DEFINITION

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

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Example

Total no. of People: 165

Covid- Binary prediction (Yes / No)

ACTUAL

covid Yes – 105 Patient | covid No– 60 people

OUR MODEL PREDICTED

Covid Yes – 110 Patient | Covid No– 55 Patient

1 Confusion Matrix

n=165	Predicted: NO	Predicted: YES
Actual: NO	TN = 50	FP = 10
Actual:	FN = 5	TP = 100

True positives (TP) — These are cases in which we predicted yes (they have the cancer), and they have the cancer

True negatives (TN) — We predicted no, and they don't have the cancer

Type here to search

26°C Rain showers

1 Confusion Matrix

n=165	Predicted: NO	Predicted: YES
Actual: NO	TN = 50	FP = 10
Actual: YES	FN = 5	TP = 100

Parameter

- True positives (TP): These are cases in which we predicted yes (they have the cancer), and they have the cancer
- True negatives (TN): We predicted no, and they don't have the cancer
- False positives (FP)/Type I error: We predicted yes, but they don't actually have the cancer
- False negatives (FN)/Type II error: We predicted no, but they actually have the cancer

n=165	Predicted: NO	Predicted: YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
	55	110	

Evaluation

- Accuracy: Overall correct — **(TP+TN)/total = (100+50)/165 = 0.91**
- Misclassification Rate: Overall wrong | Error Rate — **(FP+FN)/total = (10+5)/165 = 0.09**
- Sensitivity or Recall: True Positive Rate — **TP/actual yes = 100/105 = 0.95**
- False Positive Rate — **FP/actual no = 10/60 = 0.17**
- Specificity: True Negative Rate — **TN/actual no = 50/60 = 0.83**
- Precision — **TP/predicted yes = 100/110 = 0.91**
- Prevalence — **actual yes/total = 105/165 = 0.64**

File Edit Insert Tools View Help | Evaluating Classification model Perform... — Edited

23 Days Evaluated

Mind Map Outliner

Topic Subtopic Relationship Summary Boundary Insert ZEN Pitch Panel

Accuracy is defined as the freedom from mistake or error

2 Accuracy Paradox

Accuracy is not a reliable metric to determine a model performance for imbalanced data

That's why it's called a Paradox because, intuitively, you'd expect a Model with a higher Accuracy to have been the best Model but Accuracy Paradox tells us that this, sometimes, isn't the case

SOLUTION

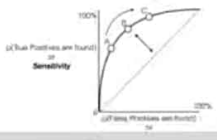
F1 Score 2*((precision*recall)/(precision+recall))

True Positive Rate (TPR) is plot against False Positive Rate (FPR) for the probabilities of the classifier predictions. Then, the area under the plot is calculated

3 Receiver Operating Characteristic (ROC) Curve

Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve

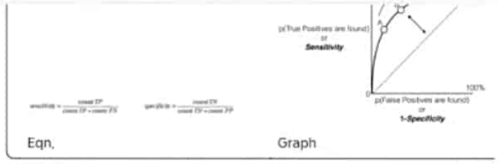
The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes



File Edit Insert Tools View Help | Evaluating Classification model Perform... — Edited | 23 Days Evaluated

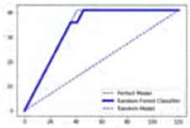
Mind Map | Outliner

Topic | Subtopic | Relationship | Summary | Boundary | Insert | ZEN | Pitch | Panel



Eqn. Graph

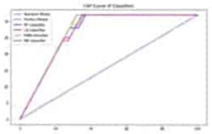
4



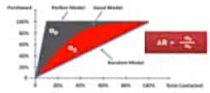
Cumulative Accuracy Profile (CAP) Curve

It tries to analyse how to effectively identify all data points of a given class using minimum number of tries

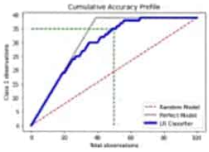
It helps us to understand and conclude about the robustness of the classification model



