**One Credit Course**

**1.)Salary\_dataset :**

**Program Code :**

1. import numpy as np

import pandas as pd

2. dataset = pd.read\_csv("Salary\_dataset.csv")

dataset.head(100)

3. x = dataset.iloc[:,0:1]

y = dataset.iloc[:,1]

x.head(10)

y.head(20)

4. from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,train\_size=0.8,random\_state=42)

5. from sklearn.linear\_model import LinearRegression

model = LinearRegression()

6. model.fit(x\_train,y\_train)

7. samplepredciton = model.predict([[1240]])

print(samplepredciton)

y\_pred = model.predict(x\_test)

y\_pred

y\_test

y\_pred

list(zip(y\_test,y\_pred))

8. import numpy as np

areas = np.array([1000, 1030, 1060, 1090])

prices = np.array([5618, 5201, 4779, 5245])   m = 200  b = 50000  # intercept

predicted\_prices = m \* areas + b

# Calculate squared differences

squared\_diff = (prices - predicted\_prices) \*\* 2

# Calculate mean squared error

mse = np.mean(squared\_diff)

print("Actual Prices:", prices)

print("Predicted Prices:", predicted\_prices)

print("Squared Differences:", squared\_diff)

print("Mean Squared Error:", mse)

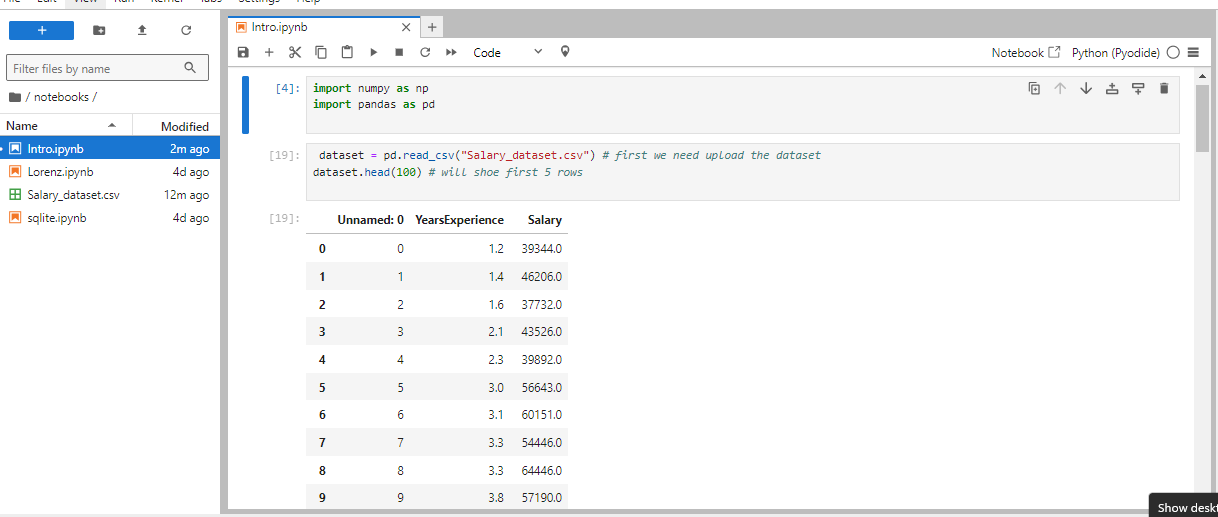
9.from sklearn.metrics import r2\_score

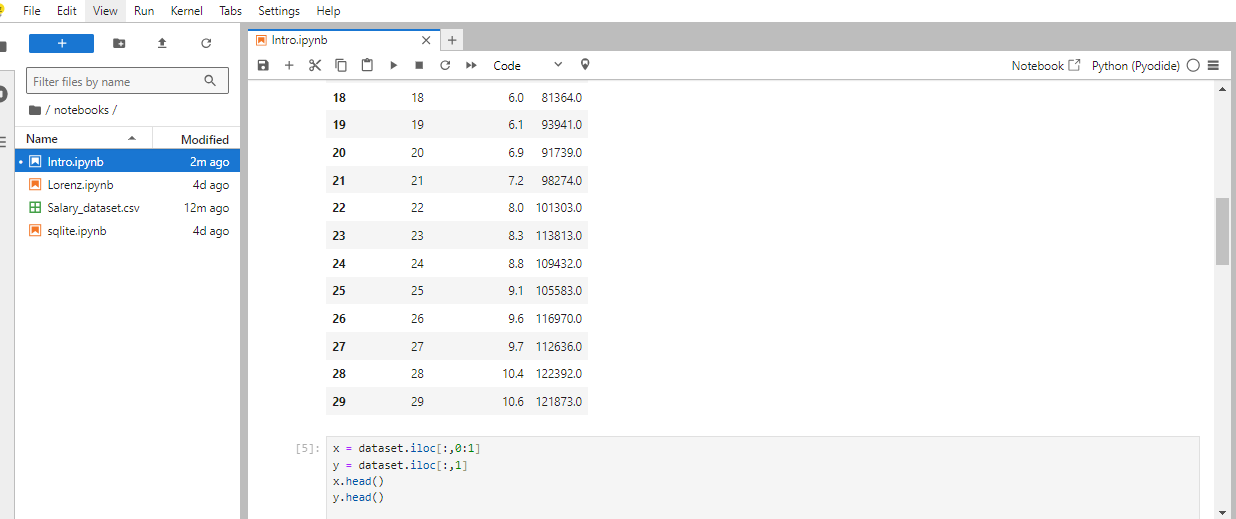
r2 = r2\_score(y\_test, y\_pred)

