**ENDG 319 - Fall 23**

**CURE Project – Deliverable 4 – Groupwork**

**Maximum Points: 30**

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**Develop a machine learning model**

**Instructions:**

1. This deliverable is worth 6 % of your overall grade and should be completed as a group and submitted by a member of the group. The mark you get is based on the quality of the work.
2. The assignment will require you to submit (a) **the jupyter notebook file/python file** **containing all the codes for all the tasks and (b) a pdf file with file name** ENDG 319\_ Group No\_CURE Deliverable 4. See the ‘Submission notes’ at page 3 of this document (go to the end of this document).
3. Submit the **pdf file** to Assignments > Dropbox > ‘CURE Project Deliverable 4’ by Nov 19, 2023, 11:59 pm (MT).
4. You must submit this deliverable on time to be able to submit the upcoming deliverables.

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**Research skill:**

(i) Dealing with real-world data

(ii) Use of engineering tools – python

(iii) Developing a research question

(iv) Working in a group effectively.

(v) Problem solving

(vi) Finding information from literature, learn it and apply it in your research.

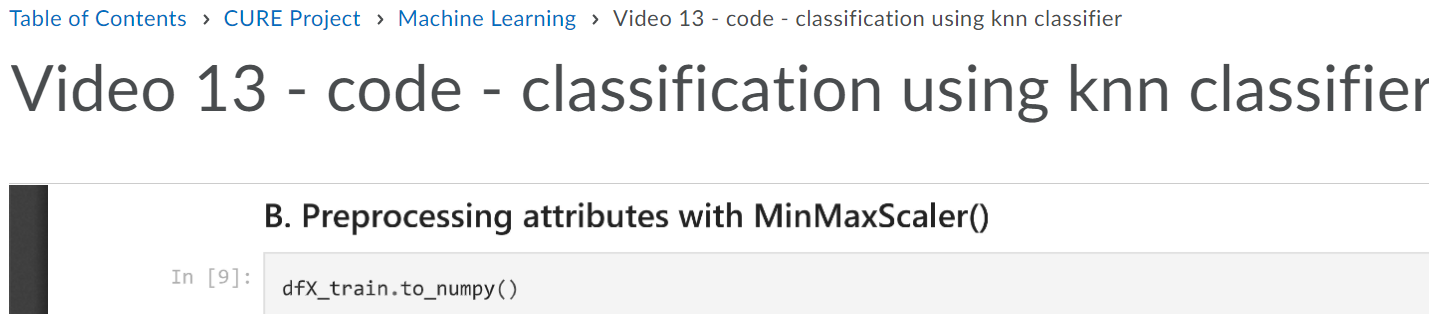
**Relevant course content:** Descriptive Statistics, Machine learning, Python

**>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>**

**Task 1 Develop a machine learning model [ 10 marks]**

1. Develop a knn-classifier model using Python. Use a fixed value of k such as 1. Preprocess attributes with MinMaxScaler(). Follow the steps shown in video 13.

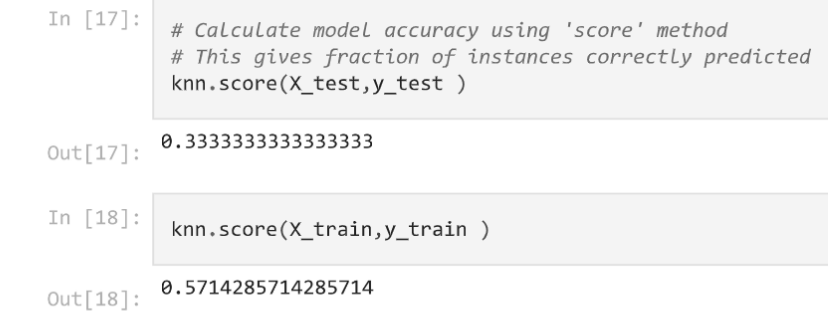
*[Hint: See the pdf file of the code, shown partly below*

]

**Task 2 Doing some analysis to find best value of k**

2(a). Show how ‘model accuracy’ varies with different k values in a graph. In the x-axis plot different k-values. In the y-axis, plot two variables: Model accuracy for training dataset and Model accuracy for test dataset. **[ 5 marks]**

[*Hint:* We used ‘score’ method to do this for a fixed k -value, as shown below.

]

You can generate the above plots in one of the following two ways (1) You can run the same program (developed in Task 1) multiple times by changing k-values manually and recording the accuracy values in a table. Then write another short program to generate the plots using the values in the table, or (2) You can modify the program (e.g, write a for loop) to get the accuracy values and generate the plots in one program.