Method

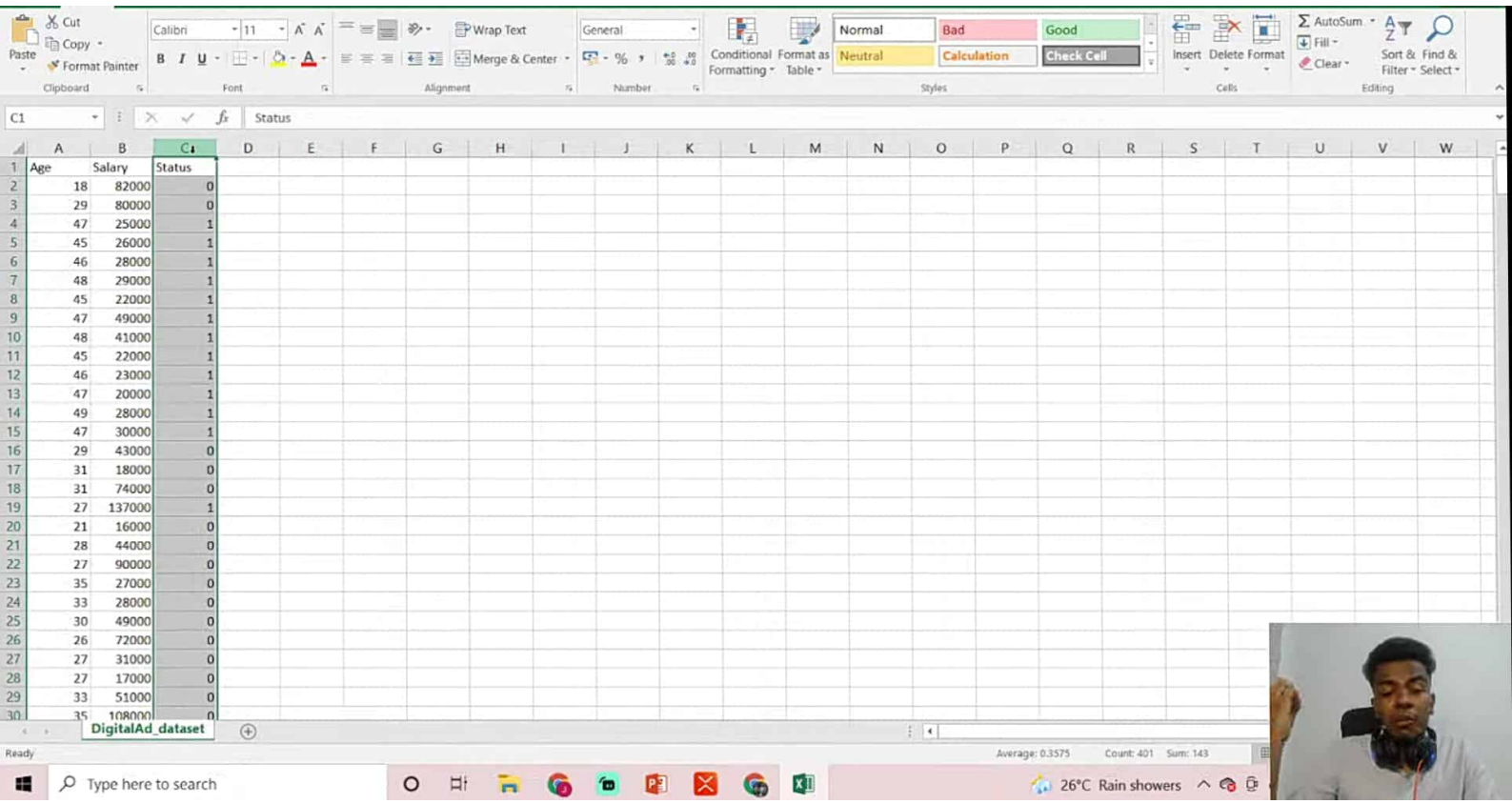


CAP Analysis using Area Under Curve



CAP Analysis using Plot

	Gains	ROC
X axis:	$\frac{count\ TP + count\ FP}{count\ all\ observations}$	$\frac{count\ FP}{count\ TN + count\ FP}$
Y axis:	$\frac{count\ TP}{count\ TP + count\ FN}$	$\frac{count\ TP}{count\ TP + count\ FN}$



Age	Salary	Status
18	82000	0
29	80000	0
47	25000	1
45	26000	1
46	28000	1
48	29000	1
45	22000	1
47	49000	1
48	41000	1
45	22000	1
46	23000	1
47	20000	1
49	28000	1
47	30000	1
29	43000	0
31	18000	0
31	74000	0
27	137000	1
21	16000	0
28	44000	0
27	90000	0
35	27000	0
33	28000	0
30	49000	0
26	72000	0
27	31000	0
27	17000	0
33	51000	0
35	108000	0

Day_9_Validating_Model_Ad_Sale

colab.research.google.com/drive/1mRUJ74J67HCZszOODXsUz4Z-uVTNtmB-#scrollTo=f8ZzLQ-6EFPQ