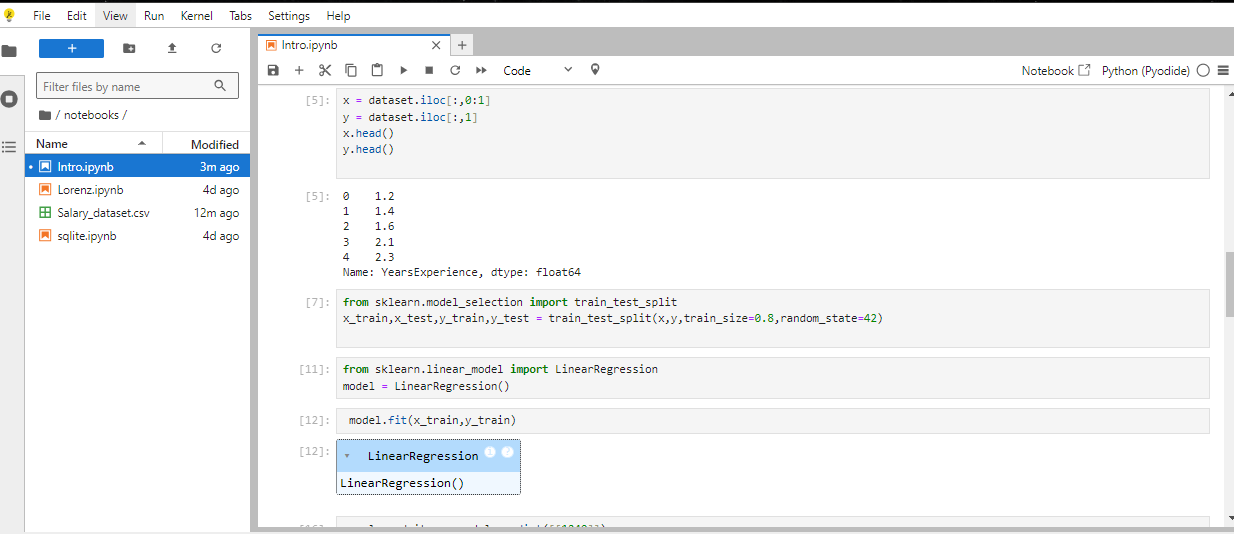
print(f'R-squared score: {r2:.2f}')

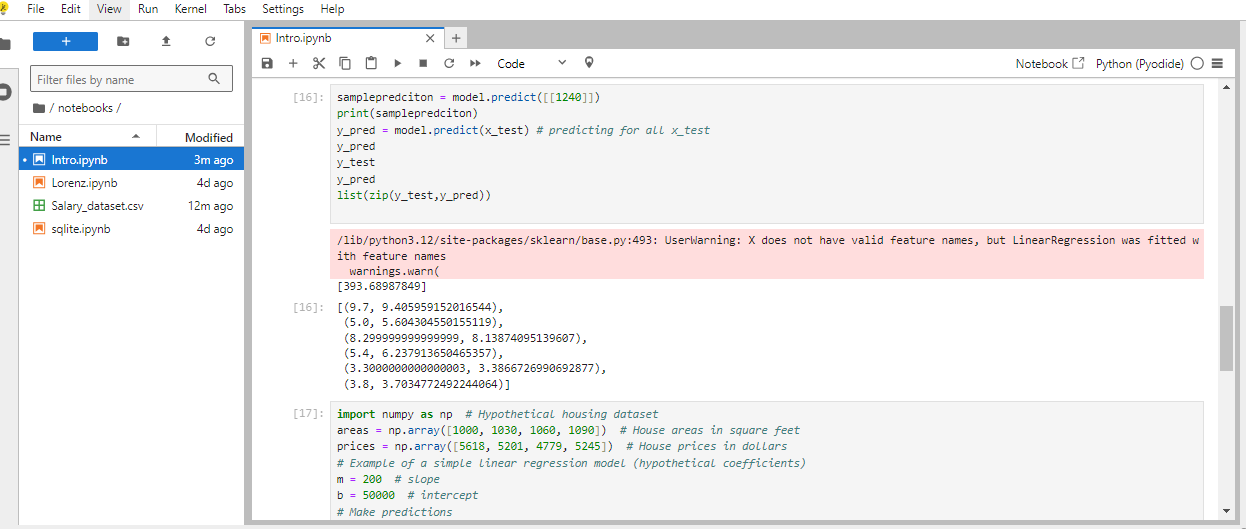
y\_test

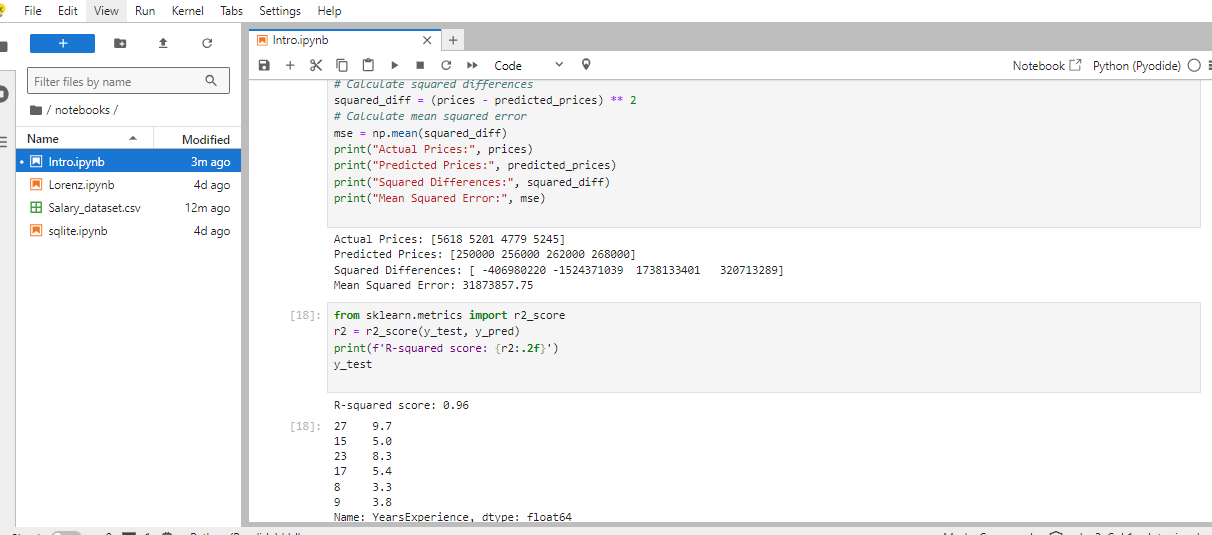
**ScreenShots:**

****

****

****

****

****

**2.)Penguins\_binary\_classification:**

**Code :**

1.import numpy as np

import pandas as pd

2. dataset = pd.read\_csv("penguins\_binary\_classification.csv") # first we need upload the dataset

dataset.head()

3. x = dataset.iloc[:,0:1]

y = dataset.iloc[:,1]

x.head()

y.head()

4. from sklearn.model\_selection import train\_test\_split

x\_train,x\_test,y\_train,y\_test = train\_test\_split(x,y,train\_size=0.8,random\_state=42)

5. from sklearn.linear\_model import LinearRegression

model = LinearRegression()

6. model.fit(x\_train,y\_train)

7. samplepredciton = model.predict([[1240]])

print(samplepredciton)

y\_pred = model.predict(x\_test) # predicting for all x\_test

y\_pred

y\_test

y\_pred

list(zip(y\_test,y\_pred))

8. import numpy as np

areas = np.array([1000, 1030, 1060, 1090])

prices = np.array([5618, 5201, 4779, 5245 m = 200  b = 50000

predicted\_prices = m \* areas + b

# Calculate squared differences

squared\_diff = (prices - predicted\_prices) \*\* 2

# Calculate mean squared error

mse = np.mean(squared\_diff)

print("Actual Prices:", prices)

print("Predicted Prices:", predicted\_prices)

print("Squared Differences:", squared\_diff)

print("Mean Squared Error:", mse)