2(b). Repeat Task 2(a) by using StandardScaler() instead of MinMaxScaler (). **[ 5 marks]**

**Task 3 Finding best value of k**

(a) Select the best value of k and best scaler (MinMax or Standard). This is your final model. **[ 3 marks]**

k = 5

(b) For the final model developed in Task 3(a), calculate and report the **confusion matrix** for test dataset used in model development. Show screenshot of the command for producing ‘confusion matrix’. Also briefly explain the confusion matrix. *Note: The concept ‘Confusion matrix’ was not covered in the video. As part of your research project, you have to learn this concept by yourself (Hint: go to the sklearn website that we showed your multiple times in different videos and find this topic. This is just a two-dimensional table that shows class accuracy).* [**5 marks**]

**ANSWER:**

A confusion matrix is a table that is often used to describe the performance of a classification model on a set of test data for which the true values are known. It's a useful tool to understand the performance of a classification algorithm by displaying the counts of the true positive, true negative, false positive, and false negative predictions made by the classifier.

The matrix is of the form:

[[true\_negative false\_positive]

[false\_negative true\_positive]]

Where:

* true\_negative: Number of instances correctly predicted as negative.
* true\_positive: Number of instances correctly predicted as positive.
* false\_negative: Number of instances wrongly predicted as negative.
* false\_positive: Number of instances wrongly predicted as positive.

Code:

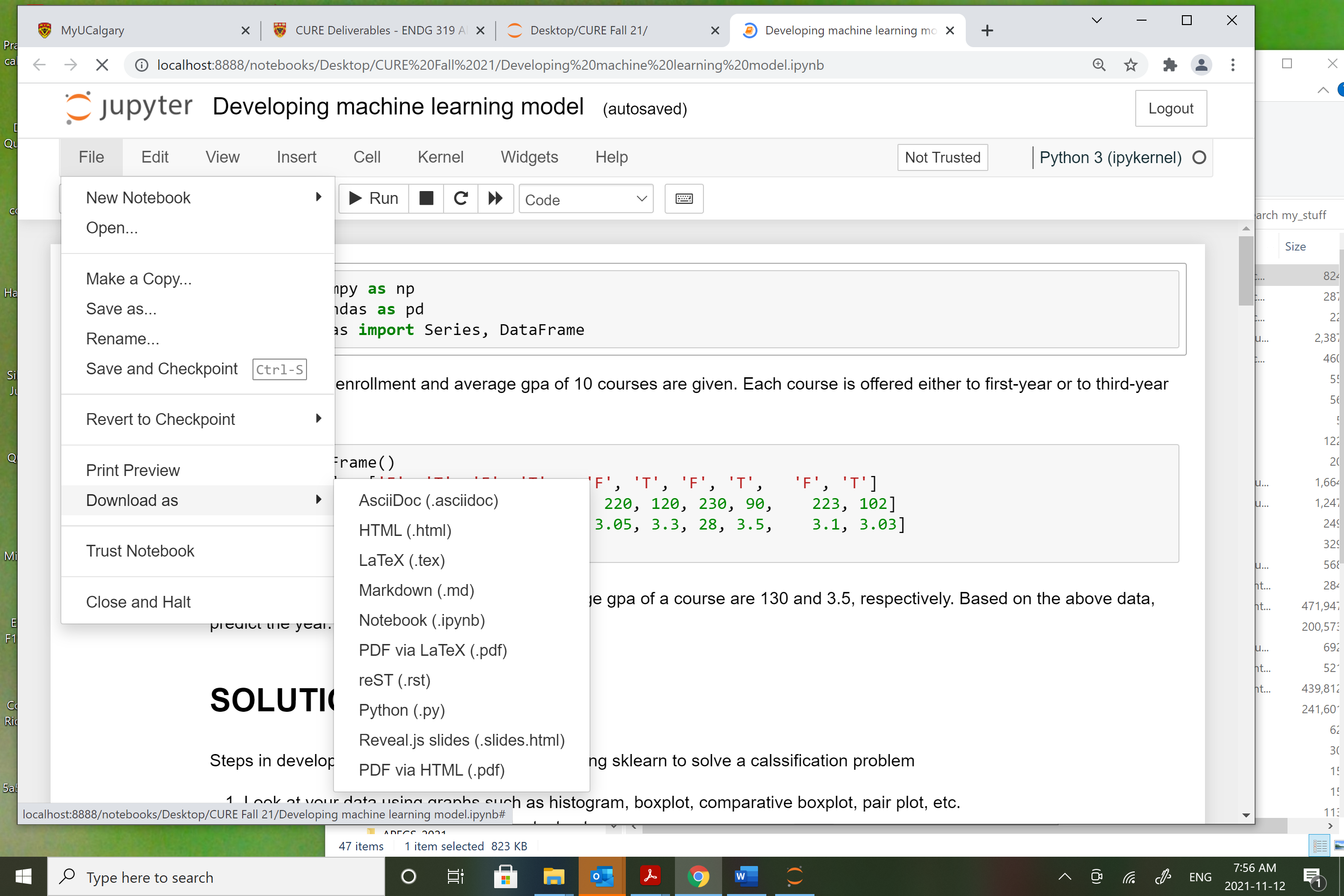
**Task 4 Apply your model**

Use your model to predict the class of an new instance that is not part of the training or test dataset. **[ 2 marks]**

