ITP 449

Machine Learning

Lecture 7

Unsupervised learning

Identifies commonalities in the data and reacts based on the presence or absence of such commonalities in each new piece of data

Clustering (k-means)
Association Analysis

Supervised learning

Determine a function that maps an input to an output based on example input-output pairs

Estimation

- *Linear Regression*
- *Logistic Regression*

Classification

- *k-Nearest Neighbors*
- *Support Vector Machines (SVMs)*
- *Decision Trees*
- *Random Forests*
- *Neural Networks*

scikit-learn

https://scikit-learn.org/stable/

Settings

Project: ITP449Summer › Project Interpreter For current project

Project Interpreter: Python 3.8 (venv) C:\Users\Nitin Kale\venv\Scripts\python.exe

- Appearance & Behavior
- Keymap
- Editor
- Plugins
- Version Control
- Project: ITP449Summer
  - Project Interpreter
  - Project Structure
- Build, Execution, Deployment
- Languages & Frameworks
- Tools

Package	Version	Latest version
cycler	0.10.0	0.10.0
joblib	0.15.1	0.15.1
kiwisolver	1.2.0	1.2.0
matplotlib	3.2.1	3.2.1
numpy	1.18.4	▲ 1.18.5
pandas	1.0.4	1.0.4
pip	20.1.1	20.1.1
pyparsing	2.4.7	2.4.7
python-dateutil	2.8.1	2.8.1
pytz	2020.1	2020.1
scikit-learn	0.23.1	0.23.1
scipy	1.4.1	1.4.1
seaborn	0.10.1	0.10.1
setuptools	46.4.0	▲ 47.1.1
six	1.15.0	1.15.0
threadpoolctl	2.1.0	2.1.0

Package 'scikit-learn' installed successfully

OK Cancel Apply

Unsupervised learning

No target (response) variable

k-means Used to segment members (items) into unique categories based on a variety of features

Association Analysis: Find rules that show the affinity between items in a set. In a shopping cart, *Salad → Salad dressing*

Supervised learning

Target variable: Predict the value or category of a target (response) variable based on a group of predictor variables (features, influencers)

Linear regression Used to predict *continuous numeric* outcomes

Logistic regression *Classification* algorithm

k-Nearest Neighbors (KNN) Classification algorithm that classifies data into two or more *categories*

Support Vector Machines (SVM) Classification algorithm that is used in *image* and *face* detection

Tree-Based algorithms Tree-based algorithms such as decision trees and Random Forests are used to solve both *classification* and *regression* problems

Näive Bayes Uses the mathematical model of probability to solve classification problems

scikit-learn steps

1. Choose a *class* of model by importing the appropriate estimator class from Scikit-Learn.
2. Choose model *parameters* by instantiating this class with desired values.
3. Arrange data into a *features* matrix and *target* vector following the discussion from before.
4. *Fit* the model to your data by calling the *fit*() method of the model instance.
5. *Apply* the model to new data: For supervised learning, predict labels for unknown data using the predict() method. For unsupervised learning, transform or infer properties of the data using the transform() or predict() method.

Data representation

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Current events
Random article
About Wikipedia
Contact us
Donate

Contribute
Help
Learn to edit
Community portal
Recent changes
Upload file

Tools
What links here
Related changes
Special pages
Permanent link
Page information
Cite this page
Wikidata item

Print/export
Download as PDF
Printable version

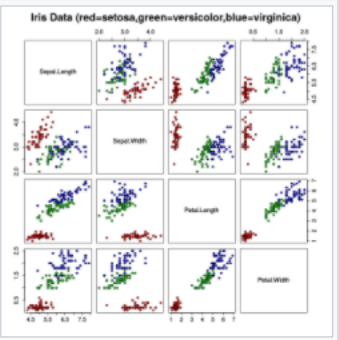
Languages
العربية
Español
فارسی
Français
Italiano
Русский

Iris flower data set

From Wikipedia, the free encyclopedia

The ***Iris* flower data set** or **Fisher's *Iris* data set** is a multivariate data set introduced by the British statistician, eugenicist, and biologist Ronald Fisher in his 1936 paper *The use of multiple measurements in taxonomic problems* as an example of linear discriminant analysis.[1] It is sometimes called **Anderson's *Iris* data set** because Edgar Anderson collected the data to quantify the morphologic variation of *Iris* flowers of three related species.[2] Two of the three species were collected in the Gaspé Peninsula "all from the same pasture, and picked on the same day and measured at the same time by the same person with the same apparatus".[3] Fisher's paper was published in the journal, the Annals of Eugenics, creating controversy about the continued use of the Iris dataset for teaching statistical techniques today.

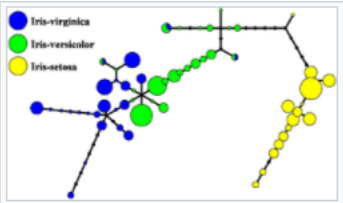
The data set consists of 50 samples from each of three species of *Iris* (*Iris setosa*, *Iris virginica* and *Iris versicolor*). Four features were measured from each sample: the length and the width of the sepals and petals, in centimeters. Based on the combination of these four features, Fisher developed a linear discriminant model to distinguish the species from each other.



Scatterplot of the data set

Contents [hide]

Use of the data set [edit]



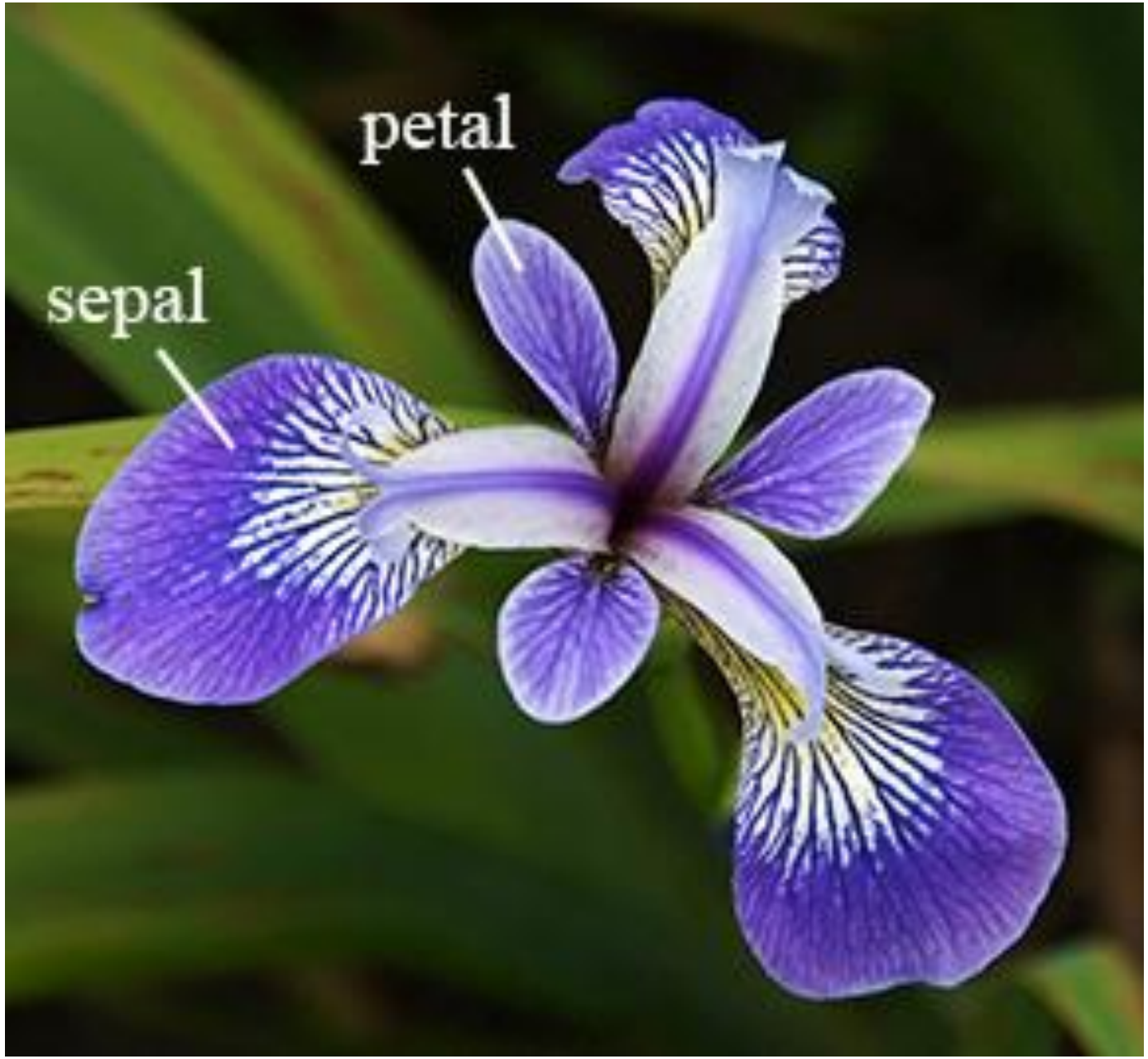
An example of the so-called "metro map" for the *Iris* data set[4] Only a small fraction of *Iris-virginica* is mixed with *Iris-versicolor*. All other samples of the different *Iris* species belong to the different nodes.

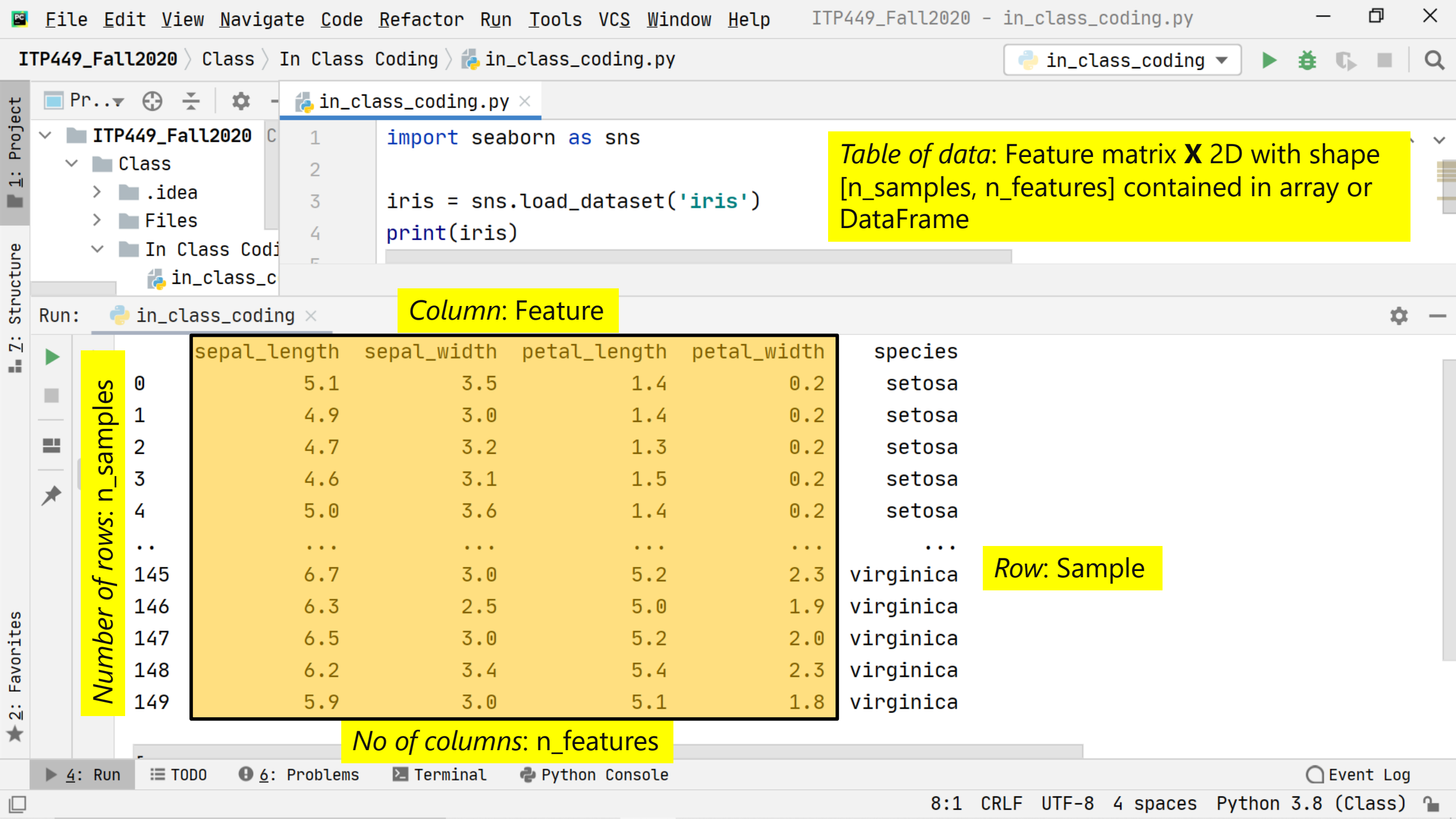
Based on Fisher's linear discriminant model, this data set became a typical test case for many statistical classification techniques in machine learning such as support vector machines.[5]

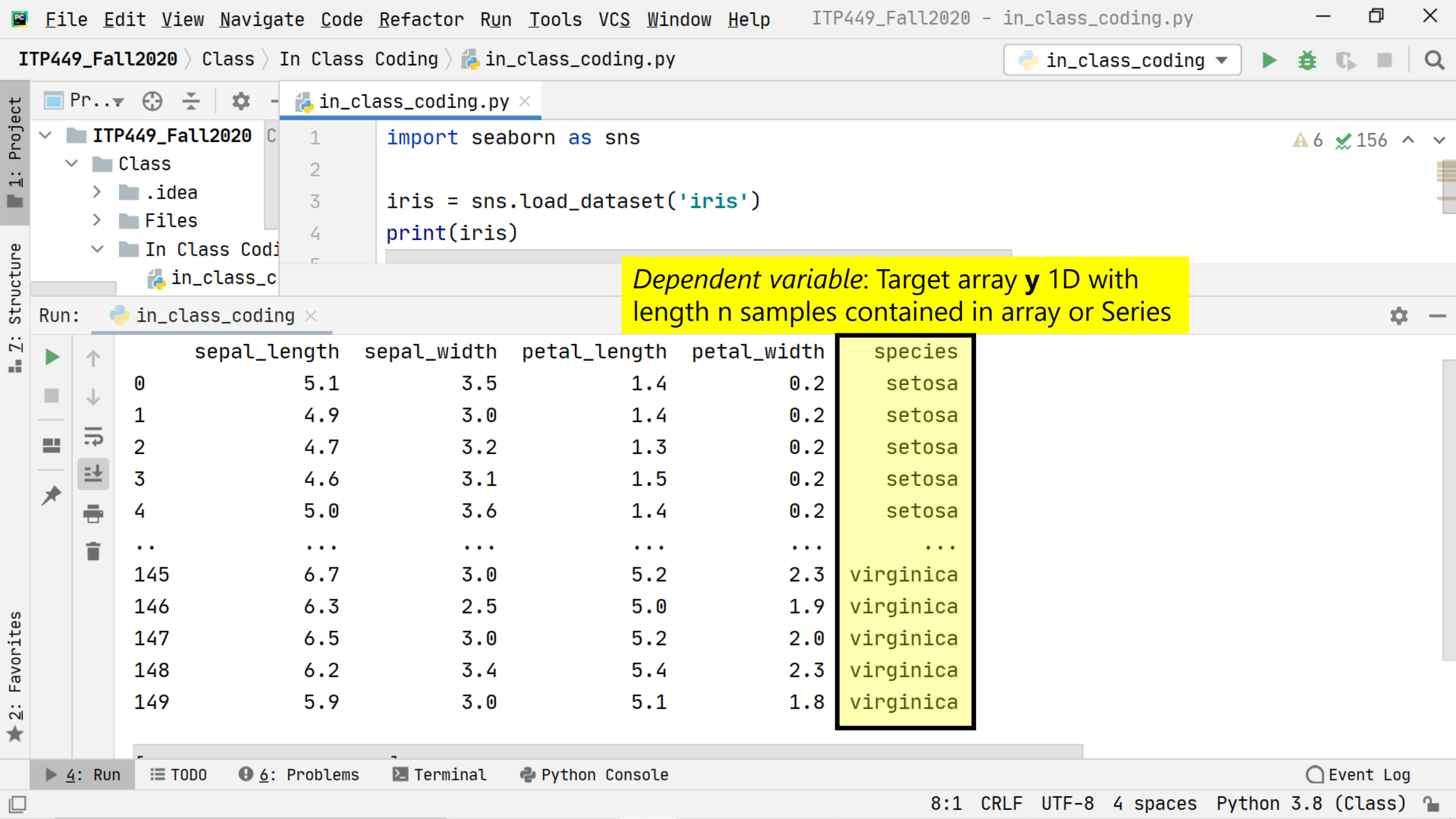
The use of this data set in cluster analysis however is not common, since the data set only contains two clusters with rather obvious separation. One of the clusters contains *Iris setosa*, while the other cluster contains both *Iris virginica* and *Iris versicolor* and is not separable without the species information Fisher used. This makes the data set a good example to explain the difference between supervised and unsupervised techniques in data mining: Fisher's linear discriminant model can only be obtained when the object species are known: class labels and clusters are not necessarily the same.[6]