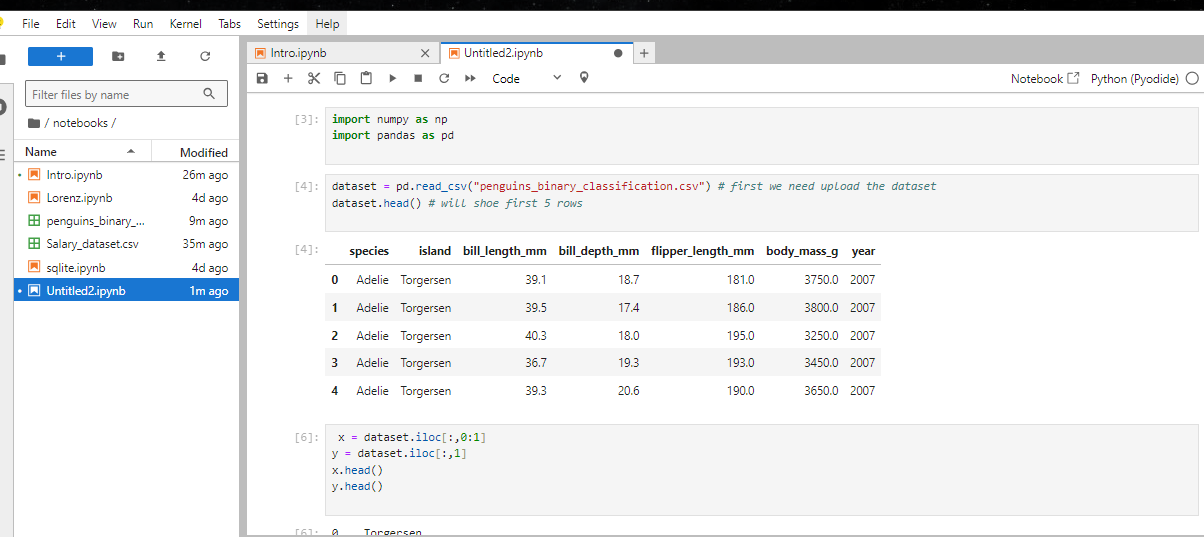
9.from sklearn.metrics import r2\_score

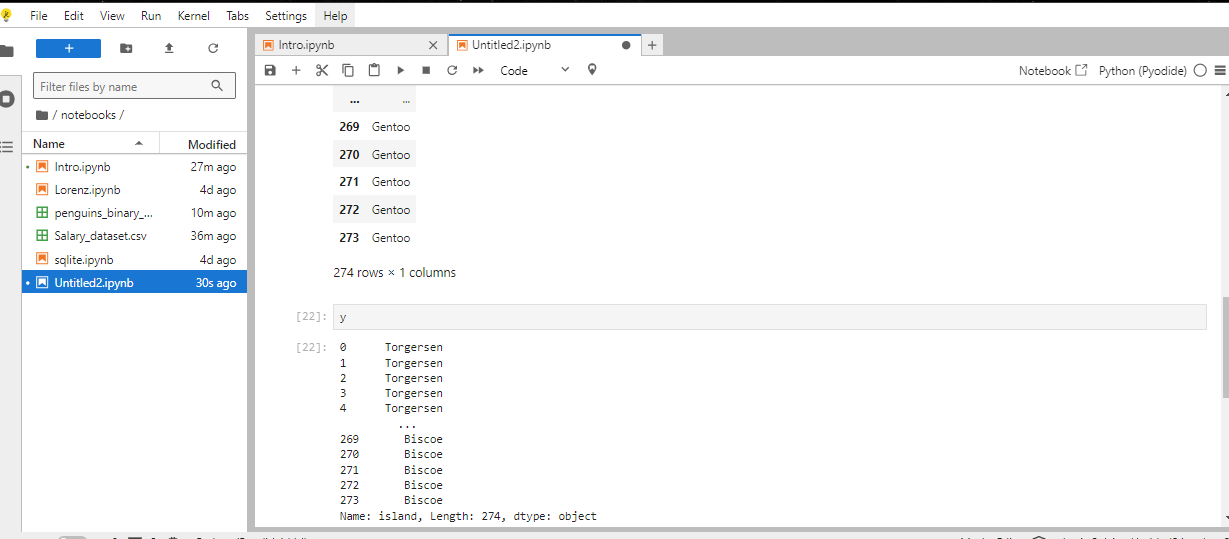
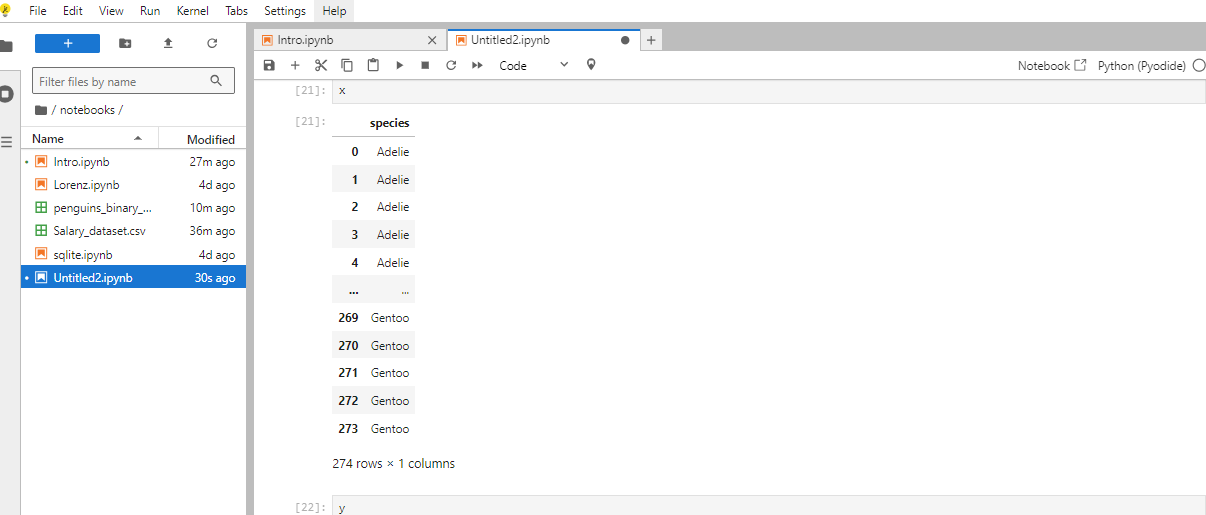
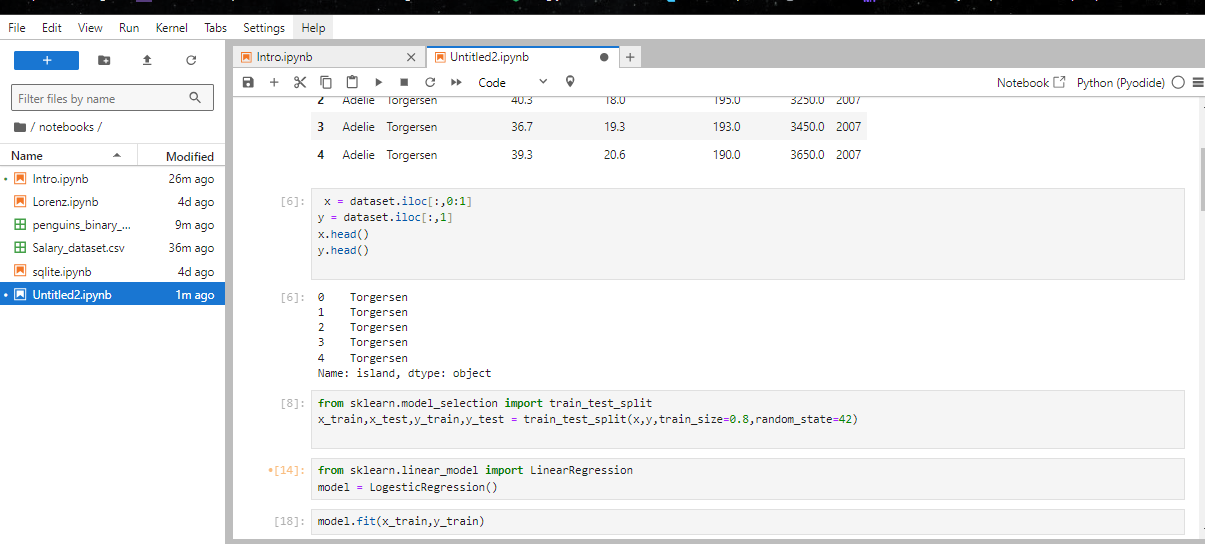
r2 = r2\_score(y\_test, y\_pred)

print(f'R-squared score: {r2:.2f}')

y\_test

**ScreenShots:**

****

****

