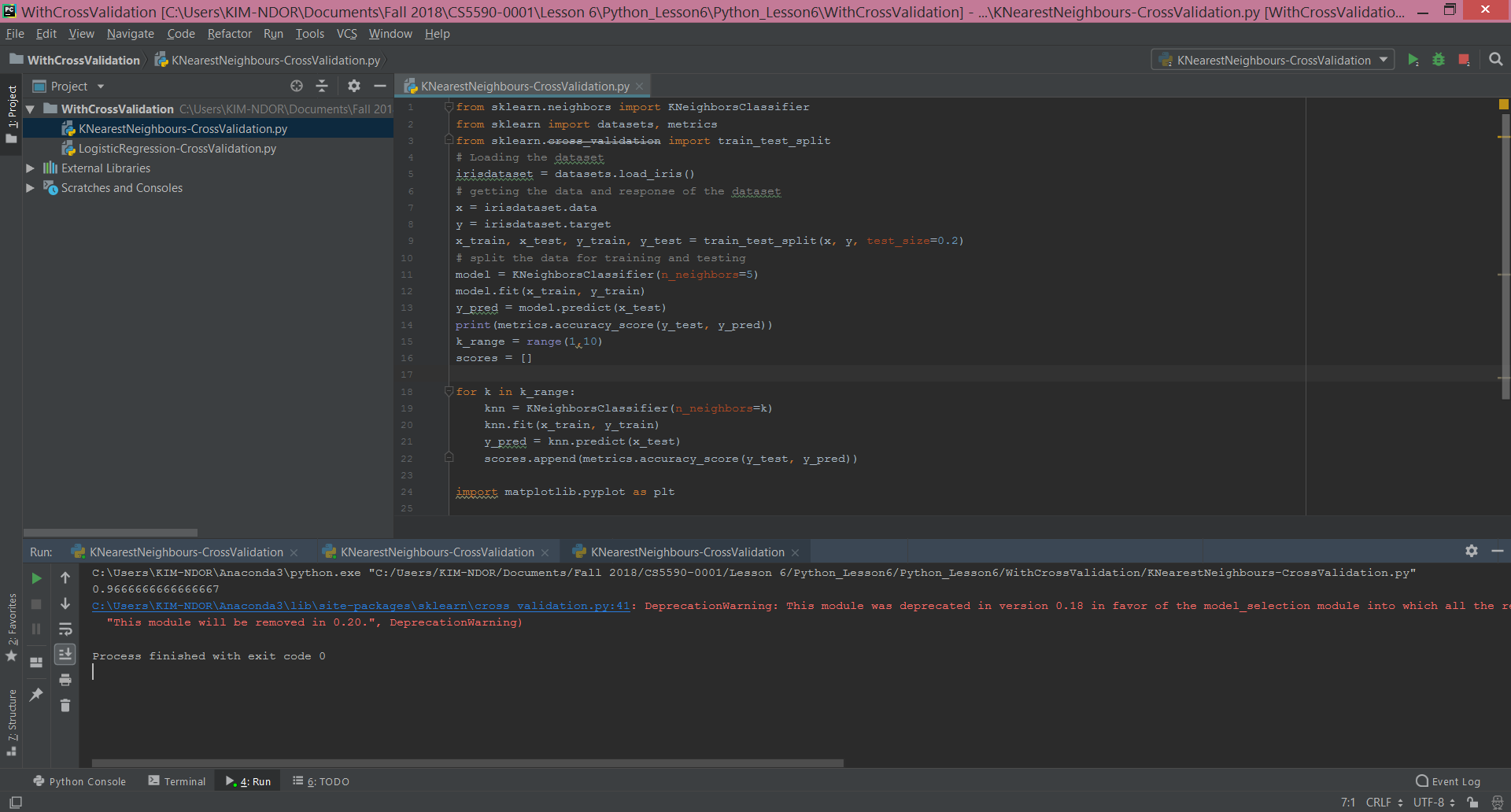


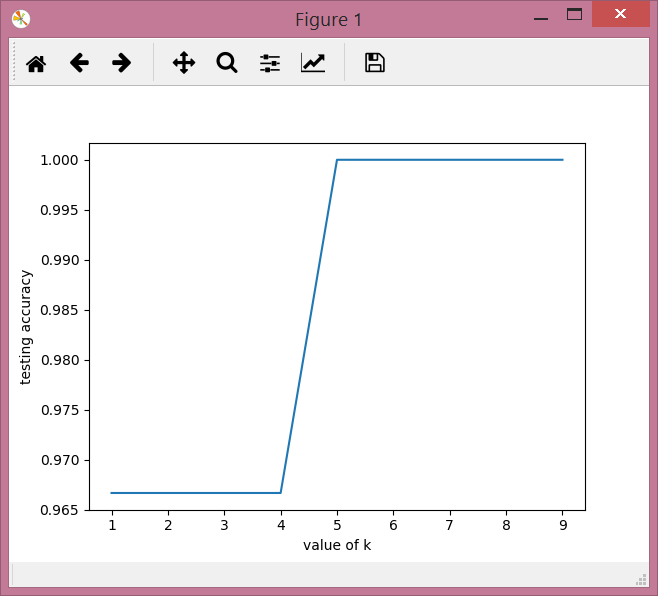
**Answers to question 4**



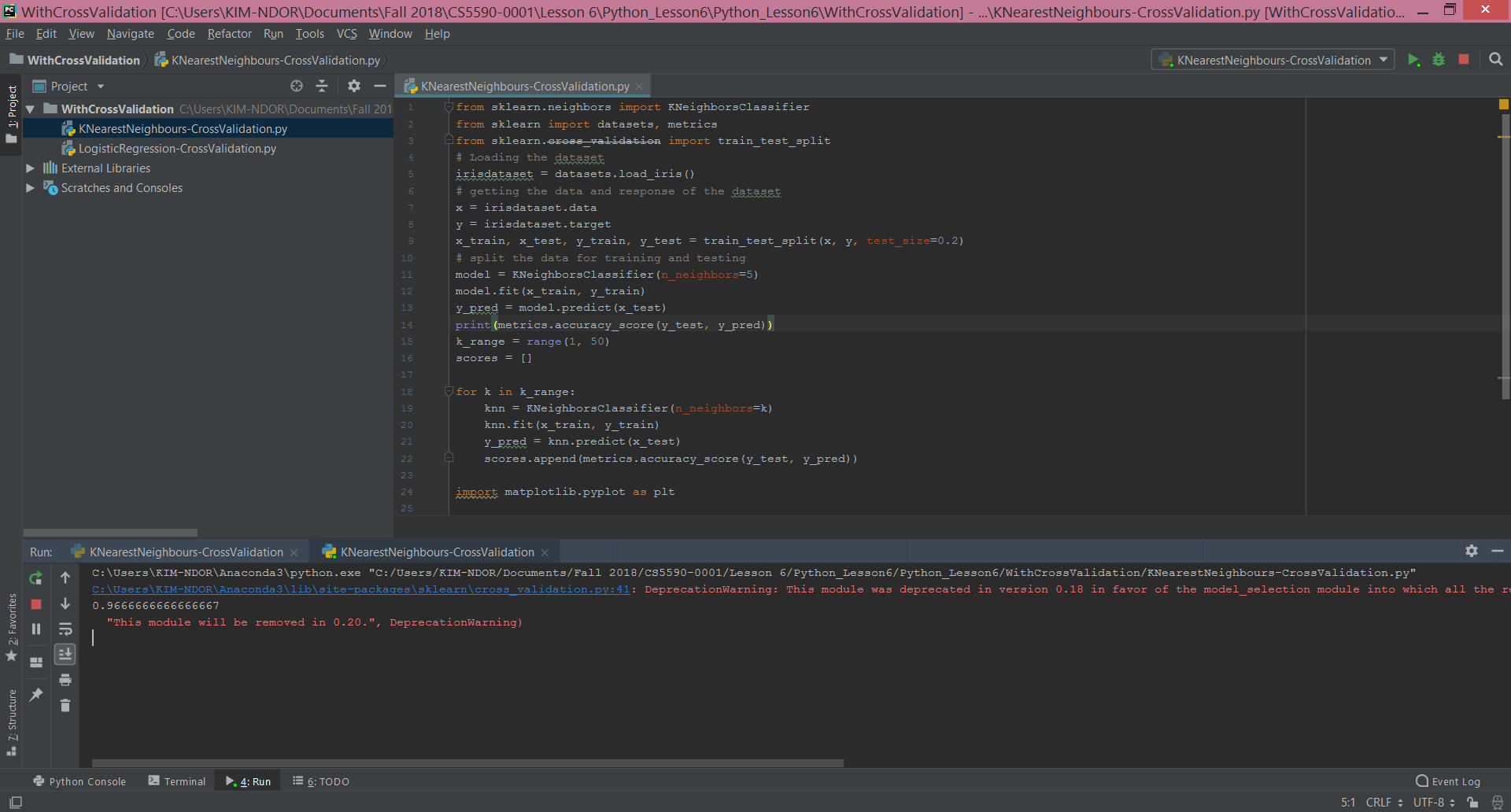
**Results**

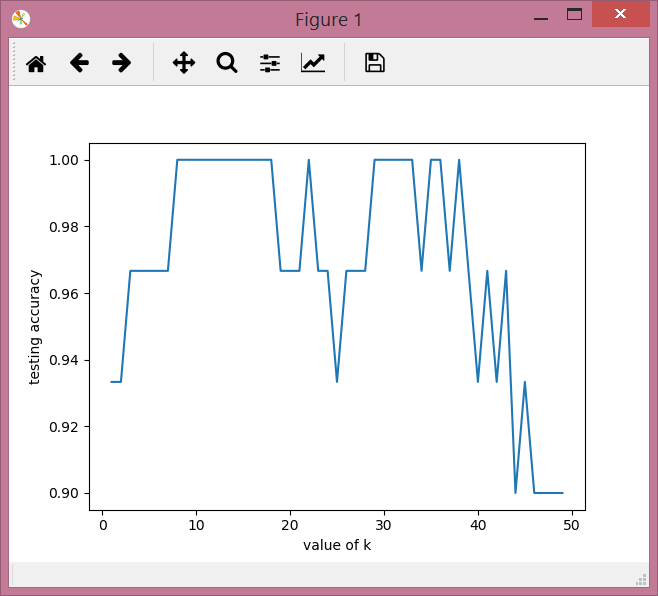
**K = 10**





**K = 50**





The accuracy is higher when k = 1 than k = 50. The accuracy increases from k = 1 to about k = 20. It oscillated from k = 20 to k = 40. Then decreases to k = 50 +.

**References**

[Python Exercises, Practice, Solution](http://www.w3resource.com/python-exercises/) (accessed on Oct. 10, 2018).

[Learningpython.org](https://www.learnpython.org/) (accessed on Oct. 9, 2018).

[Medium](https://medium.com/@sifium/machine-learning-types-of-classification-9497bd4f2e14) (accessed on Oct. 10, 2018).

Essentials of Machine Learning Algorithms (with Python and R Codes) at <https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/> (accessed on Oct. 10, 2018).