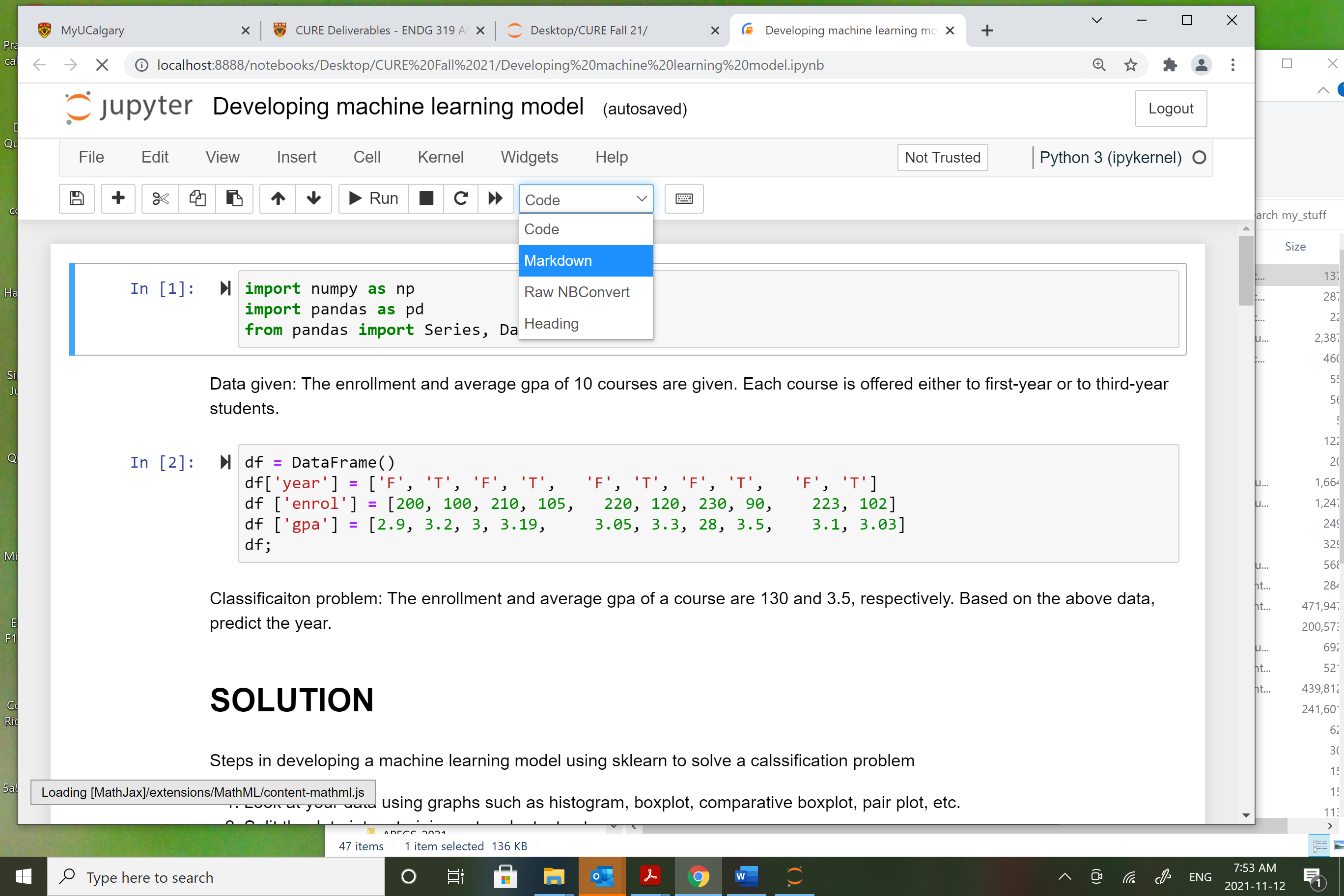
**Submission notes:**

1. Submit your code as a ipynb or py file. For all tasks, submit the code with output and comments for clarity.

*Further guidelines/hints: To save file as ipynb or py, go to File > Download as > ipynb or py as shown below.*



*To write comments in a cell, change the cell type to Markdown, type in comments such as Task numbering or any other short comments so that teaching assistants can follow your code easily, and run it. See below.*



2. In a separate pdf file, submit

(a) The graphs of model accuracy versus k values for different scenarios described in Task 2

(b) Final model parameters and scaler and accuracy:

k =

Scaler used for attribute preprocessing:

Model accuracy in the training set =

Model accuracy in the test set =

Confusion matrix:

Brief interpretations of confusion matrix:

(c) Ans of task 4.

New instance attribute values:

Predicted class: