	A	B	C	D	E
1	age	education.num	capital.gain	hours.per.week	income
2	90	9	0	40	<=50K
3	82	9	0	18	<=50K
4	66	10	0	40	<=50K
5	54	4	0	40	<=50K
6	41	10	0	40	<=50K
7	34	9	0	45	<=50K
8	38	6	0	40	<=50K
9	74	16	0	20	>50K
10	68	9	0	40	<=50K
11	41	10	0	60	>50K
12	45	16	0	35	>50K
13	38	15	0	45	>50K
14	52	13	0	20	>50K
15	32	14	0	55	>50K
16	51	16	0	40	>50K
17	46	15	0	40	>50K
18	45	7	0	76	>50K
19	57	14	0	50	>50K
20	22	12	0	40	>50K
21	34	13	0	50	>50K
22	37	13	0	40	>50K
23	29	7	0	42	<=50K
24	61	9	0	25	<=50K
25	51	10	0	40	<=50K
26	61	9	0	32	<=50K
27	21	11	0	40	<=50K
28	33	2	0	32	<=50K
29	49	3	0	40	<=50K
30	37	6	0	50	>50K

colab.research.google.com/drive/1Krg9OVeB2StEWXaw1zrJb38tbVjGNHX9

Day4_SalaryEstimation_K_NN.ipynb

File Edit View Insert Runtime Tools Help Last saved at 7:36 PM

Comment Share

+ Code + Text

RAM Disk

Editing

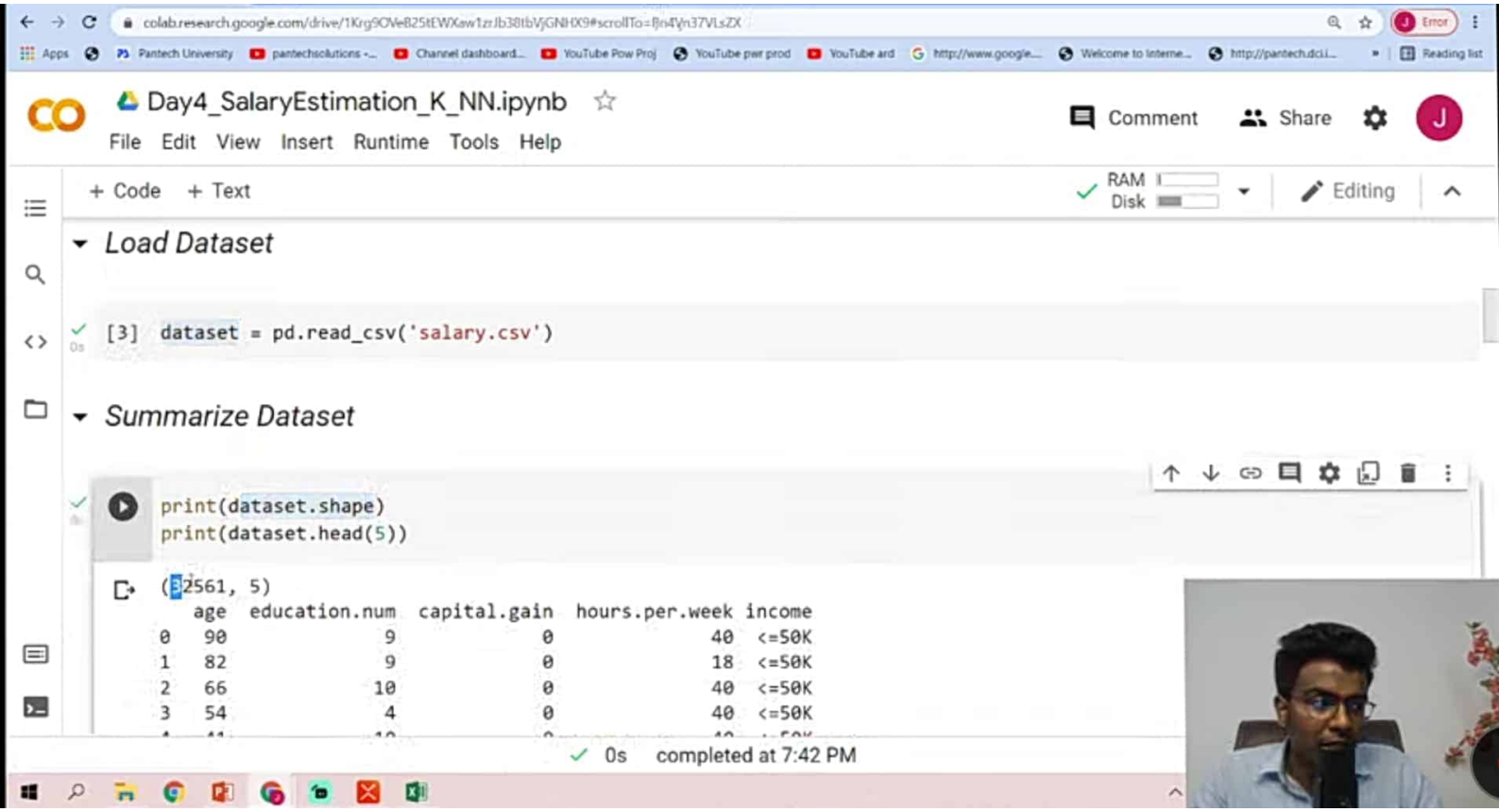
Day-4 | Salary Estimation | K-NEAREST NEIGHBOUR model

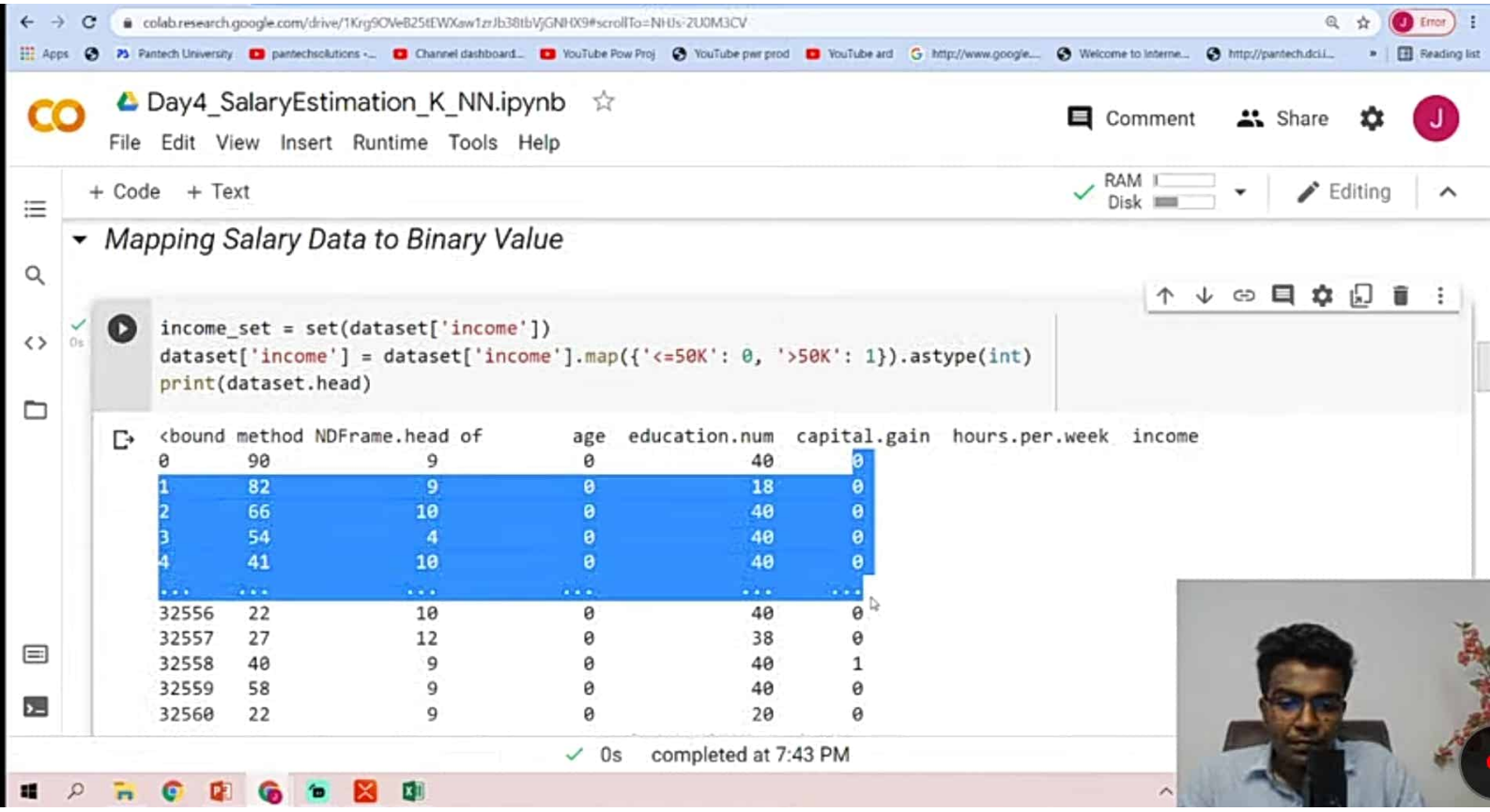
Importing Libraries

```
[ ] import pandas as pd #useful for loading the dataset
    import numpy as np #to perform array
```

Choose Dataset from Local Directory

```
[ ] from google.colab import files
    uploaded = files.upload()
```


Day4_SalaryEstimation_K_NN.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
Load Dataset
[3] dataset = pd.read_csv('salary.csv')
Summarize Dataset
print(dataset.shape)
print(dataset.head(5))
(32561, 5)
age education.num capital.gain hours.per.week income
0 90 9 0 40 <=50K
1 82 9 0 18 <=50K
2 66 10 0 40 <=50K
3 54 4 0 40 <=50K
0s completed at 7:42 PM


Day4_SalaryEstimation_K_NN.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
Mapping Salary Data to Binary Value
income_set = set(dataset['income'])
dataset['income'] = dataset['income'].map({'<=50K': 0, '>50K': 1}).astype(int)
print(dataset.head)
<bound method NDFrame.head of age education.num capital.gain hours.per.week income
0 90 9 0 40 0
1 82 9 0 18 0
2 66 10 0 40 0
3 54 4 0 40 0
4 41 10 0 40 0
... ...
32556 22 10 0 40 0
32557 27 12 0 38 0
32558 40 9 0 40 1
32559 58 9 0 40 0
32560 22 9 0 20 0
0s completed at 7:43 PM

colab.research.google.com/drive/1Krg9OVeB25tEWXaw1zrJb38tbVjGNHX9#scrollTo=LKL0-37RNzOv

Day4_SalaryEstimation_K_NN.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
[5]  32560   22              9              0              20       0

     [32561 rows x 5 columns]>
```

Segregate Dataset into X(Input/IndependentVariable) & Y(Output/DependentVariable)

```
X = dataset.iloc[:, :-1].values
X
```

```
array([[90,  9,  0, 40],
       [82,  9,  0, 18],
       [66, 10,  0, 40],
       ...,
       [40,  9,  0, 40],
       [58,  9,  0, 40],
       [22,  9,  0, 20]])
```

0s completed at 7:45 PM

colab.research.google.com/drive/1Krg9OVeB25tEWXaw1zrJb38tbVjGNHX9#scrollTo=gYqJfw9CN379

Day4_SalaryEstimation_K_NN.ipynb

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

RAM Disk Editing