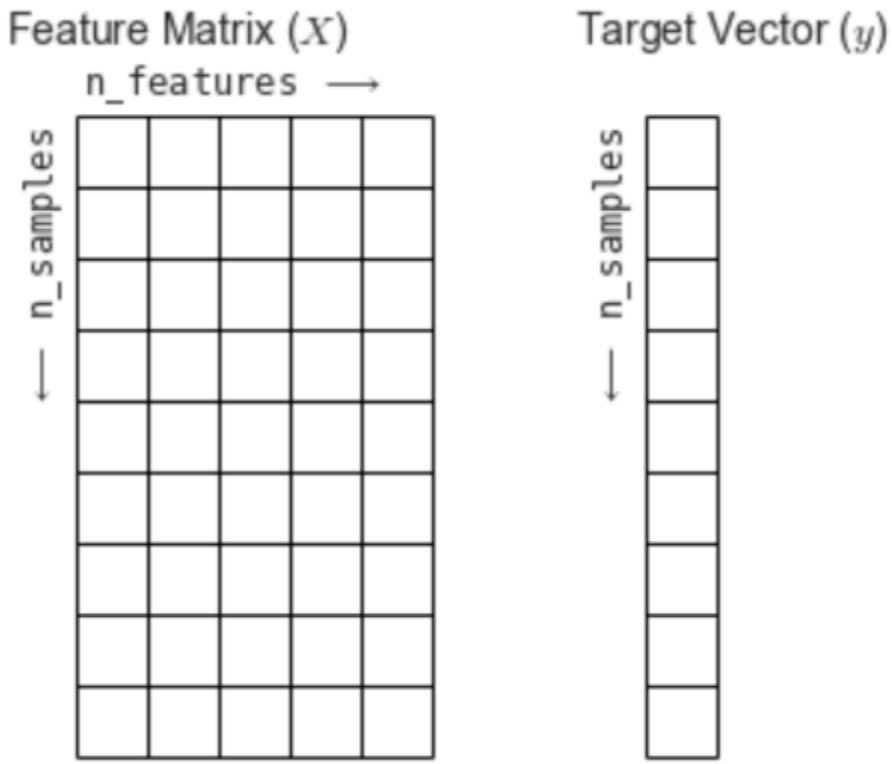
Unsatisfactory k-means clustering (the data cannot be clustered into the known classes) and actual species visualized using ELKI

Nevertheless, all three species of *Iris* are separable in the projection on the nonlinear and branching principal component.[7] The data set is approximated by the closest tree with some penalty for the excessive number of nodes, bending and stretching. Then the so-called "metro map" is constructed.[4] The data points are projected into the closest node. For each node the pie diagram of the projected points is prepared. The area of the pie is proportional to the number of the projected points. It is clear from the diagram (left) that the absolute majority of the samples of the different *Iris* species belong to the different nodes. Only a small fraction of *Iris-virginica* is mixed with *Iris-versicolor* (the mixed blue-green nodes in the diagram). Therefore, the three species of Iris (*Iris setosa*, *Iris virginica* and *Iris versicolor*) are separable by the


petal
sepal


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
in_class_coding.py
ITP449_Fall2020
Class
.idea
Files
In Class Codi
in_class_c
1 import seaborn as sns
2
3 iris = sns.load_dataset('iris')
4 print(iris)
Table of data: Feature matrix X 2D with shape [n_samples, n_features] contained in array or DataFrame
Run: in_class_coding
Column: Feature
Number of rows: n_samples
sepal_length sepal_width petal_length petal_width species
0 5.1 3.5 1.4 0.2 setosa
1 4.9 3.0 1.4 0.2 setosa
2 4.7 3.2 1.3 0.2 setosa
3 4.6 3.1 1.5 0.2 setosa
4 5.0 3.6 1.4 0.2 setosa
.. ..
145 6.7 3.0 5.2 2.3 virginica
146 6.3 2.5 5.0 1.9 virginica
147 6.5 3.0 5.2 2.0 virginica
148 6.2 3.4 5.4 2.3 virginica
149 5.9 3.0 5.1 1.8 virginica
Row: Sample
No of columns: n_features
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
2: Favorites
7: Structure
8:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
in_class_coding.py
ITP449_Fall2020
Class
.idea
Files
In Class Codi
in_class_c
1 import seaborn as sns
2
3 iris = sns.load_dataset('iris')
4 print(iris)
6 156
Dependent variable: Target array y 1D with length n samples contained in array or Series
Run: in_class_coding
sepal_length sepal_width petal_length petal_width species
0 5.1 3.5 1.4 0.2 setosa
1 4.9 3.0 1.4 0.2 setosa
2 4.7 3.2 1.3 0.2 setosa
3 4.6 3.1 1.5 0.2 setosa
4 5.0 3.6 1.4 0.2 setosa
.. ..
145 6.7 3.0 5.2 2.3 virginica
146 6.3 2.5 5.0 1.9 virginica
147 6.5 3.0 5.2 2.0 virginica
148 6.2 3.4 5.4 2.3 virginica
149 5.9 3.0 5.1 1.8 virginica
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
8:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

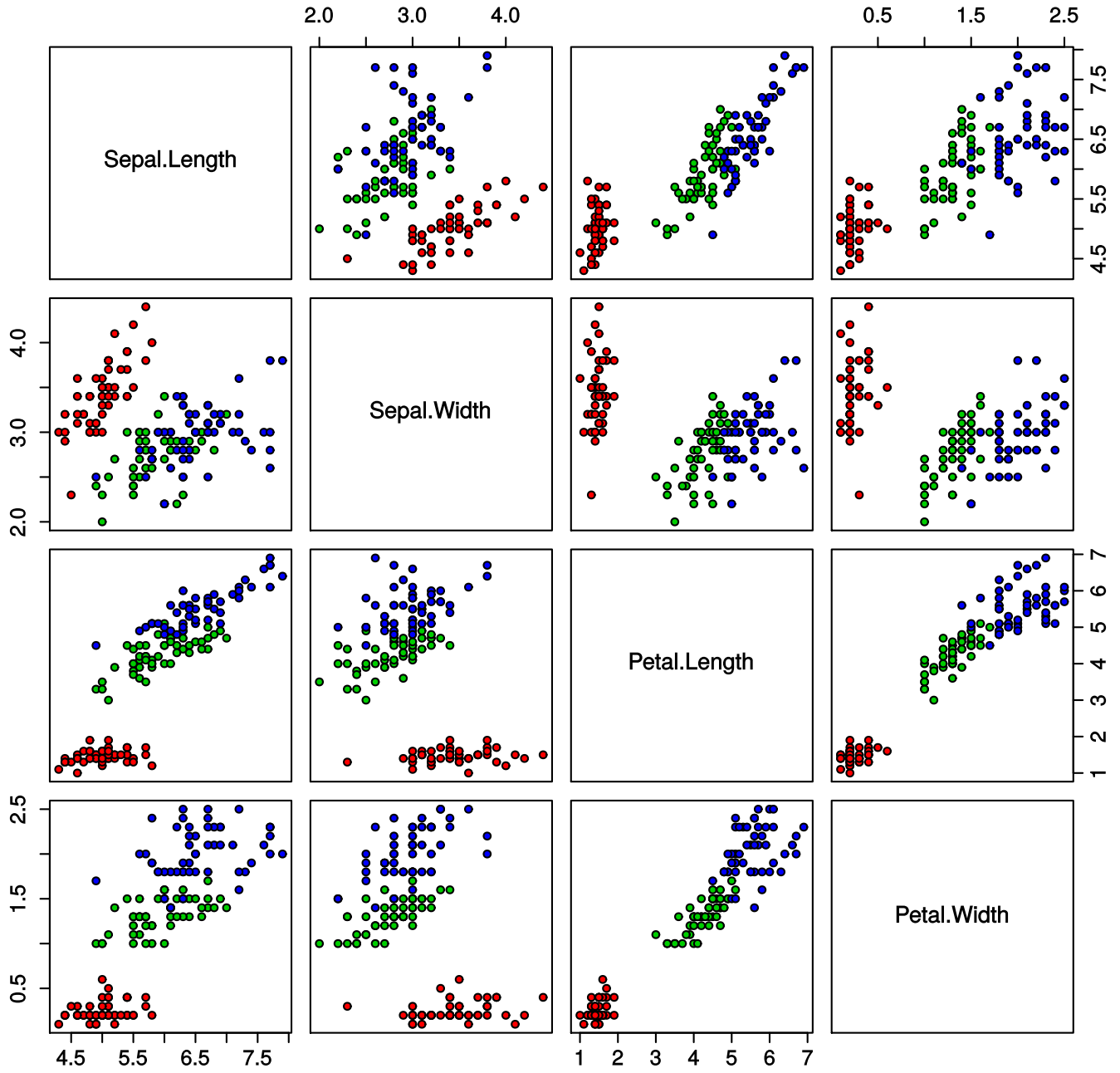



```
In[3]: X_iris = iris.drop('species', axis=1)
       X_iris.shape
```

```
Out[3]: (150, 4)
```

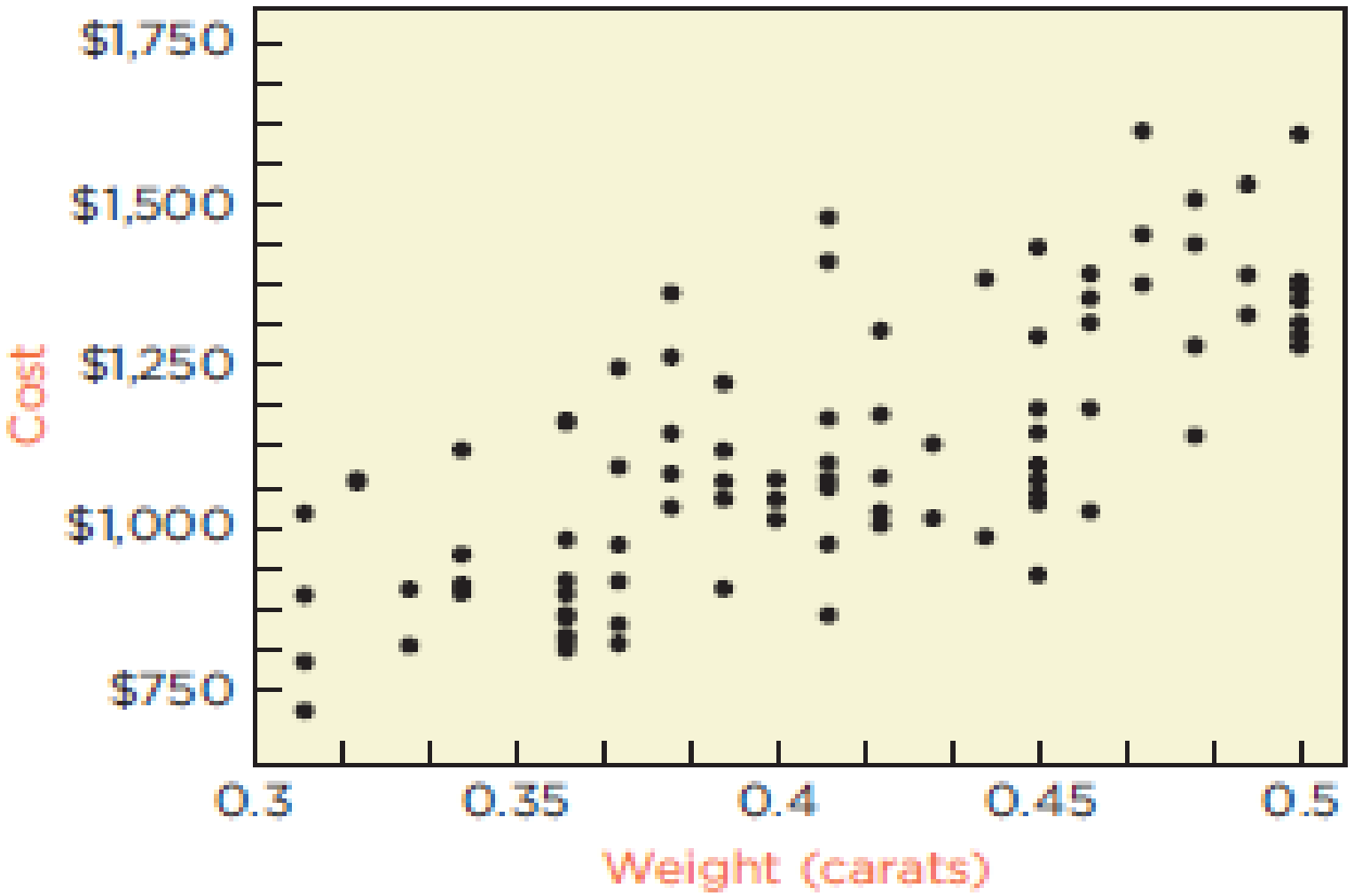
```
In[4]: y_iris = iris['species']
       y_iris.shape
```

```
Out[4]: (150,)
```


Iris Data (red=setosa,green=versicolor,blue=virginica)
Sepal.Length
Sepal.Width
Petal.Length
Petal.Width
2.0
3.0
4.0
0.5
1.5
2.5
4.5
5.5
6.5
7.5
1
2
3
4
5
6
7

Simple Linear Regression example

What is the relationship between the price and weight of diamonds?


$1,750
$1,500
$1,250
$1,000
$750
Cost
0.3
0.35
0.4
0.45
0.5
Weight (carats)

Regression Analysis

Using a sample of diamonds of various weights, regression analysis produces an equation that relates weight to price

Let *y* denote the response variable and let *x* denote the explanatory (or predictor) variable

Equation of a Line

Identify the line fit to the data by an intercept b_0 and a slope b_1

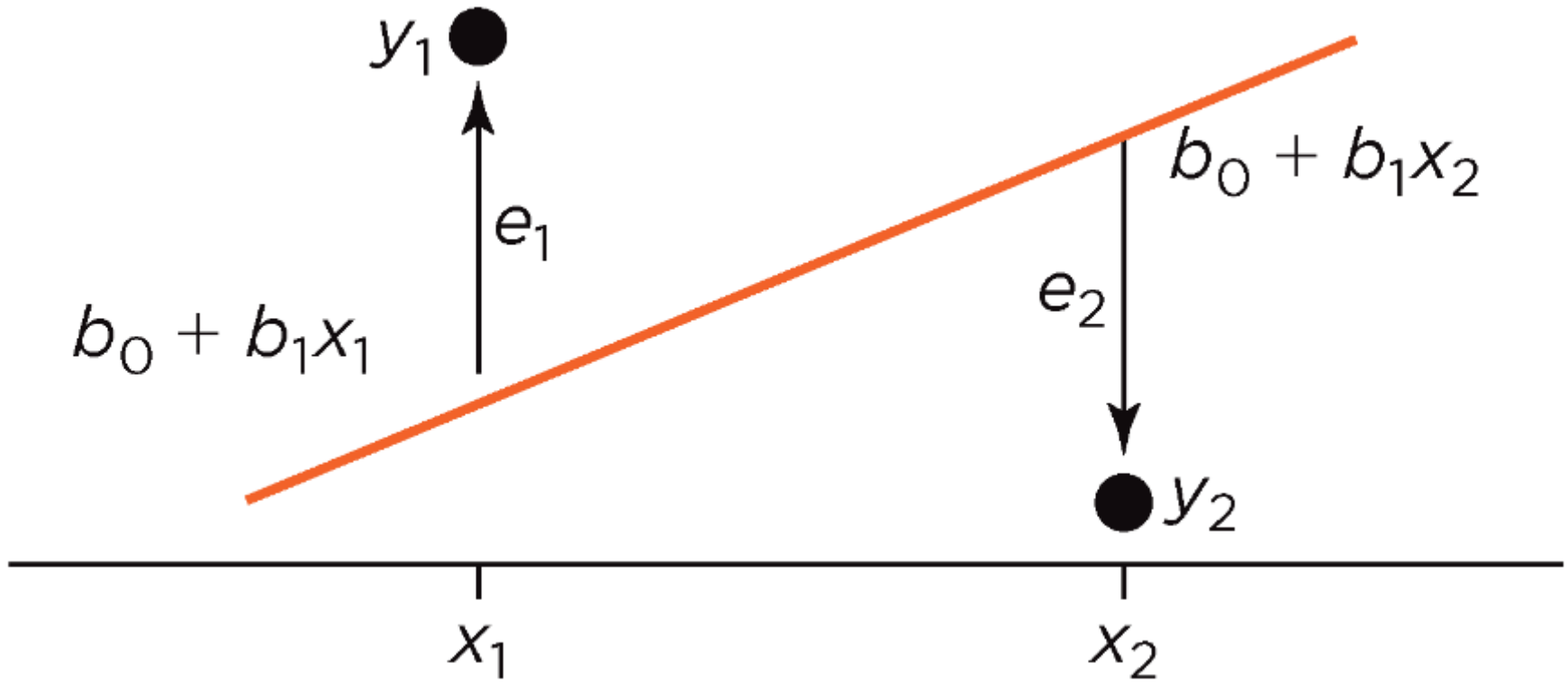
The equation of the line is $\hat{y} = b_0 + b_1 x$

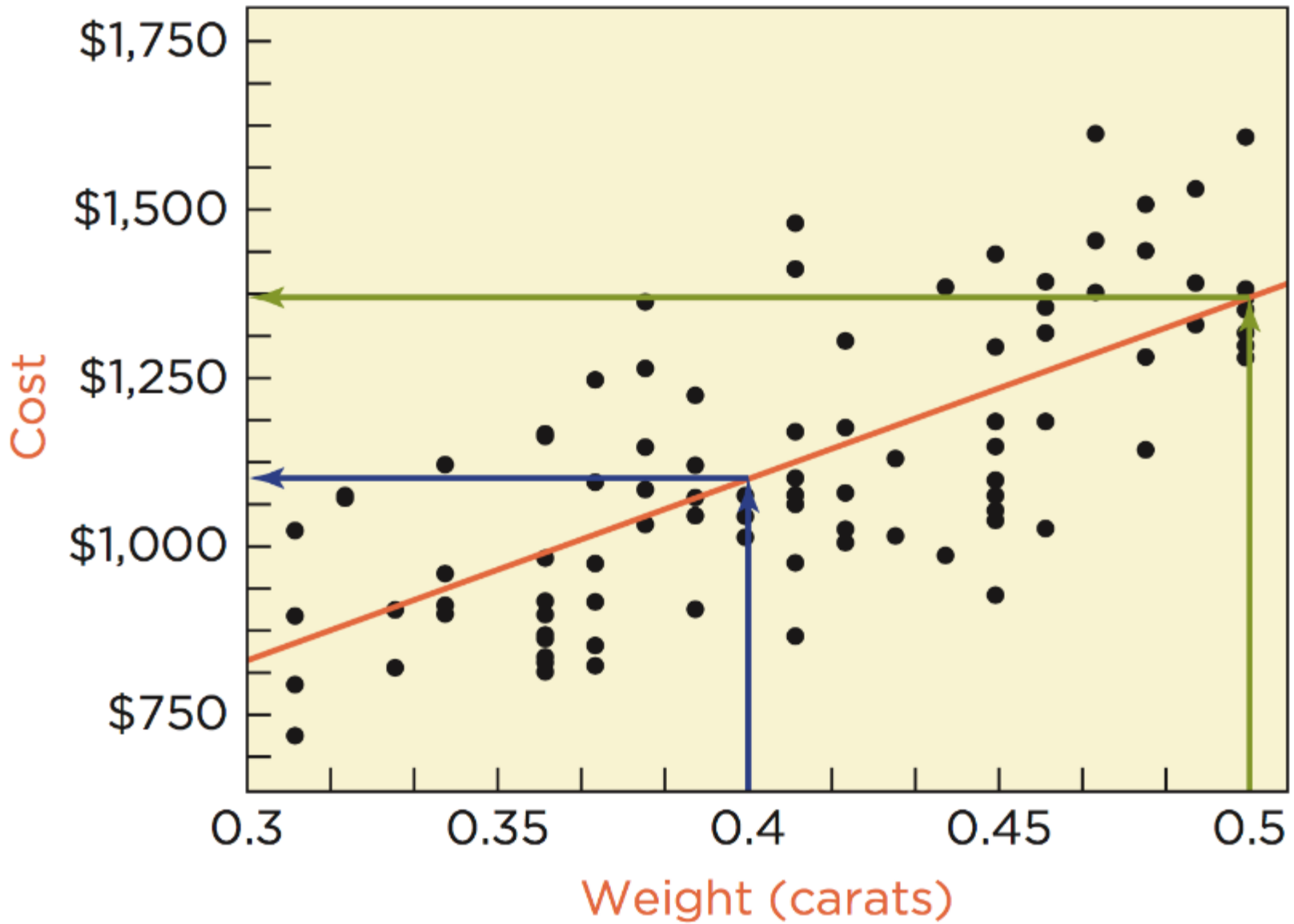
Least Squares

Residual Vertical deviation of a point from the line

The best fitting line collectively makes the squares of the residuals as small as possible

The choices of b_0 and b_1 minimize the sum of the squared residuals


y_1
e_1
$b_0 + b_1x_1$
$b_0 + b_1x_2$
e_2
y_2
x_1
x_2


$1,750
$1,500
$1,250
$1,000
$750
Cost
0.3
0.35
0.4
0.45
0.5
Weight (carats)

Diamond Example

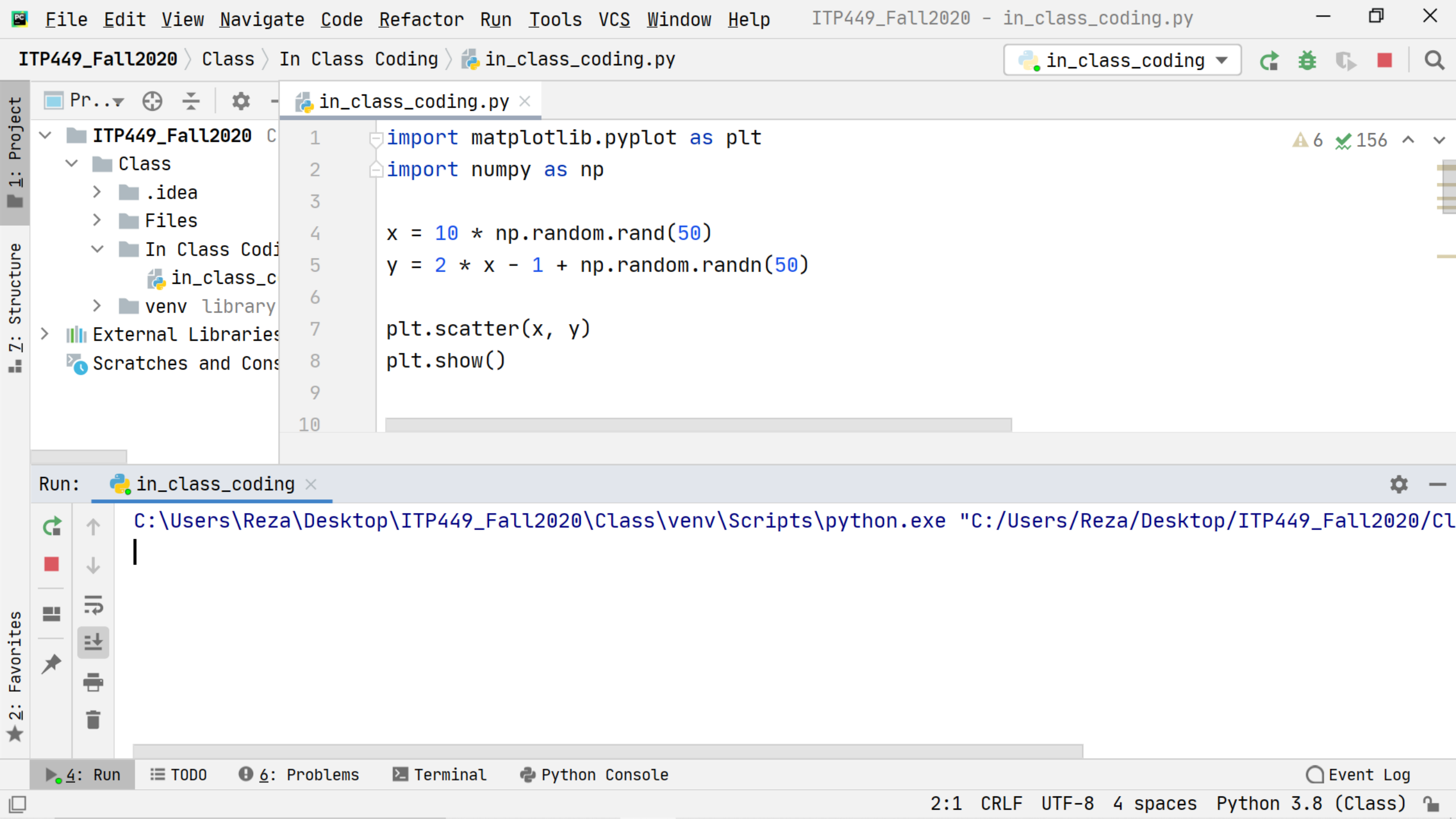
$$\textit{Estimated Price} = 15 + 2{,}697 * \textit{Weight}$$

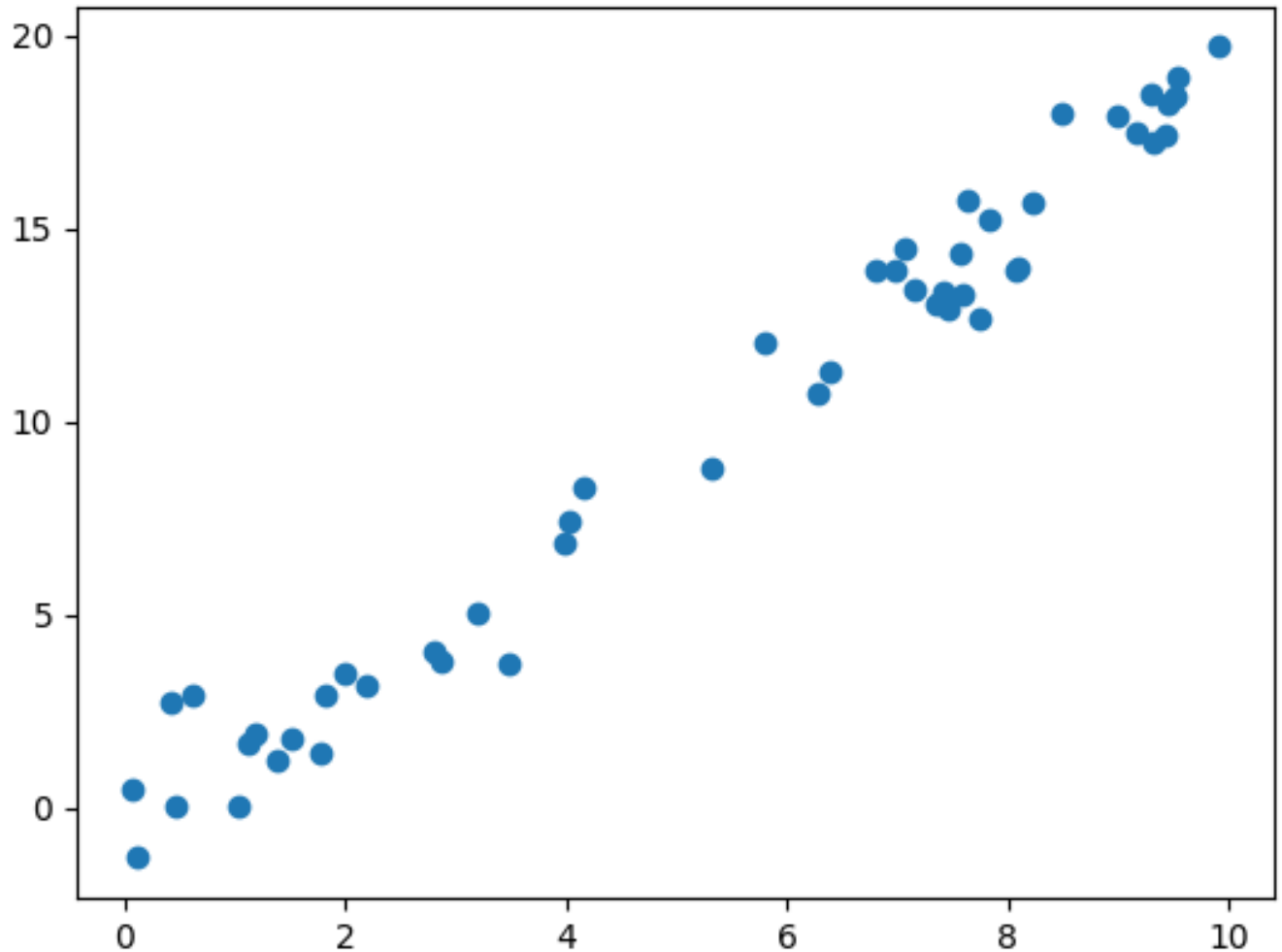
Interpretation

Intercept Average response when x = 0 and where the line crosses the *y* axis

Slope Estimates the marginal cost (additional cost for each additional weight) used to find the total cost

Example


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
Pr...
in_class_coding.py
ITP449_Fall2020 C
Class
.idea
Files
In Class Codi
in_class_c
venv library
External Libraries
Scratches and Cons
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 x = 10 * np.random.rand(50)
5 y = 2 * x - 1 + np.random.randn(50)
6
7 plt.scatter(x, y)
8 plt.show()
9
10
6
156
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Cl
7: Structure
2: Favorites
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
2:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


20
15
10
5
0
0
2
4
6
8
10

1. Choose a class of model

scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, … — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, … — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, … — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction. — Examples

News

On-going development: What's new (Changelog)

Scikit-learn 0.21 will drop support for Python 2.7 and Python 3.4.

December 2018. scikit-learn 0.20.2 is available for download (Changelog).

November 2018. scikit-learn 0.20.1 is available for download (Changelog).

Community

About us See authors and contributing

More Machine Learning Find related projects

Questions? See FAQ and stackoverflow

Mailing list: scikit-learn@python.org

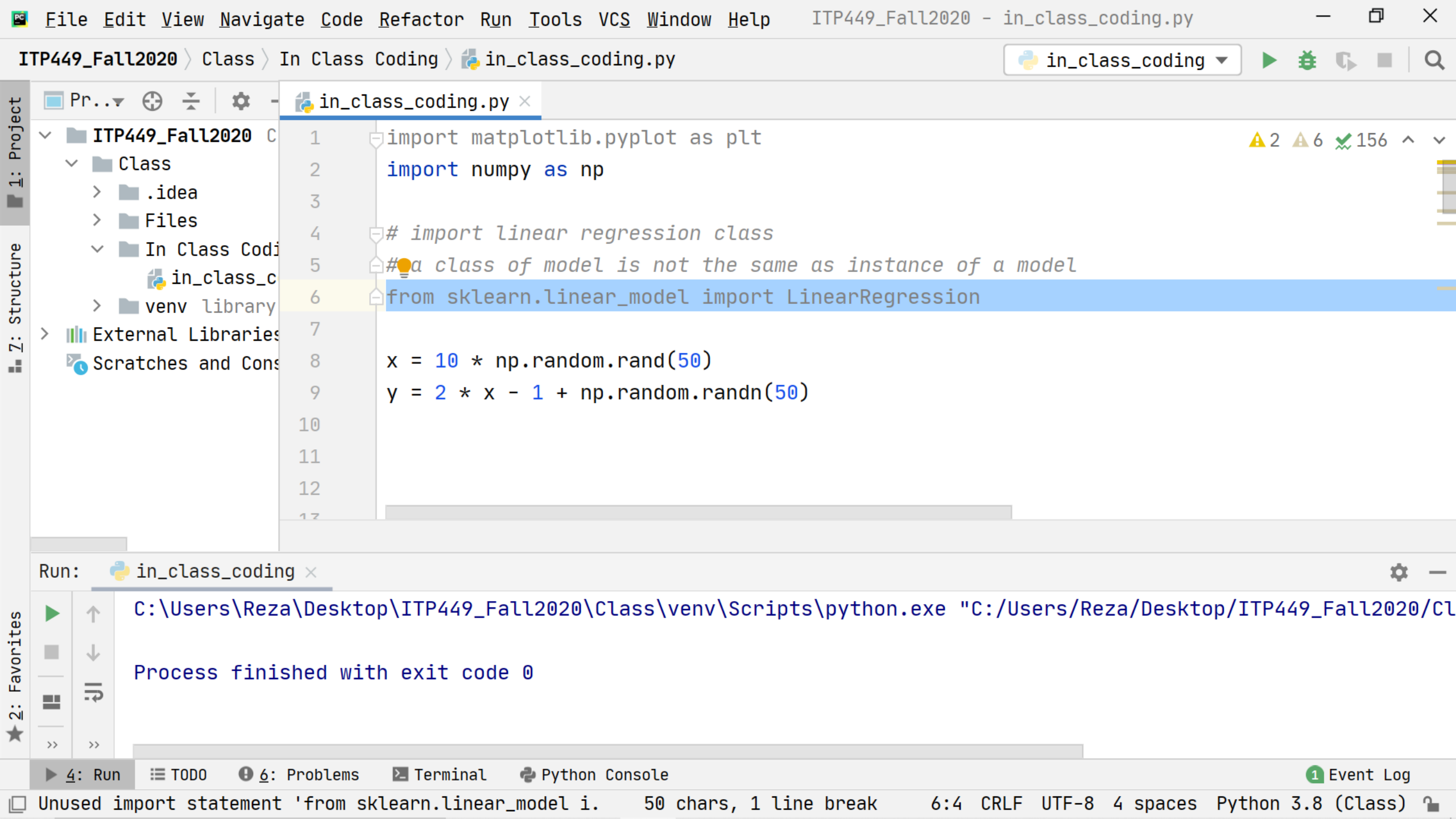
IRC: #scikit-learn @ freenode

Who uses scikit-learn?

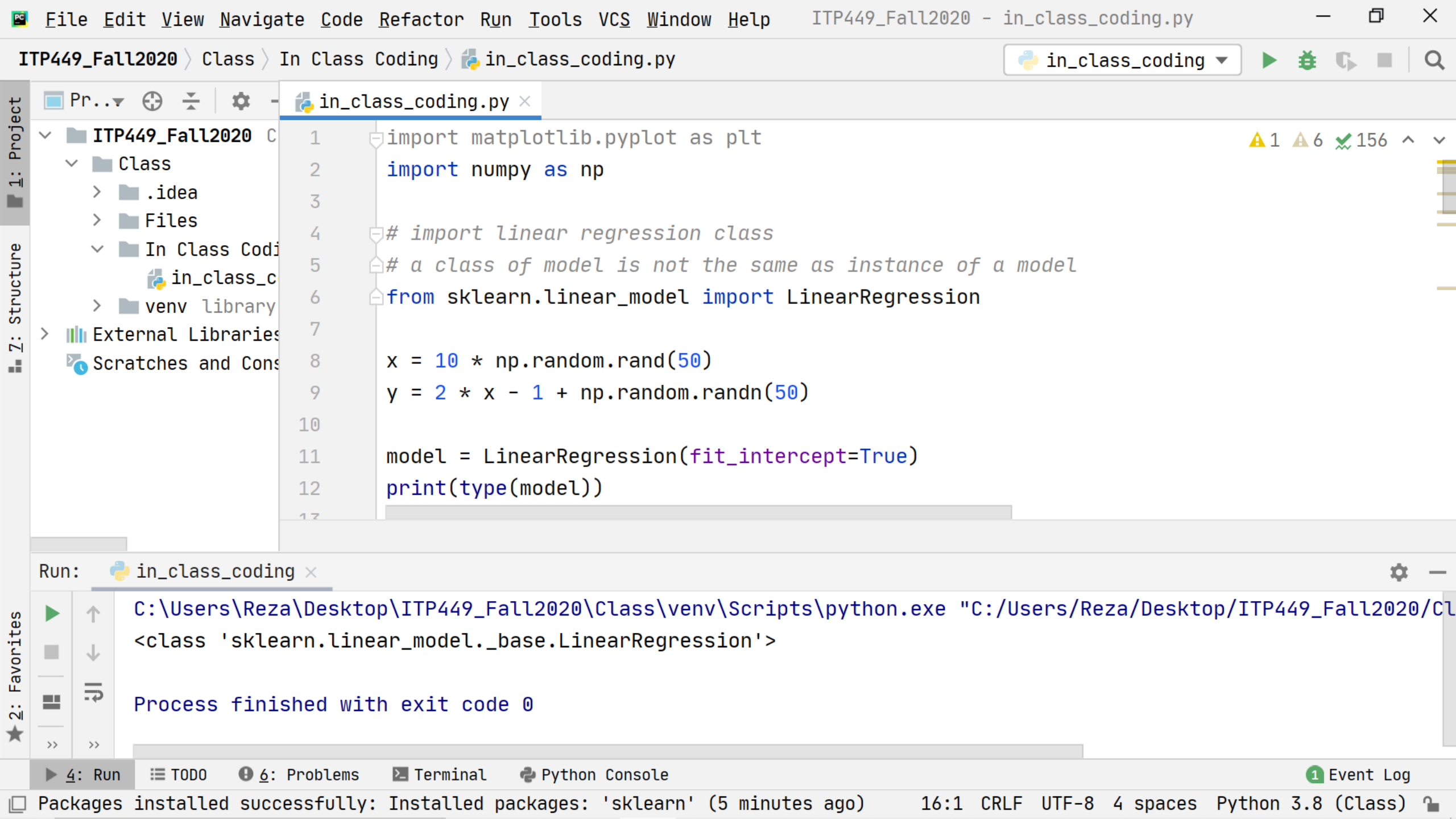
TELECOM ParisTech

…cellent scik-"The great benefit of scikit-learn is its …ing curve […]"

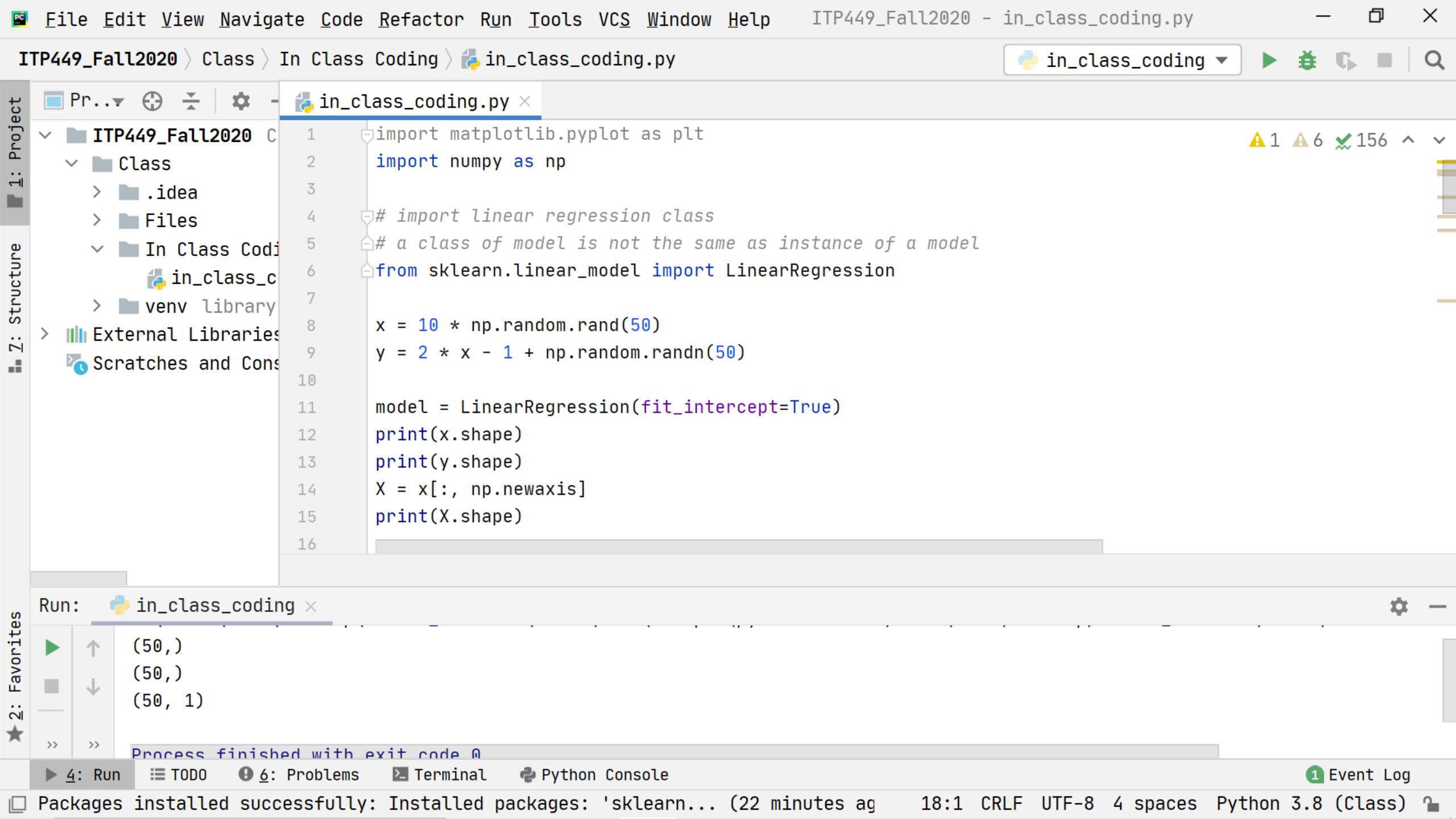
More testimonials

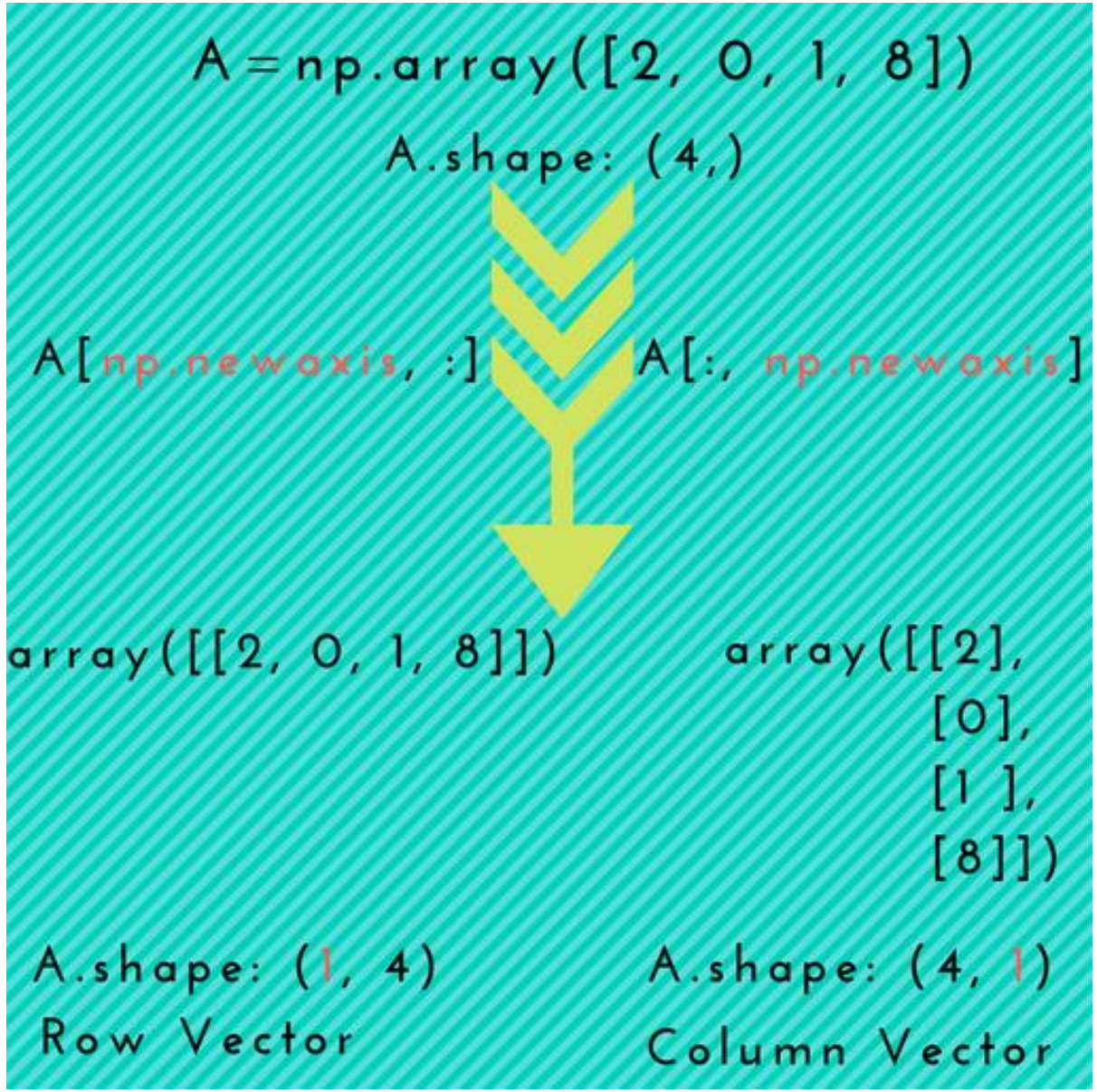

File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
7: Structure
2: Favorites
Pr..
in_class_coding.py
ITP449_Fall2020 C
Class
.idea
Files
In Class Codi
in_class_c
venv library
External Libraries
Scratches and Cons
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # import linear regression class
5 #a class of model is not the same as instance of a model
6 from sklearn.linear_model import LinearRegression
7
8 x = 10 * np.random.rand(50)
9 y = 2 * x - 1 + np.random.randn(50)
10
11
12
2 6 156
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Cl
Process finished with exit code 0
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Unused import statement 'from sklearn.linear_model i.
50 chars, 1 line break
6:4
CRLF
UTF-8
4 spaces
Python 3.8 (Class)

2. Choose model parameters

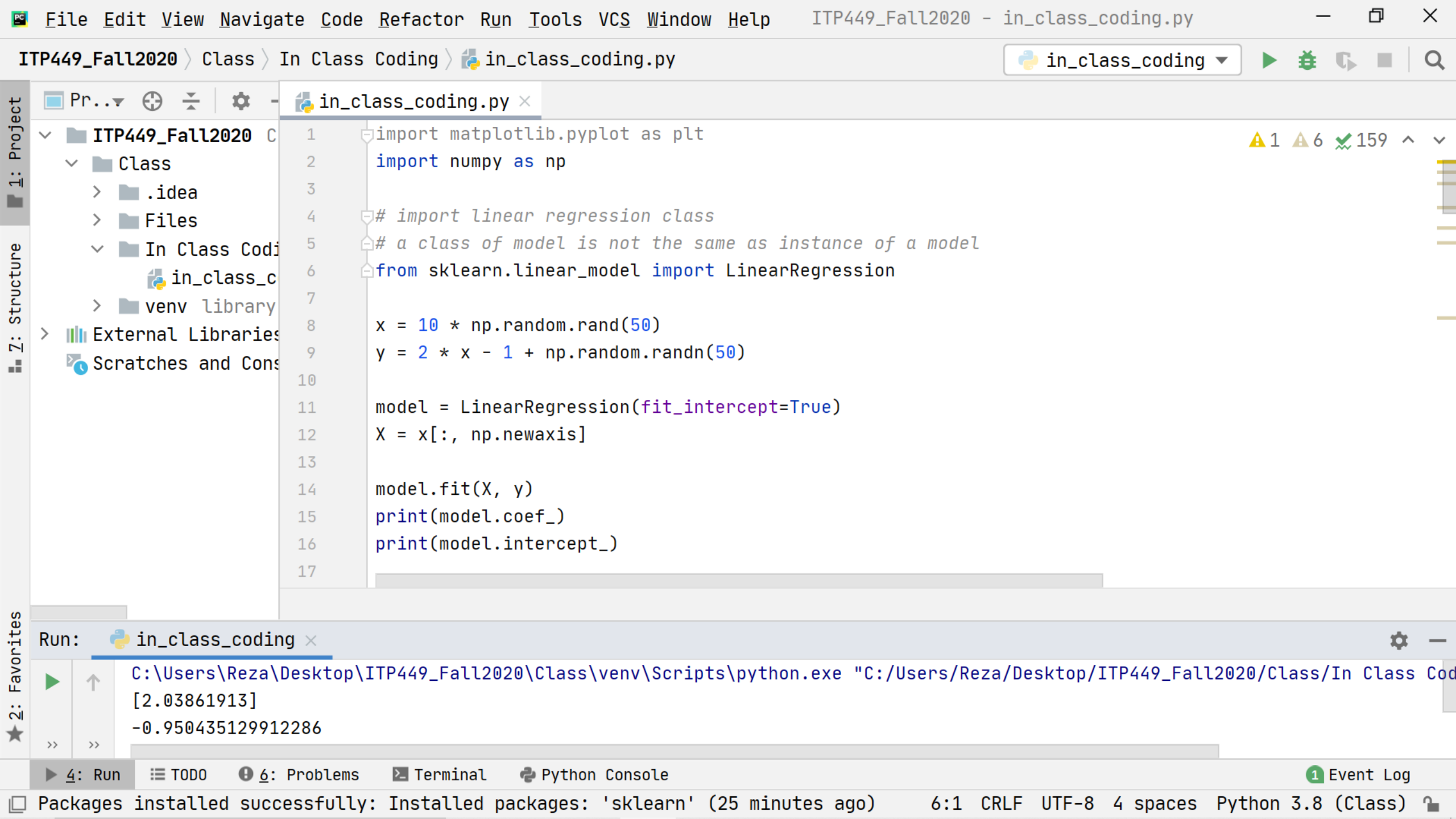

File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
7: Structure
2: Favorites
Pr...
in_class_coding.py
ITP449_Fall2020 C
Class
.idea
Files
In Class Codi
in_class_c
venv library
External Libraries
Scratches and Cons
1 6 156
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # import linear regression class
5 # a class of model is not the same as instance of a model
6 from sklearn.linear_model import LinearRegression
7
8 x = 10 * np.random.rand(50)
9 y = 2 * x - 1 + np.random.randn(50)
10
11 model = LinearRegression(fit_intercept=True)
12 print(type(model))
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Cl
<class 'sklearn.linear_model._base.LinearRegression'>
Process finished with exit code 0
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Packages installed successfully: Installed packages: 'sklearn' (5 minutes ago)
16:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

3. Arrange data


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
7: Structure
2: Favorites
Pr...
in_class_coding.py
ITP449_Fall2020 C
Class
.idea
Files
In Class Codi
in_class_c
venv library
External Libraries
Scratches and Cons
1 6 156
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # import linear regression class
5 # a class of model is not the same as instance of a model
6 from sklearn.linear_model import LinearRegression
7
8 x = 10 * np.random.rand(50)
9 y = 2 * x - 1 + np.random.randn(50)
10
11 model = LinearRegression(fit_intercept=True)
12 print(x.shape)
13 print(y.shape)
14 X = x[:, np.newaxis]
15 print(X.shape)
16
Run: in_class_coding
(50,)
(50,)
(50, 1)
Process finished with exit code 0
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Packages installed successfully: Installed packages: 'sklearn... (22 minutes ag
18:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


A = np.array([2, 0, 1, 8])
A.shape: (4,)
A[np.newaxis, :]
A[:, np.newaxis]
array([[2, 0, 1, 8]])
array([[2],
[0],
[1],
[8]])
A.shape: (1, 4)
Row Vector
A.shape: (4, 1)
Column Vector

4. Fit the model to the data


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
7: Structure
2: Favorites
Pr...
in_class_coding.py
ITP449_Fall2020 C
Class
.idea
Files
In Class Codi
in_class_c
venv library
External Libraries
Scratches and Cons
1 6 159
1 import matplotlib.pyplot as plt
2 import numpy as np
3
4 # import linear regression class
5 # a class of model is not the same as instance of a model
6 from sklearn.linear_model import LinearRegression
7
8 x = 10 * np.random.rand(50)
9 y = 2 * x - 1 + np.random.randn(50)
10
11 model = LinearRegression(fit_intercept=True)
12 X = x[:, np.newaxis]
13
14 model.fit(X, y)
15 print(model.coef_)
16 print(model.intercept_)
17
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Cod
[2.03861913]
-0.950435129912286
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Packages installed successfully: Installed packages: 'sklearn' (25 minutes ago)
6:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

5. Predict labels for unknown data

in_class_coding.py

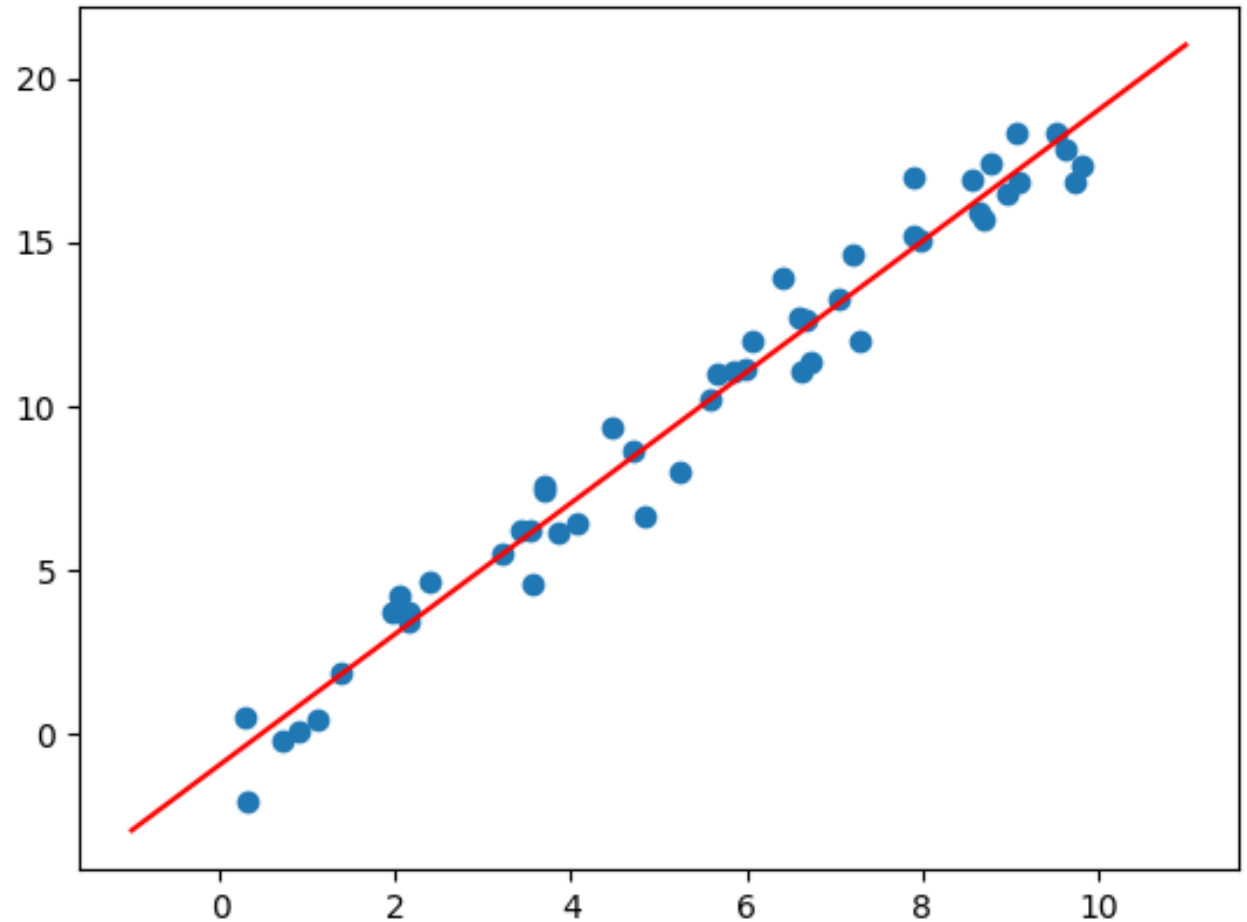
```
1  import numpy as np
2  from sklearn.linear_model import LinearRegression
3
4  x = 10 * np.random.rand(50)
5  y = 2 * x - 1 + np.random.randn(50)
6
7  model = LinearRegression(fit_intercept=True)
8  X = x[:, np.newaxis]
9  model.fit(X, y)
10
11 # Predict
12 xfit = np.linspace(-1, 11, num=50)
13 print(xfit.shape)
14
15 Xfit = xfit[:, np.newaxis]
16 print(Xfit.shape)
17 yfit = model.predict(Xfit)
```

6 162

Run: in_class_coding

```
(50,)
(50, 1)
```

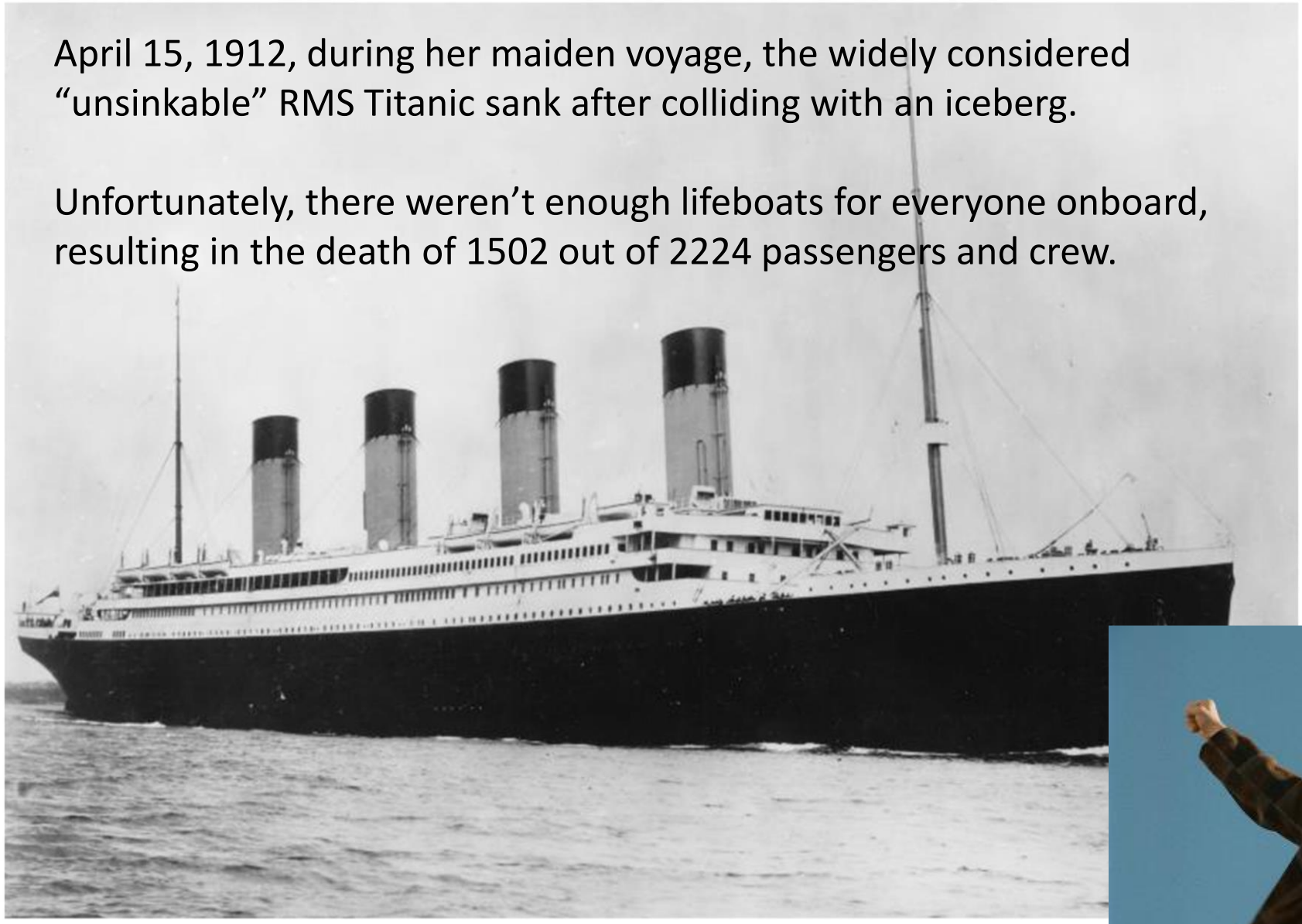
```python
import matplotlib.pyplot as plt
import numpy as np
from sklearn.linear_model import LinearRegression

x = 10 * np.random.rand(50)
y = 2 * x - 1 + np.random.randn(50)

model = LinearRegression(fit_intercept=True)
X = x[:, np.newaxis]
model.fit(X, y)

# Predict
xfit = np.linspace(-1, 11, num=50)
Xfit = xfit[:, np.newaxis]
yfit = model.predict(Xfit)

plt.scatter(x, y)
plt.plot(xfit, yfit, color='r')
plt.show()
```


20
15
10
5
0
0
2
4
6
8
10

Data Wrangling

April 15, 1912, during her maiden voyage, the widely considered “unsinkable” RMS Titanic sank after colliding with an iceberg.

Unfortunately, there weren’t enough lifeboats for everyone onboard, resulting in the death of 1502 out of 2224 passengers and crew.



Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

age: Age is fractional if less than 1. If the age is estimated, it is in the form of xx.5

sibsp: The dataset defines family relations in this way...
Sibling = brother, sister, stepbrother, stepsister
Spouse = husband, wife (mistresses and fiancés were ignored)

parch: The dataset defines family relations in this way...
Parent = mother, father
Child = daughter, son, stepdaughter, stepson
Some children travelled only with a nanny, therefore parch=0 for them.

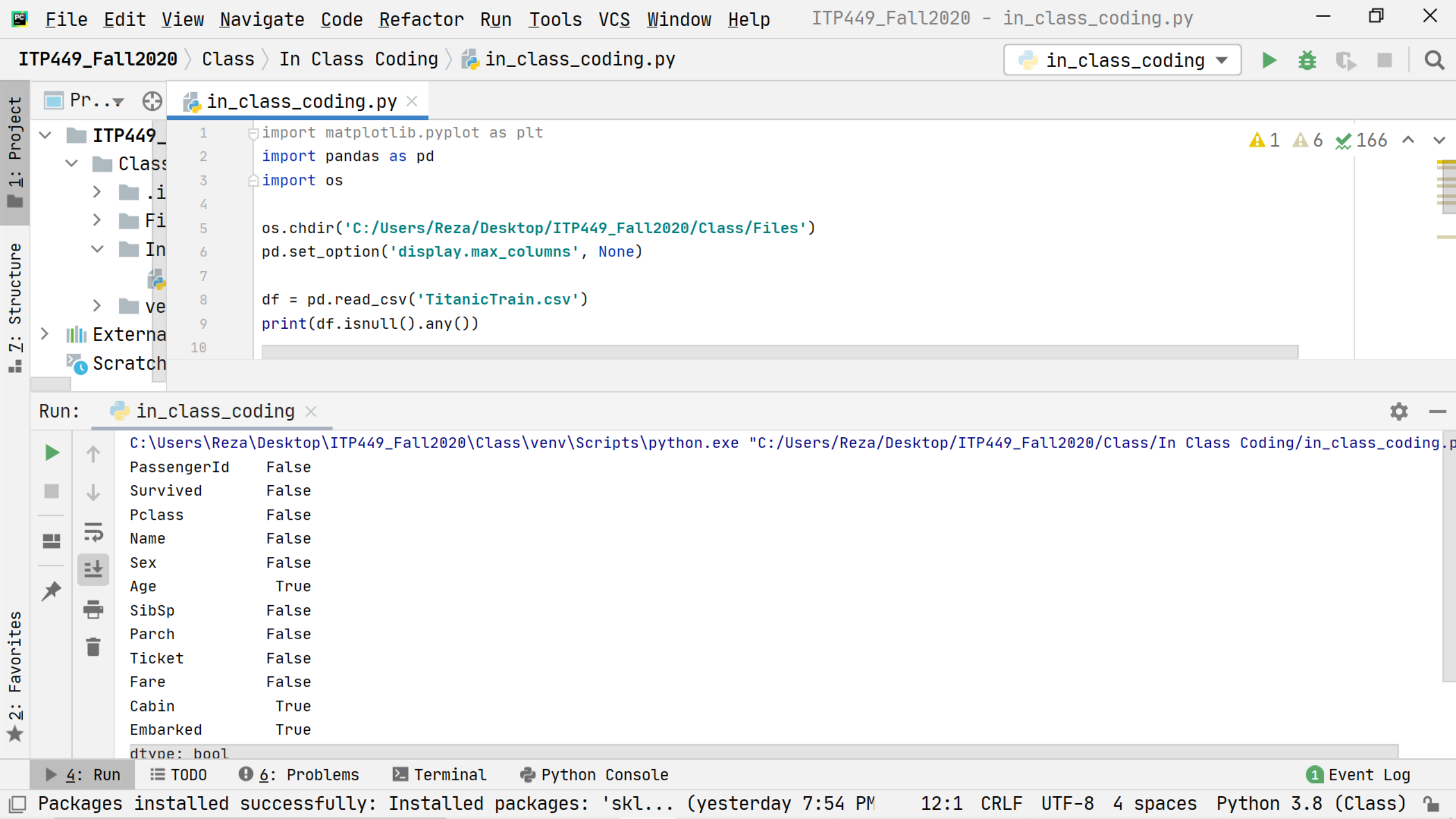
```python
import matplotlib.pyplot as plt
import pandas as pd
import os

os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
pd.set_option('display.max_columns', None)

df = pd.read_csv('TitanicTrain.csv')
print(df.head())
print(df.info())
print(df.describe())
```

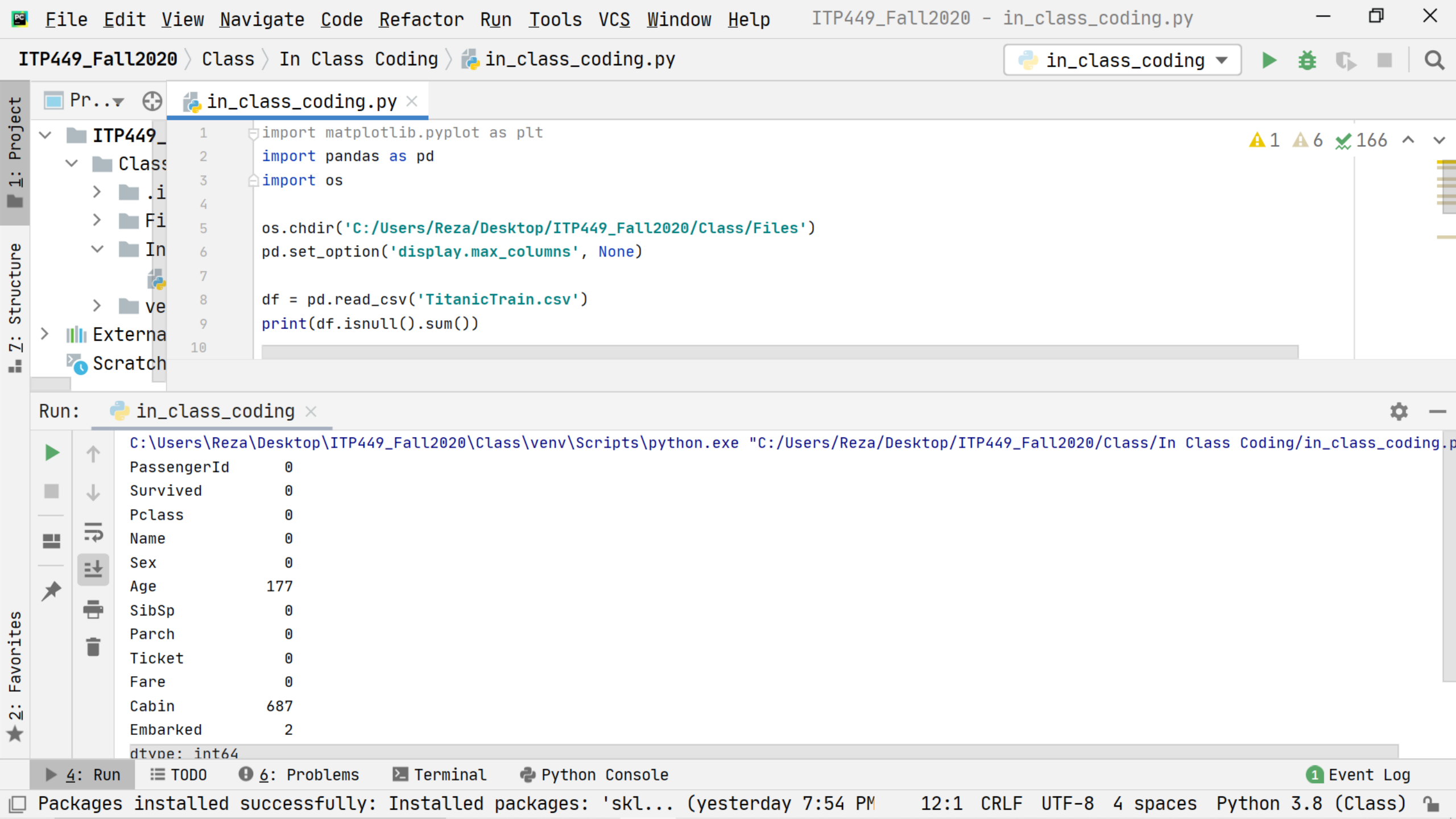

Run: in_class_coding

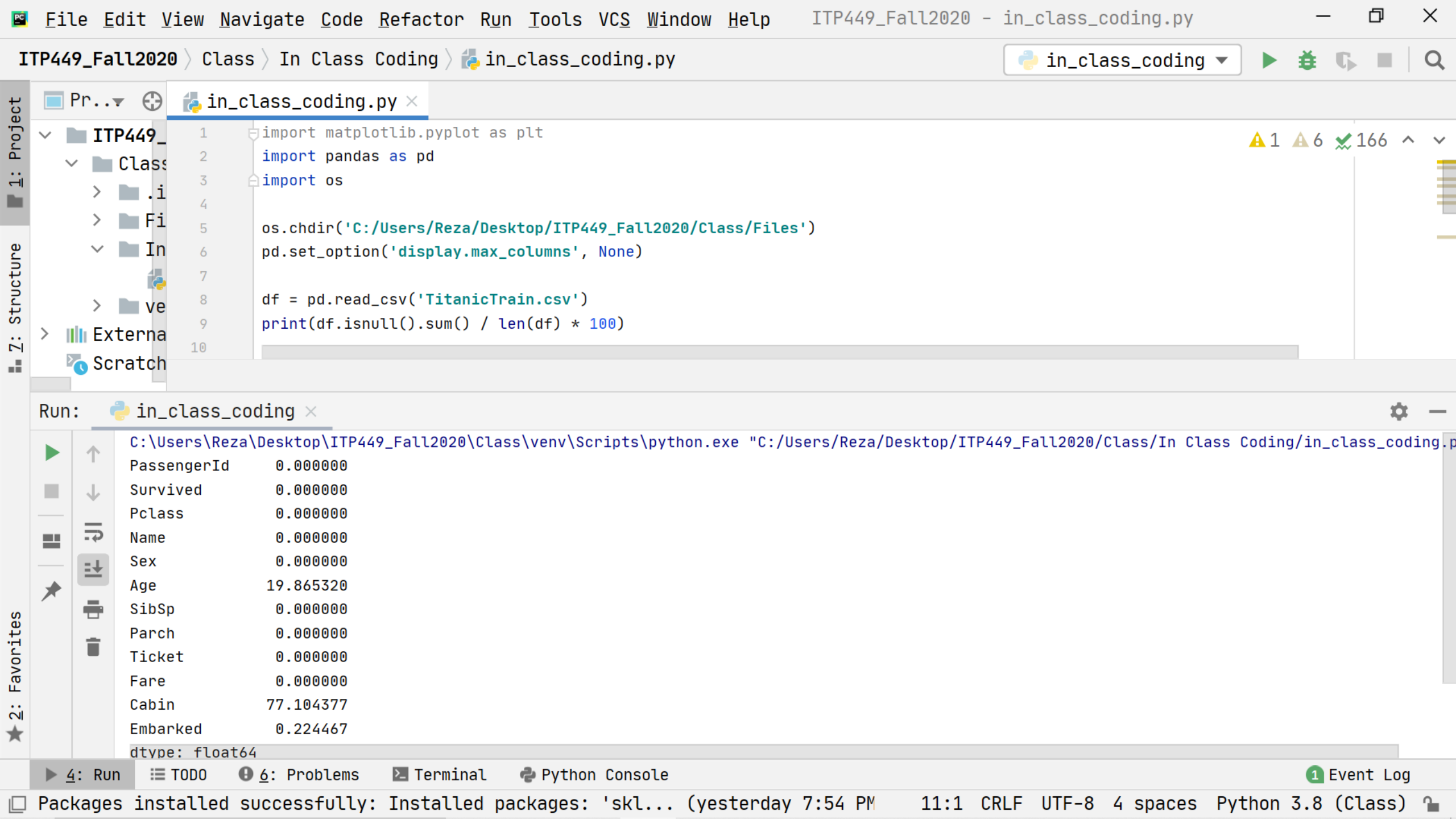
```
             Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

Data Preparation
Missing values

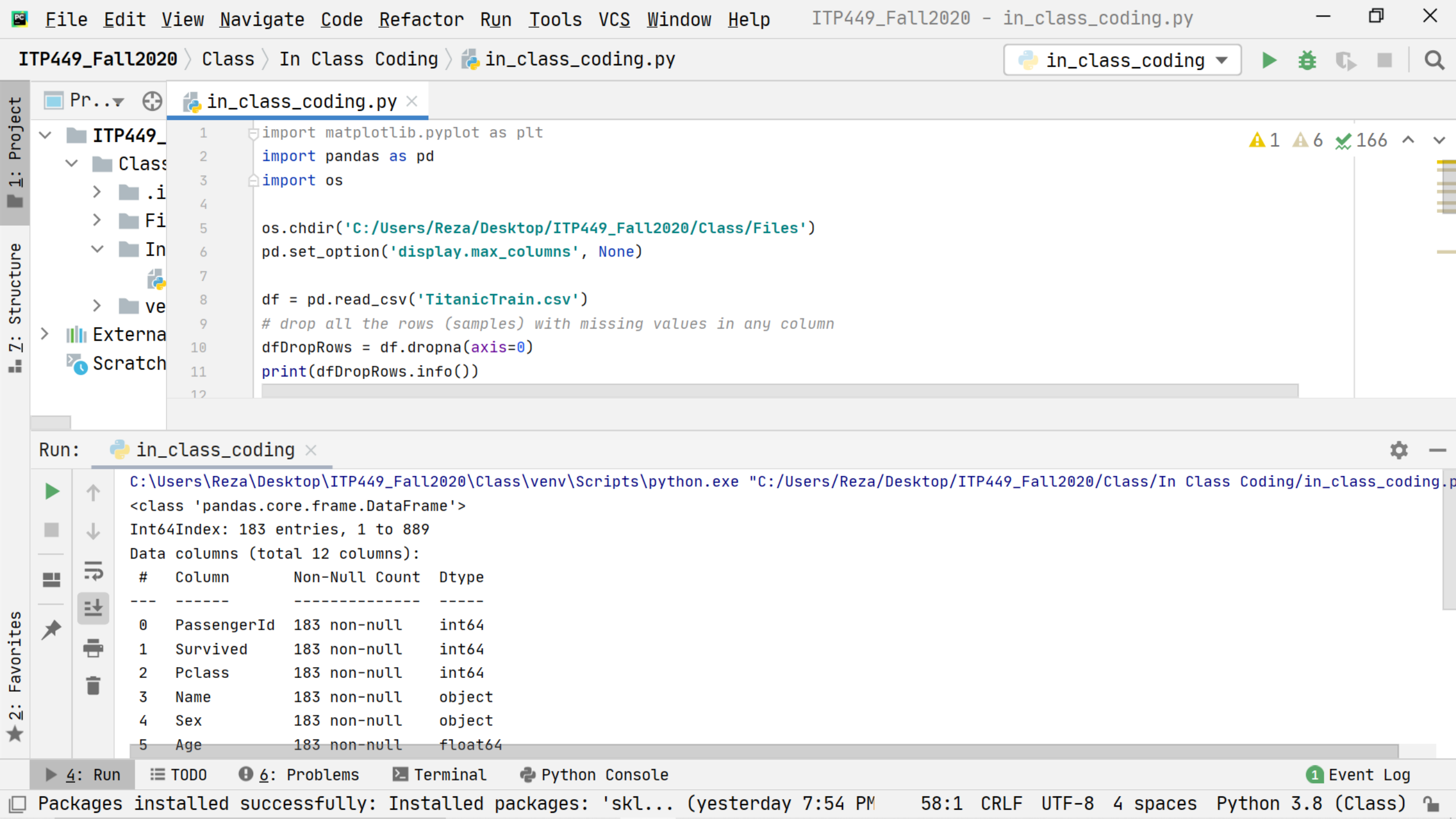
Drop or replace with metric (mean)
Can also use ML to predict missing vals

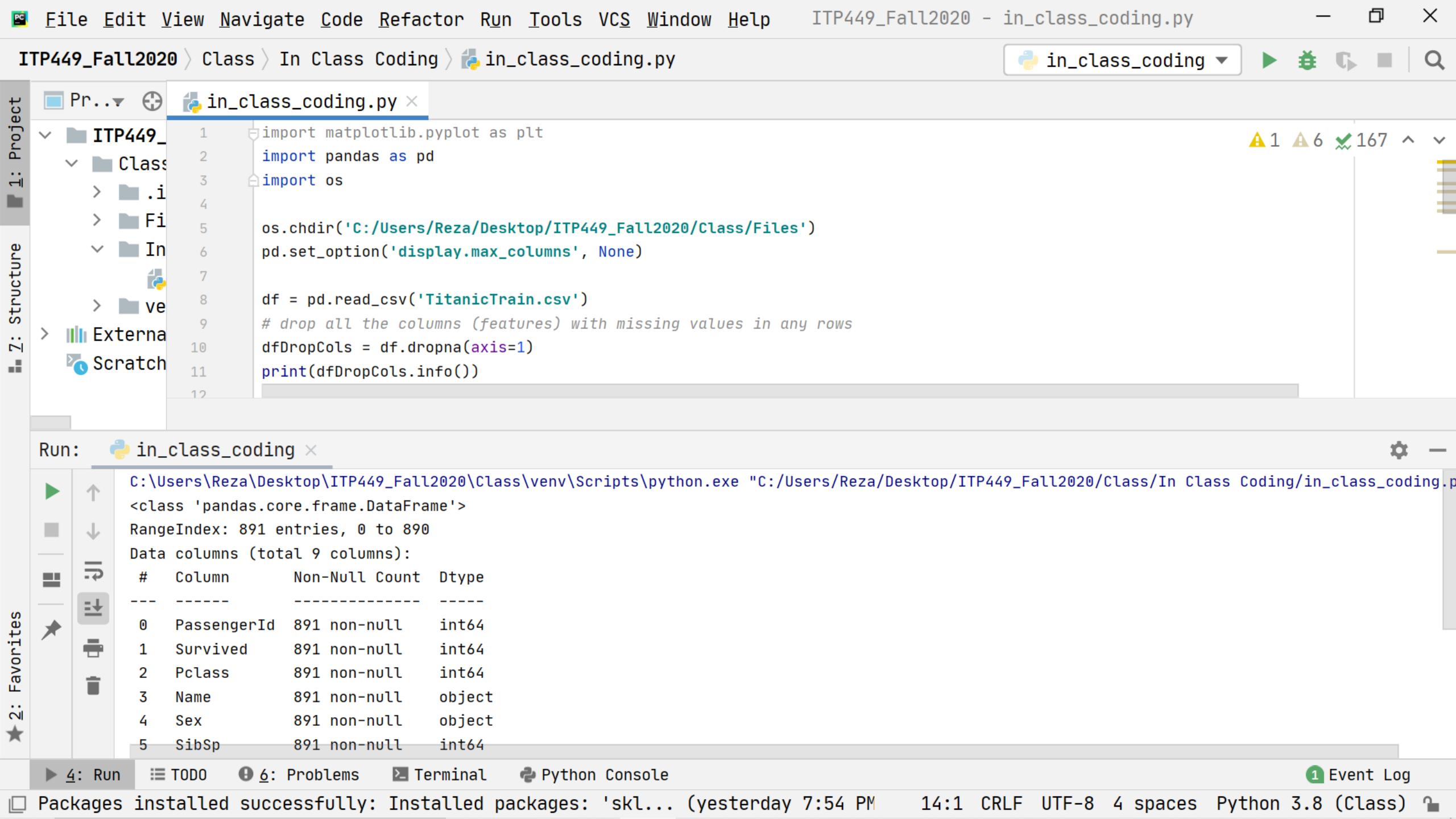

File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
in_class_coding.py
1: Project
7: Structure
2: Favorites
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import os
4
5 os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
6 pd.set_option('display.max_columns', None)
7
8 df = pd.read_csv('TitanicTrain.csv')
9 print(df.isnull().any())
10
1 6 166
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Coding/in_class_coding.p
PassengerId False
Survived False
Pclass False
Name False
Sex False
Age True
SibSp False
Parch False
Ticket False
Fare False
Cabin True
Embarked True
dtype: bool
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Packages installed successfully: Installed packages: 'skl... (yesterday 7:54 PM
12:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

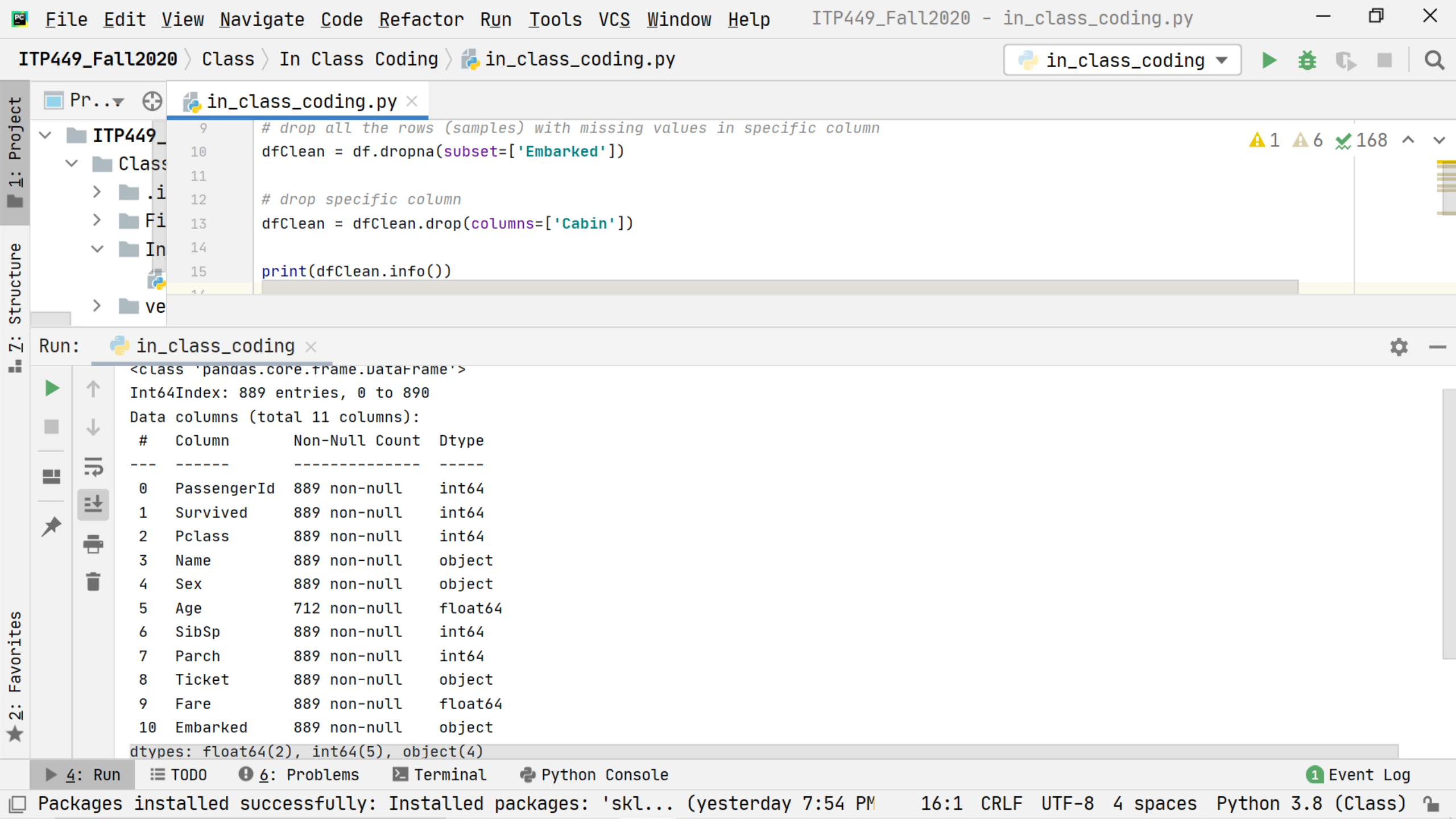

File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
7: Structure
2: Favorites
in_class_coding.py
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import os
4
5 os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
6 pd.set_option('display.max_columns', None)
7
8 df = pd.read_csv('TitanicTrain.csv')
9 print(df.isnull().sum())
10
1 6 166
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Coding/in_class_coding.p
PassengerId 0
Survived 0
Pclass 0
Name 0
Sex 0
Age 177
SibSp 0
Parch 0
Ticket 0
Fare 0
Cabin 687
Embarked 2
dtype: int64
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Packages installed successfully: Installed packages: 'skl... (yesterday 7:54 PM
12:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

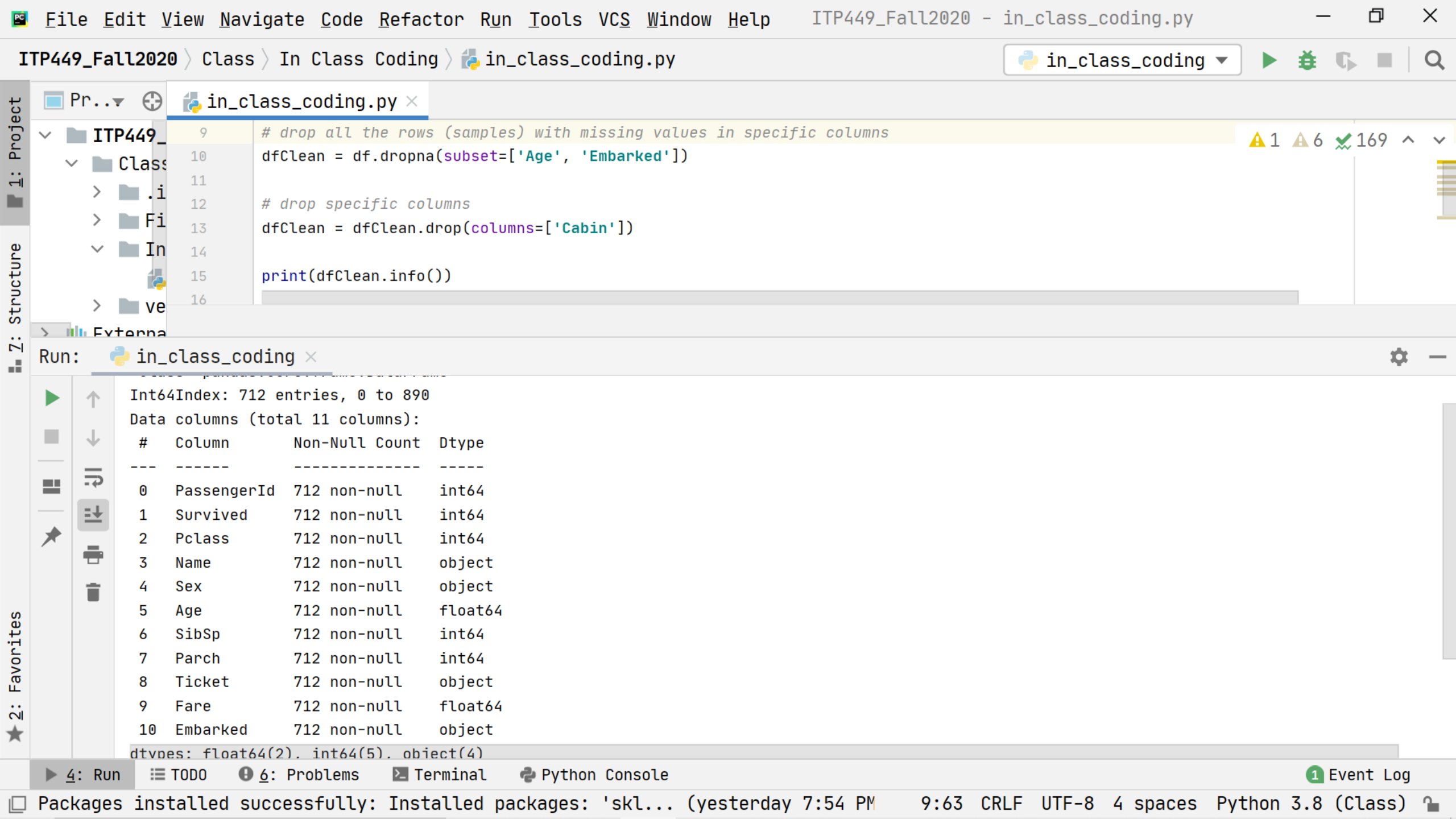

File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
7: Structure
2: Favorites
in_class_coding.py
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import os
4
5 os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
6 pd.set_option('display.max_columns', None)
7
8 df = pd.read_csv('TitanicTrain.csv')
9 print(df.isnull().sum() / len(df) * 100)
10
1 6 166
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Coding/in_class_coding.p
PassengerId 0.000000
Survived 0.000000
Pclass 0.000000
Name 0.000000
Sex 0.000000
Age 19.865320
SibSp 0.000000
Parch 0.000000
Ticket 0.000000
Fare 0.000000
Cabin 77.104377
Embarked 0.224467
dtype: float64
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Packages installed successfully: Installed packages: 'skl... (yesterday 7:54 PM
11:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

Drop it ...


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
7: Structure
2: Favorites
in_class_coding.py
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import os
4
5 os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
6 pd.set_option('display.max_columns', None)
7
8 df = pd.read_csv('TitanicTrain.csv')
9 # drop all the rows (samples) with missing values in any column
10 dfDropRows = df.dropna(axis=0)
11 print(dfDropRows.info())
1 6 166
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Coding/in_class_coding.p
<class 'pandas.core.frame.DataFrame'>
Int64Index: 183 entries, 1 to 889
Data columns (total 12 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 PassengerId 183 non-null int64
1 Survived 183 non-null int64
2 Pclass 183 non-null int64
3 Name 183 non-null object
4 Sex 183 non-null object
5 Age 183 non-null float64
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Packages installed successfully: Installed packages: 'skl... (yesterday 7:54 PM
58:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
1: Project
7: Structure
2: Favorites
in_class_coding.py
ITP449_
Class
.i
Fi
In
ve
Externa
Scratch
1 import matplotlib.pyplot as plt
2 import pandas as pd
3 import os
4
5 os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
6 pd.set_option('display.max_columns', None)
7
8 df = pd.read_csv('TitanicTrain.csv')
9 # drop all the columns (features) with missing values in any rows
10 dfDropCols = df.dropna(axis=1)
11 print(dfDropCols.info())
1 6 167
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Coding/in_class_coding.p
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 9 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 PassengerId 891 non-null int64
1 Survived 891 non-null int64
2 Pclass 891 non-null int64
3 Name 891 non-null object
4 Sex 891 non-null object
5 SibSp 891 non-null int64
4: Run
TODO
6: Problems
Terminal
Python Console
1 Event Log
Packages installed successfully: Installed packages: 'skl... (yesterday 7:54 PM
14:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
in_class_coding.py
# drop all the rows (samples) with missing values in specific column
dfClean = df.dropna(subset=['Embarked'])
# drop specific column
dfClean = dfClean.drop(columns=['Cabin'])
print(dfClean.info())
1 6 168
Run: in_class_coding
<class 'pandas.core.frame.DataFrame'>
Int64Index: 889 entries, 0 to 890
Data columns (total 11 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 PassengerId 889 non-null int64
1 Survived 889 non-null int64
2 Pclass 889 non-null int64
3 Name 889 non-null object
4 Sex 889 non-null object
5 Age 712 non-null float64
6 SibSp 889 non-null int64
7 Parch 889 non-null int64
8 Ticket 889 non-null object
9 Fare 889 non-null float64
10 Embarked 889 non-null object
dtypes: float64(2), int64(5), object(4)
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
Packages installed successfully: Installed packages: 'skl... (yesterday 7:54 PM
16:1 CRLF UTF-8 4 spaces Python 3.8 (Class)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
in_class_coding.py
# drop all the rows (samples) with missing values in specific columns
dfClean = df.dropna(subset=['Age', 'Embarked'])
# drop specific columns
dfClean = dfClean.drop(columns=['Cabin'])
print(dfClean.info())
Run: in_class_coding
Int64Index: 712 entries, 0 to 890
Data columns (total 11 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 PassengerId 712 non-null int64
1 Survived 712 non-null int64
2 Pclass 712 non-null int64
3 Name 712 non-null object
4 Sex 712 non-null object
5 Age 712 non-null float64
6 SibSp 712 non-null int64
7 Parch 712 non-null int64
8 Ticket 712 non-null object
9 Fare 712 non-null float64
10 Embarked 712 non-null object
dtypes: float64(2), int64(5), object(4)
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
Packages installed successfully: Installed packages: 'skl... (yesterday 7:54 PM
9:63 CRLF UTF-8 4 spaces Python 3.8 (Class)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449_Fall2020 - in_class_coding.py

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

in_class_coding

1: Project

in_class_coding.py

```
9   # drop all the rows (samples) with missing values in specific columns
10  dfClean = df.dropna(subset=['Embarked'])
11
12  # drop specific columns
13  dfClean = dfClean.drop(columns=['Age', 'Cabin'])
14
15  print(dfClean.info())
16
```

1 6 170

Run: in_class_coding

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 889 entries, 0 to 890
Data columns (total 10 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  889 non-null    int64
 1   Survived     889 non-null    int64
 2   Pclass       889 non-null    int64
 3   Name         889 non-null    object
 4   Sex          889 non-null    object
 5   SibSp        889 non-null    int64
 6   Parch        889 non-null    int64
 7   Ticket       889 non-null    object
 8   Fare         889 non-null    float64
 9   Embarked     889 non-null    object
dtypes: float64(1), int64(5), object(4)
```

Type here to search

ENG 1:14 AM 10/5/2020

Fill it …

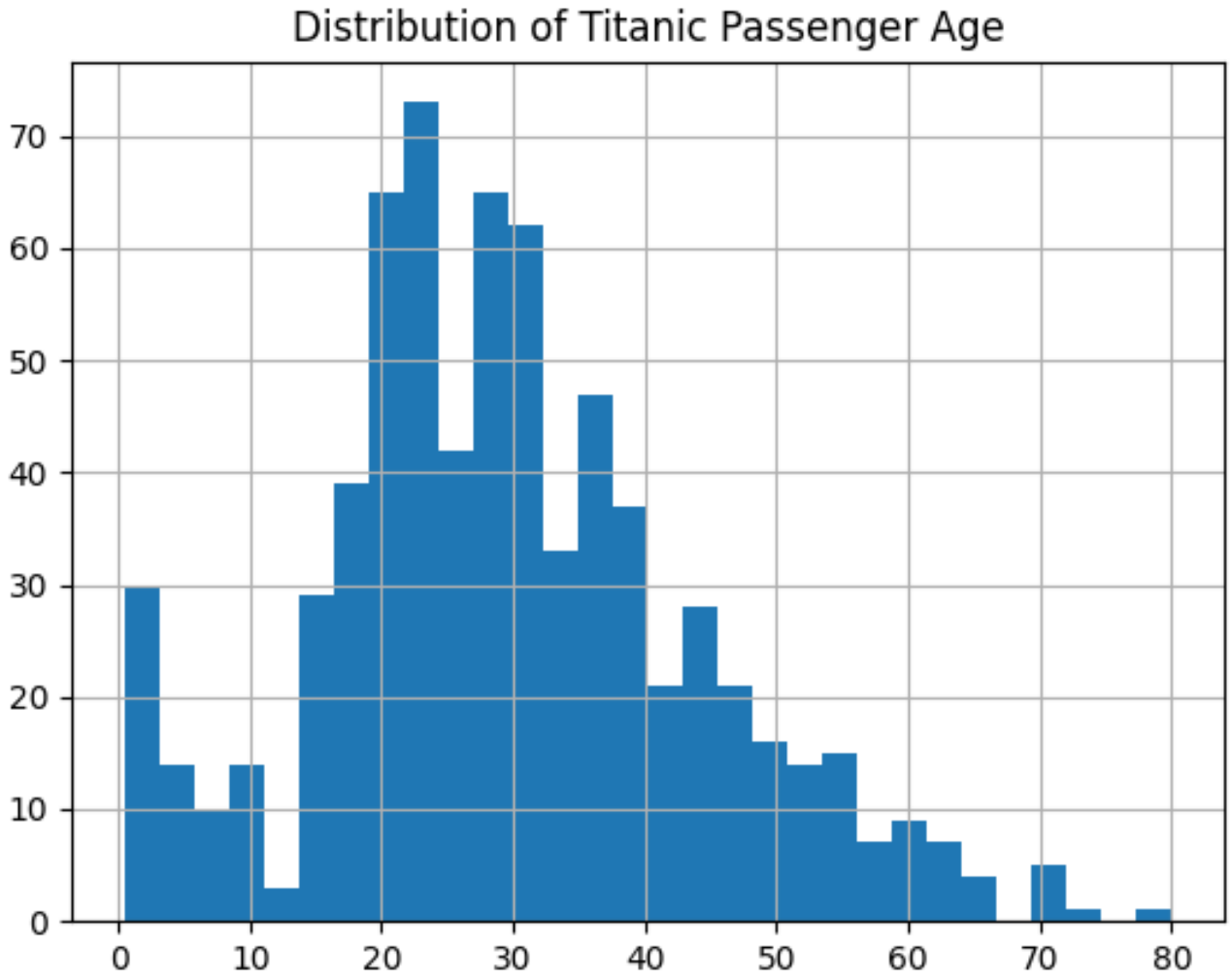
Mean income = $35,000

Median income = $35,000

Mean income = $111,142,222

Median income = $35,000

```python
import matplotlib.pyplot as plt
import pandas as pd
import os

os.chdir('C:/Users/Reza/Desktop/ITP449_Fall2020/Class/Files')
pd.set_option('display.max_columns', None)

df = pd.read_csv('TitanicTrain.csv')
# drop all the rows (samples) with missing values in specific columns
dfClean = df.dropna(subset=['Embarked'])

# drop specific columns
dfClean = dfClean.drop(columns=['Cabin'])

plt.hist(dfClean.loc[~dfClean['Age'].isnull(), 'Age'], bins=30)
plt.title('Distribution of Titanic Passenger Age')
plt.grid()
plt.show()
```

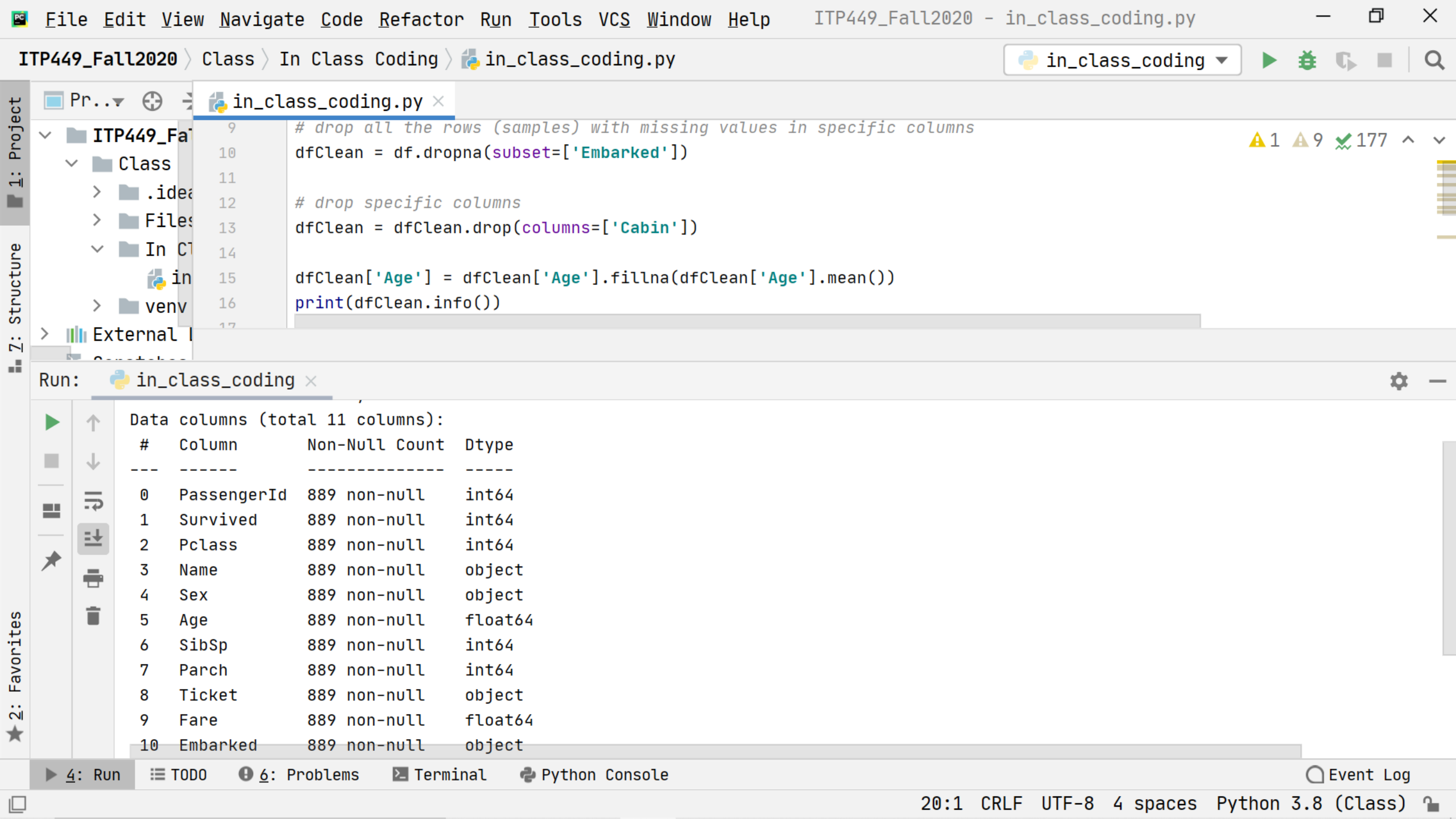

Distribution of Titanic Passenger Age
70
60
50
40
30
20
10
0
0
10
20
30
40
50
60
70
80

```
9   # drop all the rows (samples) with missing values in specific columns
10  dfClean = df.dropna(subset=['Embarked'])
11
12  # drop specific columns
13  dfClean = dfClean.drop(columns=['Cabin'])
14
15  print(dfClean['Age'].describe())
16
```

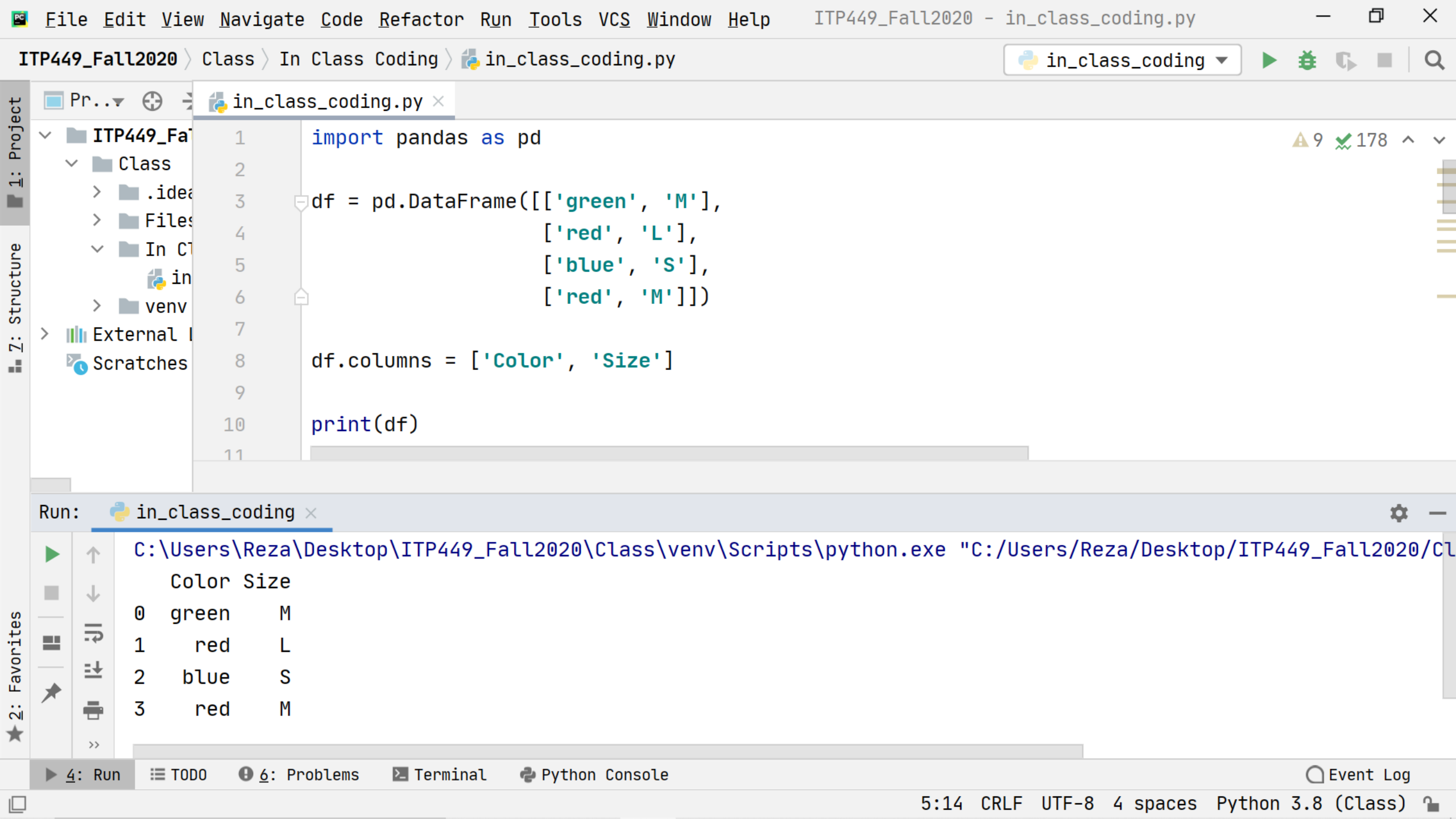
1 9 176

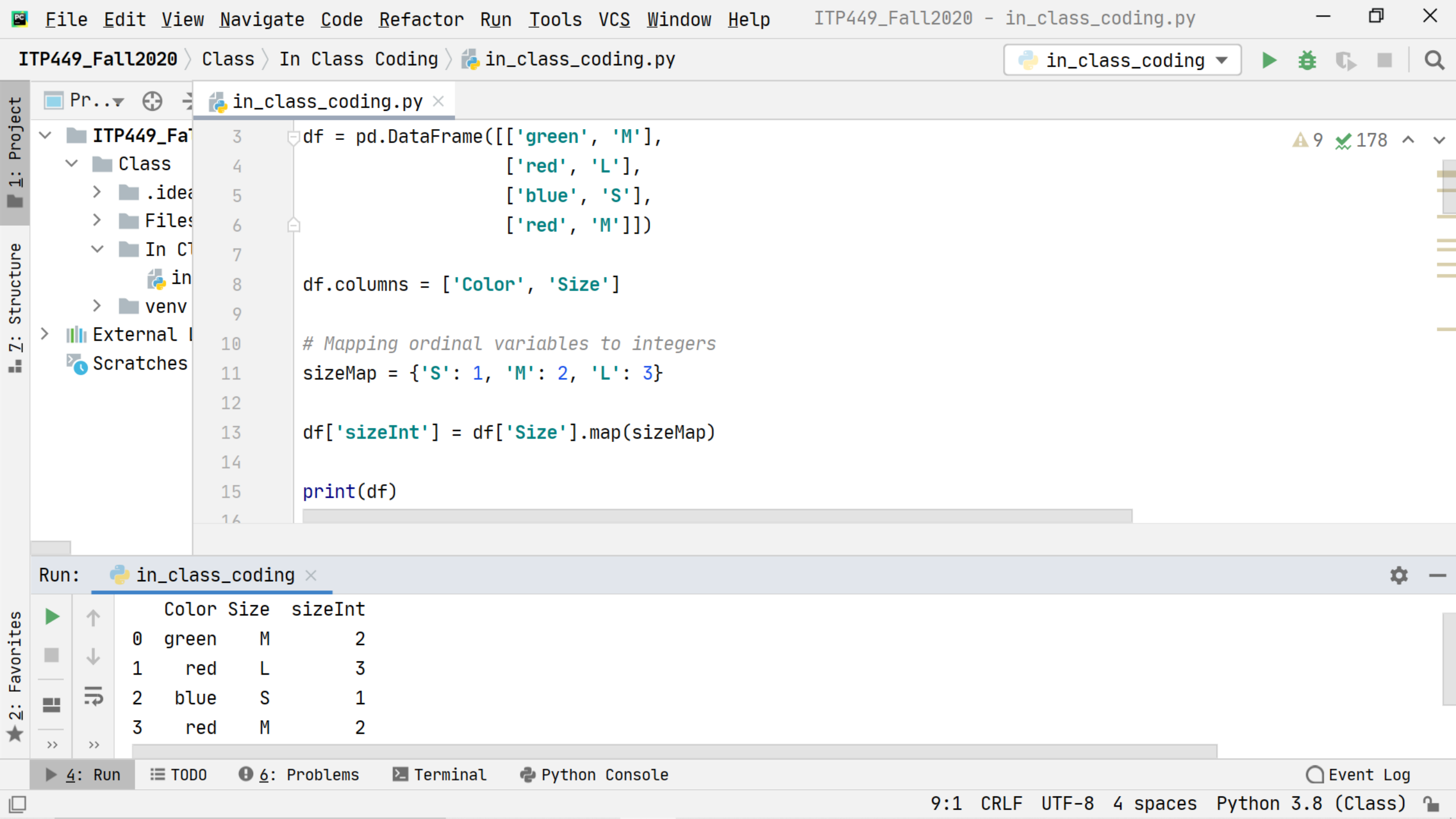
Run: in_class_coding

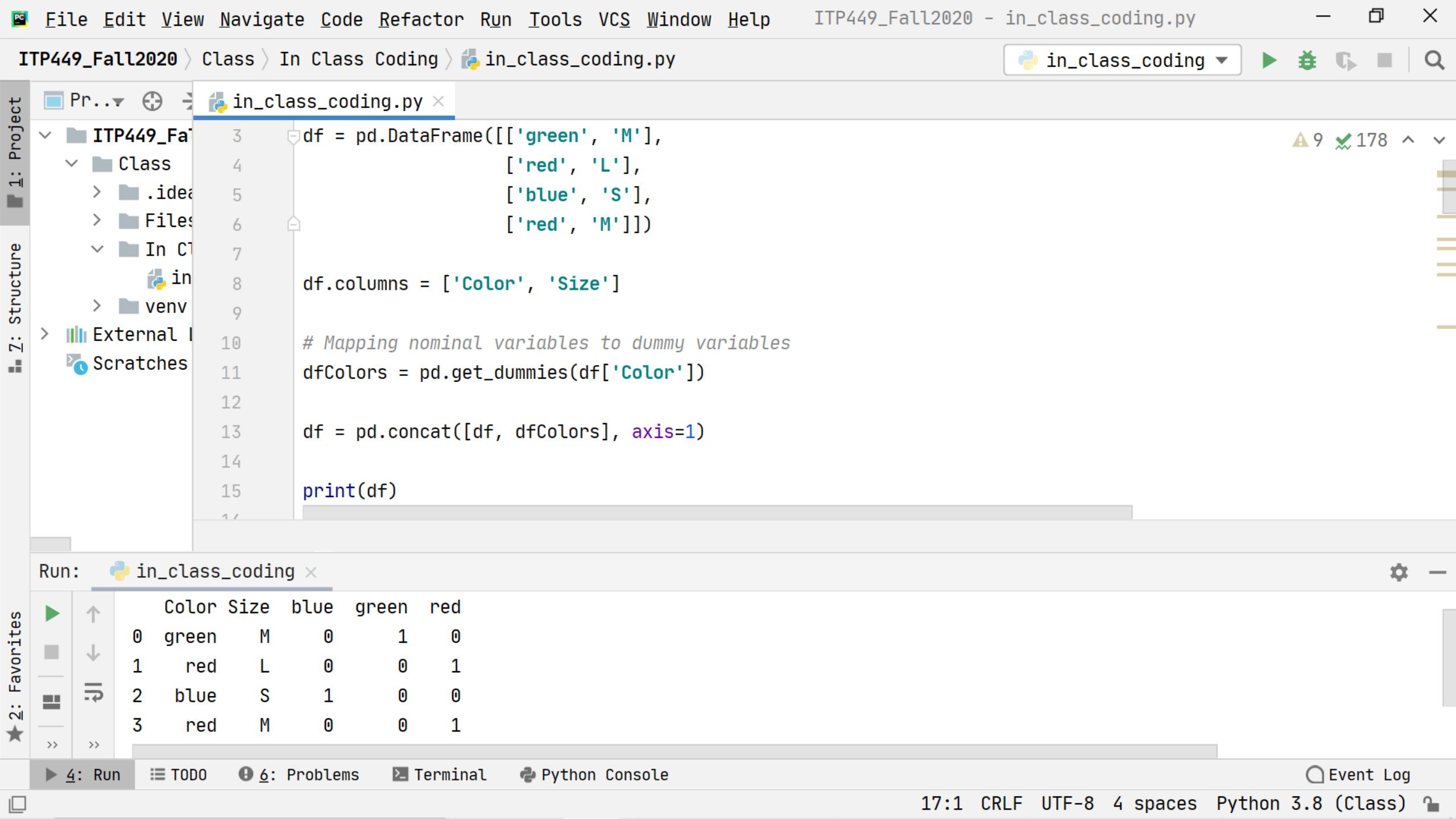
```
count    712.000000
mean      29.642093
std       14.492933
min        0.420000
25%       20.000000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
```

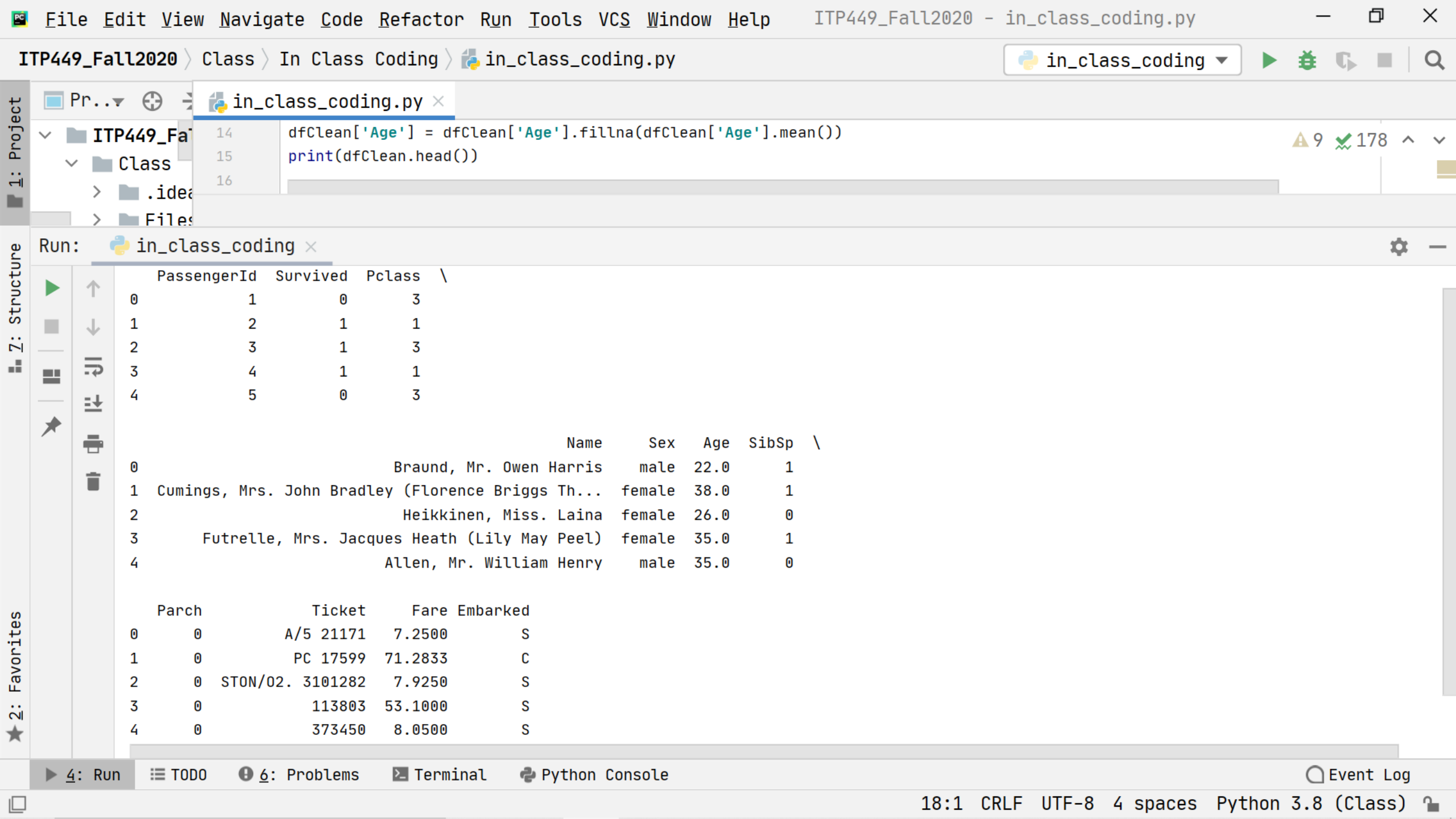

File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
in_class_coding.py
# drop all the rows (samples) with missing values in specific columns
dfClean = df.dropna(subset=['Embarked'])
# drop specific columns
dfClean = dfClean.drop(columns=['Cabin'])
dfClean['Age'] = dfClean['Age'].fillna(dfClean['Age'].mean())
print(dfClean.info())
Run: in_class_coding
Data columns (total 11 columns):
Column Non-Null Count Dtype
--- ------ -------------- -----
0 PassengerId 889 non-null int64
1 Survived 889 non-null int64
2 Pclass 889 non-null int64
3 Name 889 non-null object
4 Sex 889 non-null object
5 Age 889 non-null float64
6 SibSp 889 non-null int64
7 Parch 889 non-null int64
8 Ticket 889 non-null object
9 Fare 889 non-null float64
10 Embarked 889 non-null object
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
20:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

How to handle missing categorical data


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
Pr..
in_class_coding.py
ITP449_Fa
Class
.idea
Files
In Cl
in
venv
External l
Scratches
1: Project
7: Structure
2: Favorites
import pandas as pd
df = pd.DataFrame([['green', 'M'],
['red', 'L'],
['blue', 'S'],
['red', 'M']])
df.columns = ['Color', 'Size']
print(df)
9
178
Run: in_class_coding
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Cl
Color Size
0 green M
1 red L
2 blue S
3 red M
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
5:14 CRLF UTF-8 4 spaces Python 3.8 (Class)


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
in_class_coding.py
ITP449_Fa
Class
.idea
Files
In Cl
in
venv
External l
Scratches
df = pd.DataFrame([['green', 'M'],
['red', 'L'],
['blue', 'S'],
['red', 'M']])
df.columns = ['Color', 'Size']
# Mapping ordinal variables to integers
sizeMap = {'S': 1, 'M': 2, 'L': 3}
df['sizeInt'] = df['Size'].map(sizeMap)
print(df)
9
178
Run: in_class_coding
Color Size sizeInt
0 green M 2
1 red L 3
2 blue S 1
3 red M 2
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
9:1 CRLF UTF-8 4 spaces Python 3.8 (Class)
1: Project
7: Structure
2: Favorites


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
in_class_coding.py
ITP449_Fa
Class
.idea
Files
In Cl
in
venv
External l
Scratches
3 df = pd.DataFrame([['green', 'M'],
4 ['red', 'L'],
5 ['blue', 'S'],
6 ['red', 'M']])
7
8 df.columns = ['Color', 'Size']
9
10 # Mapping nominal variables to dummy variables
11 dfColors = pd.get_dummies(df['Color'])
12
13 df = pd.concat([df, dfColors], axis=1)
14
15 print(df)
9
178
Run: in_class_coding
Color Size blue green red
0 green M 0 1 0
1 red L 0 0 1
2 blue S 1 0 0
3 red M 0 0 1
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
17:1 CRLF UTF-8 4 spaces Python 3.8 (Class)
1: Project
7: Structure
2: Favorites


File Edit View Navigate Code Refactor Run Tools VCS Window Help
ITP449_Fall2020 - in_class_coding.py
ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py
in_class_coding
in_class_coding.py
14 dfClean['Age'] = dfClean['Age'].fillna(dfClean['Age'].mean())
15 print(dfClean.head())
16
9 178
Run: in_class_coding
PassengerId Survived Pclass \
0 1 0 3
1 2 1 1
2 3 1 3
3 4 1 1
4 5 0 3
Name Sex Age SibSp \
0 Braund, Mr. Owen Harris male 22.0 1
1 Cumings, Mrs. John Bradley (Florence Briggs Th... female 38.0 1
2 Heikkinen, Miss. Laina female 26.0 0
3 Futrelle, Mrs. Jacques Heath (Lily May Peel) female 35.0 1
4 Allen, Mr. William Henry male 35.0 0
Parch Ticket Fare Embarked
0 0 A/5 21171 7.2500 S
1 0 PC 17599 71.2833 C
2 0 STON/O2. 3101282 7.9250 S
3 0 113803 53.1000 S
4 0 373450 8.0500 S
4: Run
TODO
6: Problems
Terminal
Python Console
Event Log
18:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

File Edit View Navigate Code Refactor Run Tools VCS Window Help ITP449_Fall2020 - in_class_coding.py

ITP449_Fall2020 〉 Class 〉 In Class Coding 〉 in_class_coding.py

in_class_coding

in_class_coding.py

```
14  dfClean['Age'] = dfClean['Age'].fillna(dfClean['Age'].mean())
15  dfSex = pd.get_dummies(dfClean['Sex'])
16  dfClean = pd.concat([dfClean, dfSex], axis=1)
17  print(dfClean.head())
```

9 178

Run: in_class_coding

```
                                                Name     Sex   Age  SibSp  \
0                            Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th...  female  38.0      1
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Embarked  female  male
0      0         A/5 21171   7.2500        S       0     1
1      0          PC 17599  71.2833        C       1     0
2      0  STON/O2. 3101282   7.9250        S       1     0
3      0            113803  53.1000        S       1     0
4      0            373450   8.0500        S       0     1
```

4: Run TODO 6: Problems Terminal Python Console Event Log

21:1 CRLF UTF-8 4 spaces Python 3.8 (Class)

Partitioning into train and test sets

```
18  from sklearn.model_selection import train_test_split
19
20  y = dfClean['Survived']
21  X = dfClean.loc[:, ['Pclass', 'Age', 'female', 'male']]
22
23  print(y.shape)
24  print(X.shape)
25
26  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
27
28  print(y_train.shape)
29  print(X_train.shape)
30  print(y_test.shape)
31  print(X_test.shape)
32
```

10 179

Run: in_class_coding

```
C:\Users\Reza\Desktop\ITP449_Fall2020\Class\venv\Scripts\python.exe "C:/Users/Reza/Desktop/ITP449_Fall2020/Class/In Class Coding/in_cla
(889,)
(889, 4)
(622,)
(622, 4)
(267,)
(267, 4)
```