```
Y = dataset.iloc[:, -1].values
Y
```

```
array([0, 0, 0, ..., 1, 0, 0])
```

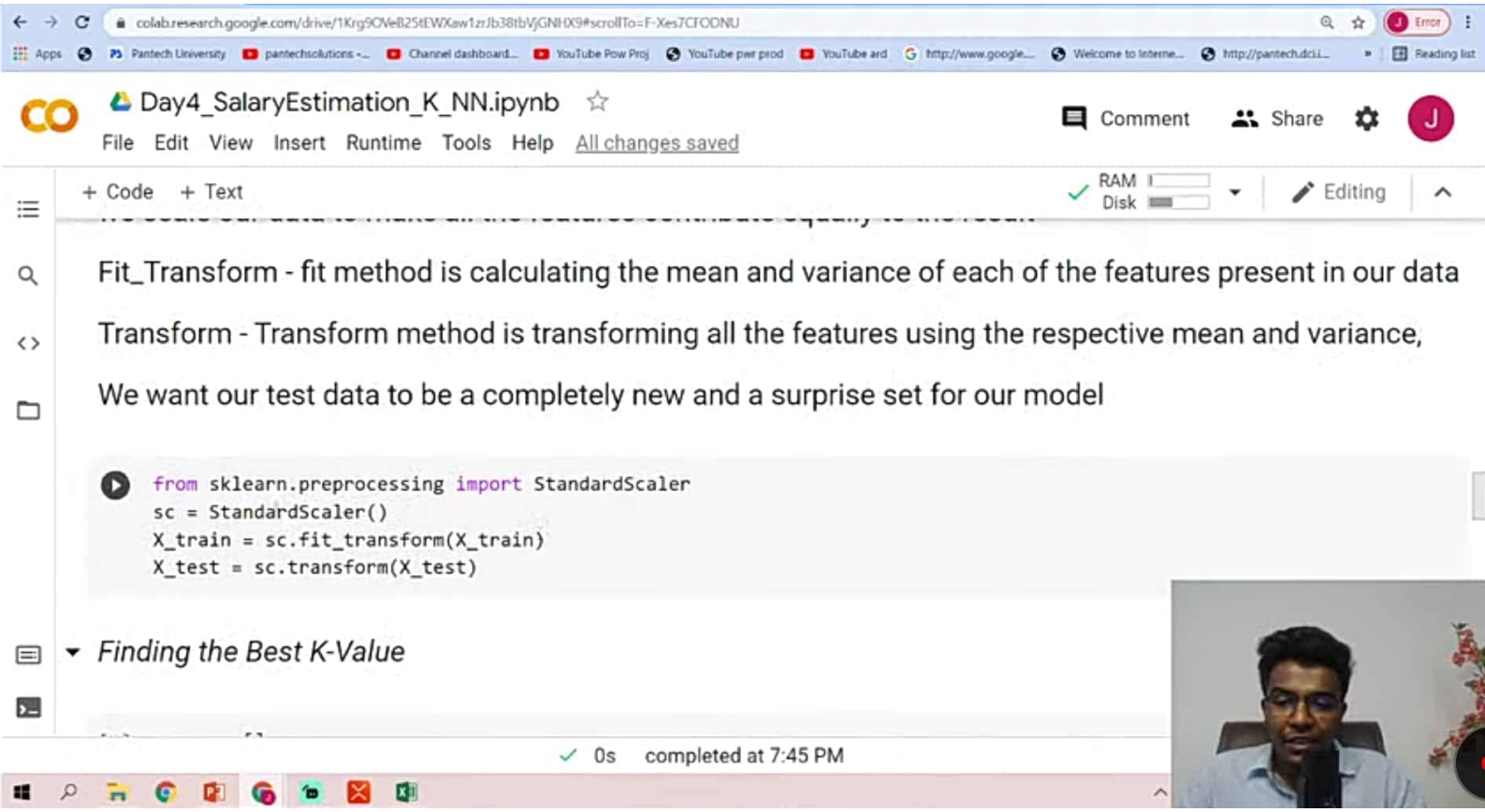
Splitting Dataset into Train & Test

```
[ ] from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.25, random_state = 0)
```

Feature Scaling

we scale our data to make all the features contribute equally to the result

0s completed at 7:45 PM


colab.research.google.com/drive/1Krg9OVeB25tEWXaw1zrJb38tbVjGNHX9#scrollTo=F-Xes7CFODNU
Day4_SalaryEstimation_K_NN.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
Fit_Transform - fit method is calculating the mean and variance of each of the features present in our data
Transform - Transform method is transforming all the features using the respective mean and variance,
We want our test data to be a completely new and a surprise set for our model
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
Finding the Best K-Value
0s completed at 7:45 PM

colab.research.google.com/drive/1Krg9OVeB25tEWXaw1zrJb38tbVjGNHX9#scrollTo=WKES0Vn1YINO

Day4_SalaryEstimation_K_NN.ipynb

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

Finding the Best K-Value

```
error = []
from sklearn.neighbors import KNeighborsClassifier
import matplotlib.pyplot as plt

# Calculating error for K values between 1 and 40
for i in range(1, 40):
    model = KNeighborsClassifier(n_neighbors=i)
    model.fit(X_train, y_train)
    pred_i = model.predict(X_test)
    error.append(np.mean(pred_i != y_test))

plt.figure(figsize=(12, 6))
plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
```

Executing (9s) Cell > predict() > check_array() > _assert_all_finite() > _safe_accumulator_op() > sum() > _wrapre

colab.research.google.com/drive/1Krg9OVeB25tEWXaw1zrJb38tbVjGNHX9#scrollTo=WKES0Vn1YINO

Day4_SalaryEstimation_K_NN.ipynb

File Edit View Insert Runtime Tools Help All changes saved

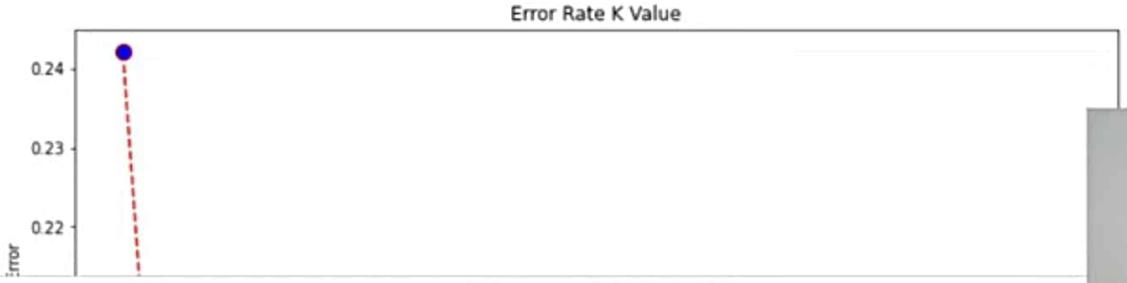
Comment Share

+ Code + Text

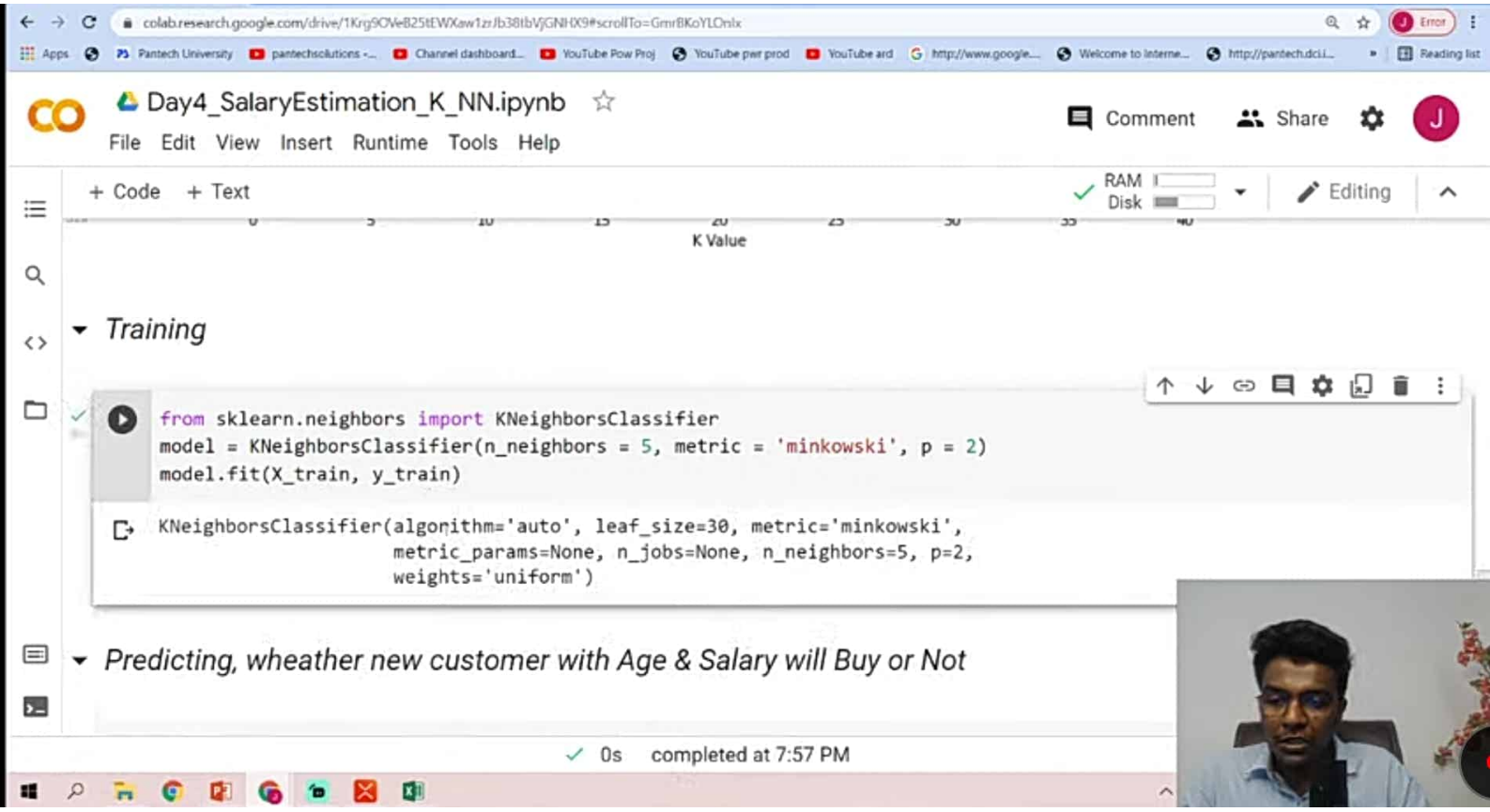
RAM Disk Editing

```
plt.figure(figsize=(12, 6))
plt.plot(range(1, 40), error, color='red', linestyle='dashed', marker='o',
         markerfacecolor='blue', markersize=10)
plt.title('Error Rate K Value')
plt.xlabel('K Value')
plt.ylabel('Mean Error')
```

```
Text(0, 0.5, 'Mean Error')
```



32s completed at 7:48 PM


colab.research.google.com/drive/1Krg9OVeB25tEWXaw1zrJb38tbVjGNHX9#scrollTo=GmrBKoYLOnlx
Day4_SalaryEstimation_K_NN.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
K Value
Training
from sklearn.neighbors import KNeighborsClassifier
model = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p = 2)
model.fit(X_train, y_train)
KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
metric_params=None, n_jobs=None, n_neighbors=5, p=2,
weights='uniform')
Predicting, wheather new customer with Age & Salary will Buy or Not
0s completed at 7:57 PM

Day4_SalaryEstimation_K_NN.ipynb

File Edit View Insert Runtime Tools Help Saving...

Comment Share

+ Code + Text

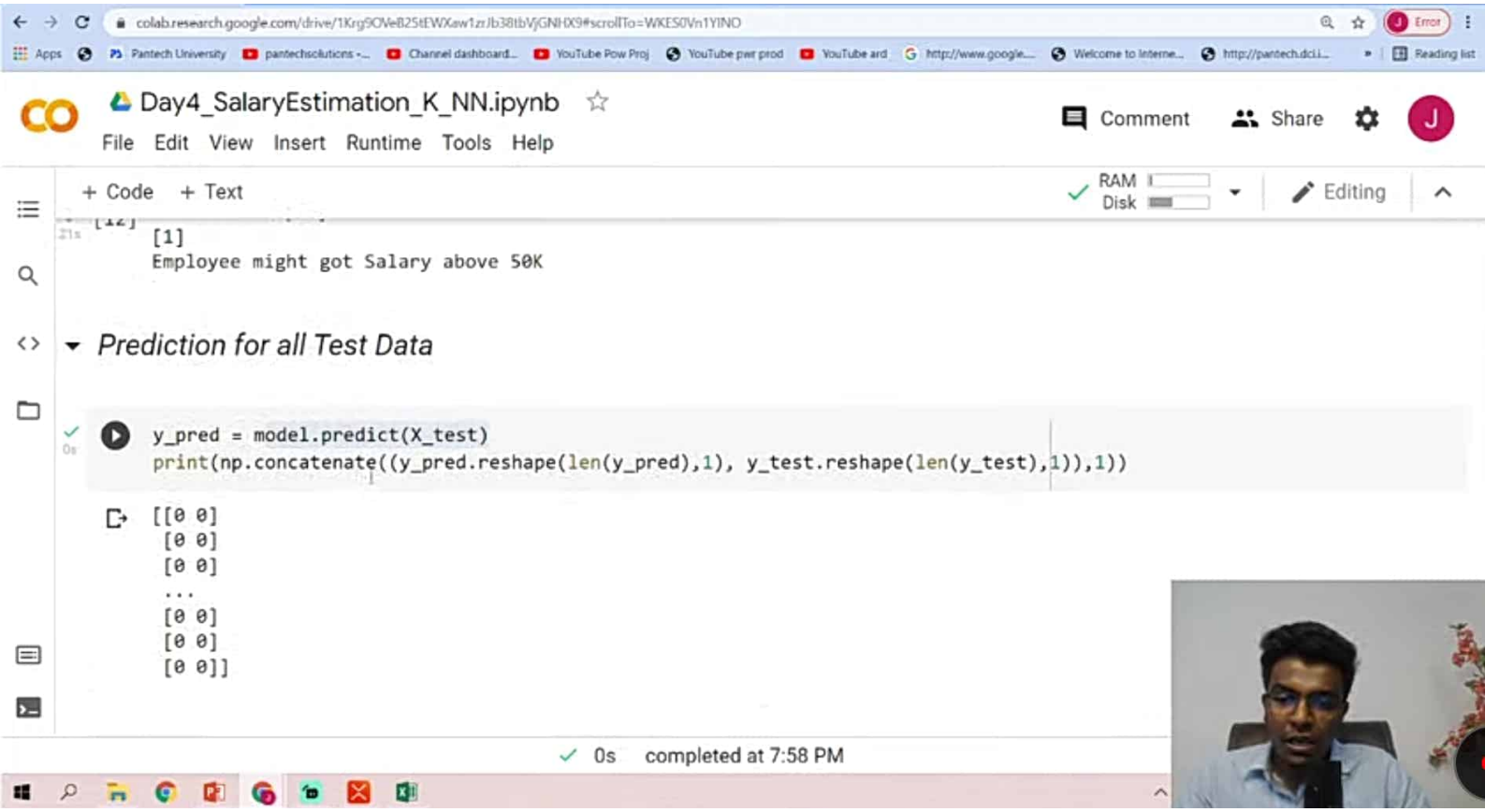
RAM Disk Editing

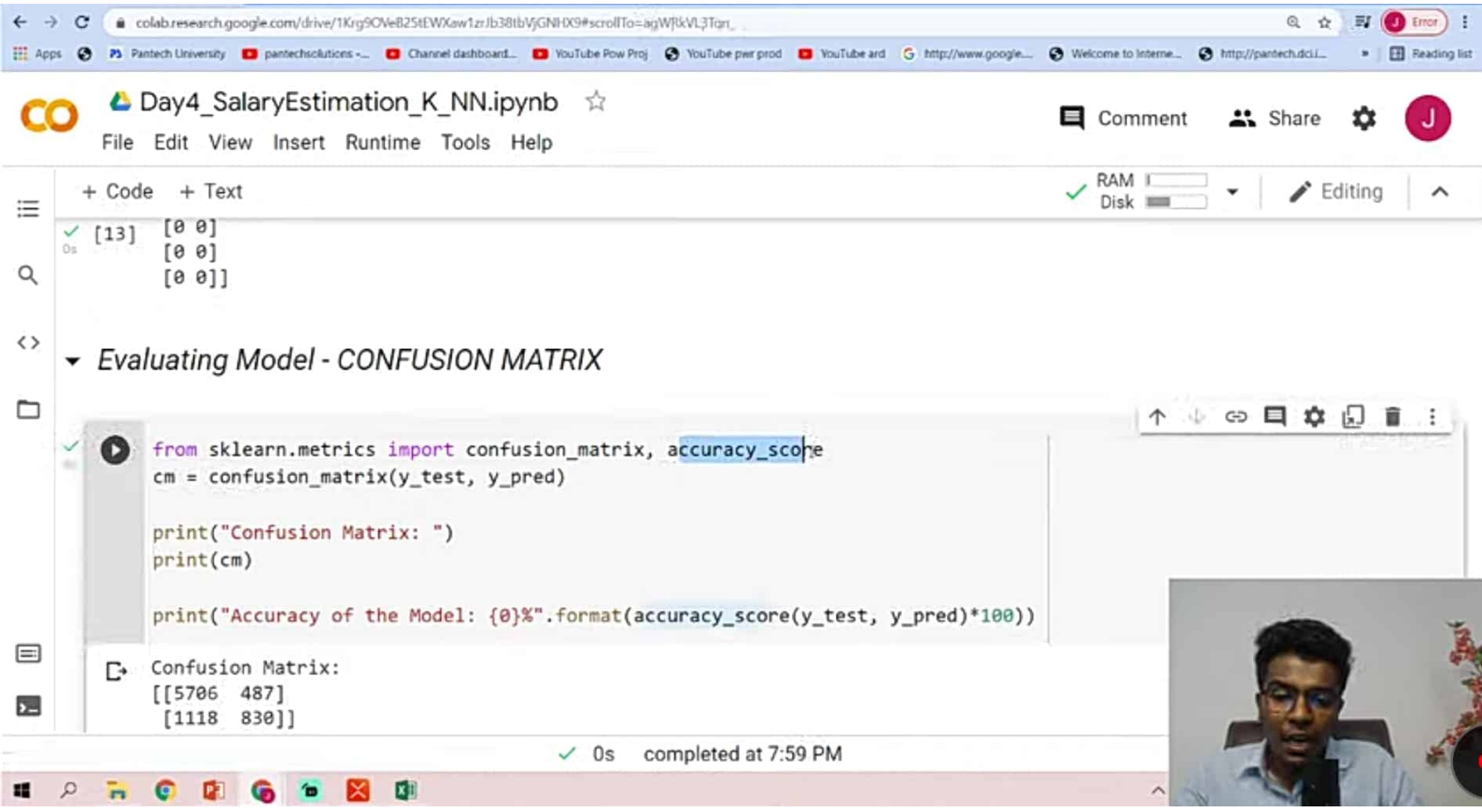
```
age = int(input("Enter New Employee's Age: "))
edu = int(input("Enter New Employee's Education: "))
cg = int(input("Enter New Employee's Captital Gain: "))
wh = int(input("Enter New Employee's Hour's Per week: "))
newEmp = [[age,edu,cg,wh]]
result = model.predict(sc.transform(newEmp))
print(result)

if result == 1:
  print("Employee might got Salary above 50K")
else:
  print("Customer might not got  Salary above 50K")
```

Prediction for all Test Data

0s completed at 7:57 PM


colab.research.google.com/drive/1Krg9OVeB25tEWXaw1zrJb38tbVjGNHX9#scrollTo=WKES0Vn1YINO
Day4_SalaryEstimation_K_NN.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
[1]
Employee might got Salary above 50K
Prediction for all Test Data
y_pred = model.predict(X_test)
print(np.concatenate((y_pred.reshape(len(y_pred),1), y_test.reshape(len(y_test),1)),1))
[[0 0]
[0 0]
[0 0]
...
[0 0]
[0 0]
[0 0]]
0s completed at 7:58 PM


Day4_SalaryEstimation_K_NN.ipynb
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
[13] [0 0]
[0 0]
[0 0]]
Evaluating Model - CONFUSION MATRIX
from sklearn.metrics import confusion_matrix, accuracy_score
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix: ")
print(cm)
print("Accuracy of the Model: {0}%".format(accuracy_score(y_test, y_pred)*100))
Confusion Matrix:
[[5706 487]
[1118 830]]
0s completed at 7:59 PM