Apps | (2) SLAM for the ro... | Sensors - ROS Wiki | (2) [ROS Q&A] 031... | The Secret of Nikol... | (2) How to Set Up T... | (8) Setting Up IR re... | jetson.pdf | How to recover the... | fastai_deeplearn_pa... | Reading list

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

Comment | Share

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk | Editing

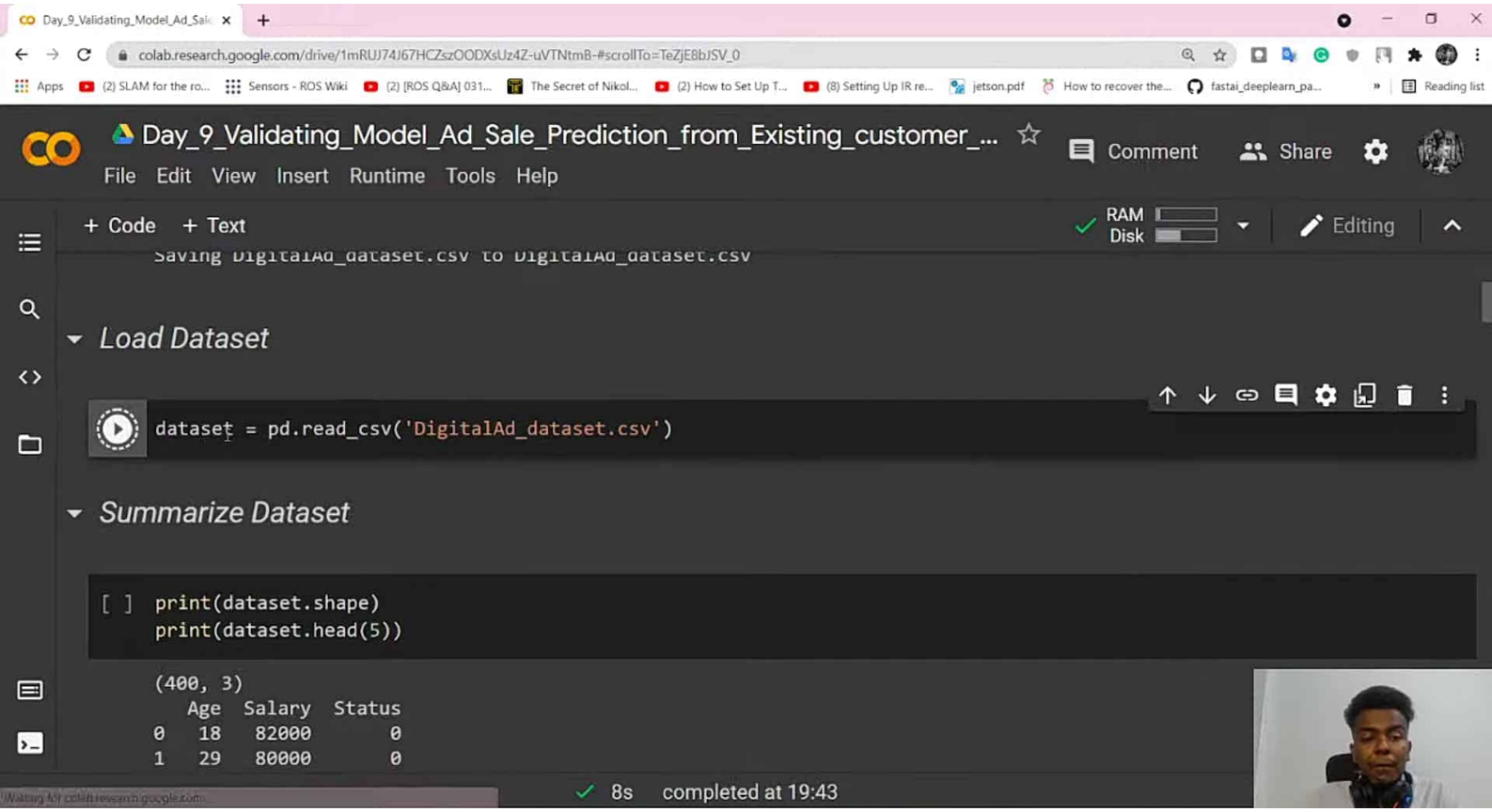
Importing Libraries

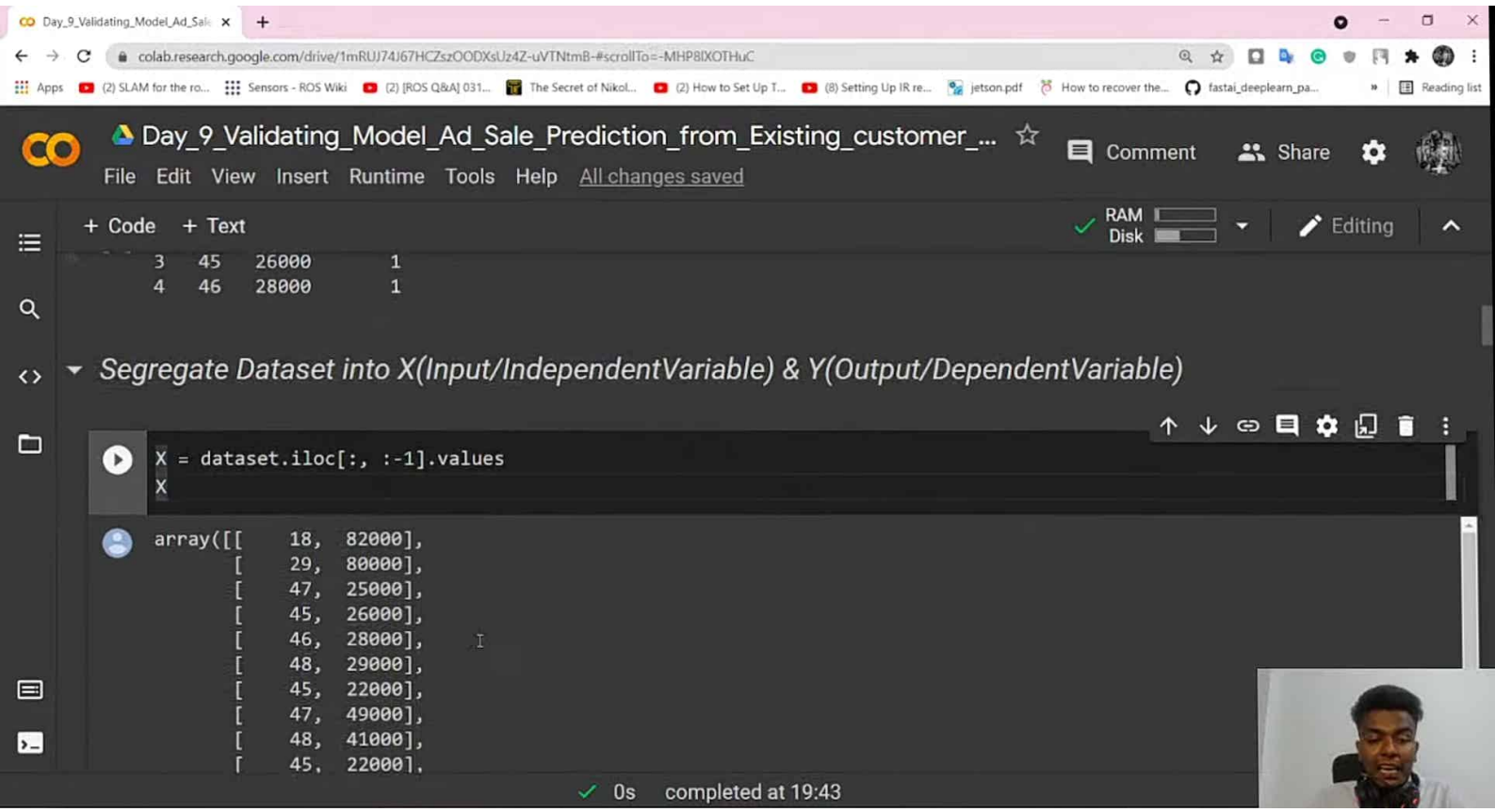
```
import pandas as pd #useful for loading the dataset
import numpy as np #to perform array
```

Choose Dataset file from Local Directory

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

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current b
rerun this cell to enable.
Saving DigitalAd_dataset.csv to DigitalAd_dataset (3).csv


Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
Saving DigitalAd_dataset.csv to DigitalAd_dataset.csv
Load Dataset
dataset = pd.read_csv('DigitalAd_dataset.csv')
Summarize Dataset
[ ] print(dataset.shape)
print(dataset.head(5))
(400, 3)
Age Salary Status
0 18 82000 0
1 29 80000 0
8s completed at 19:43


Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
3 45 26000 1
4 46 28000 1
Segregate Dataset into X(Input/IndependentVariable) & Y(Output/DependentVariable)
X = dataset.iloc[:, :-1].values
X
array([[18, 82000],
[29, 80000],
[47, 25000],
[45, 26000],
[46, 28000],
[48, 29000],
[45, 22000],
[47, 49000],
[48, 41000],
[45, 22000],
0s completed at 19:43

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help

Comment Share

+ Code + Text

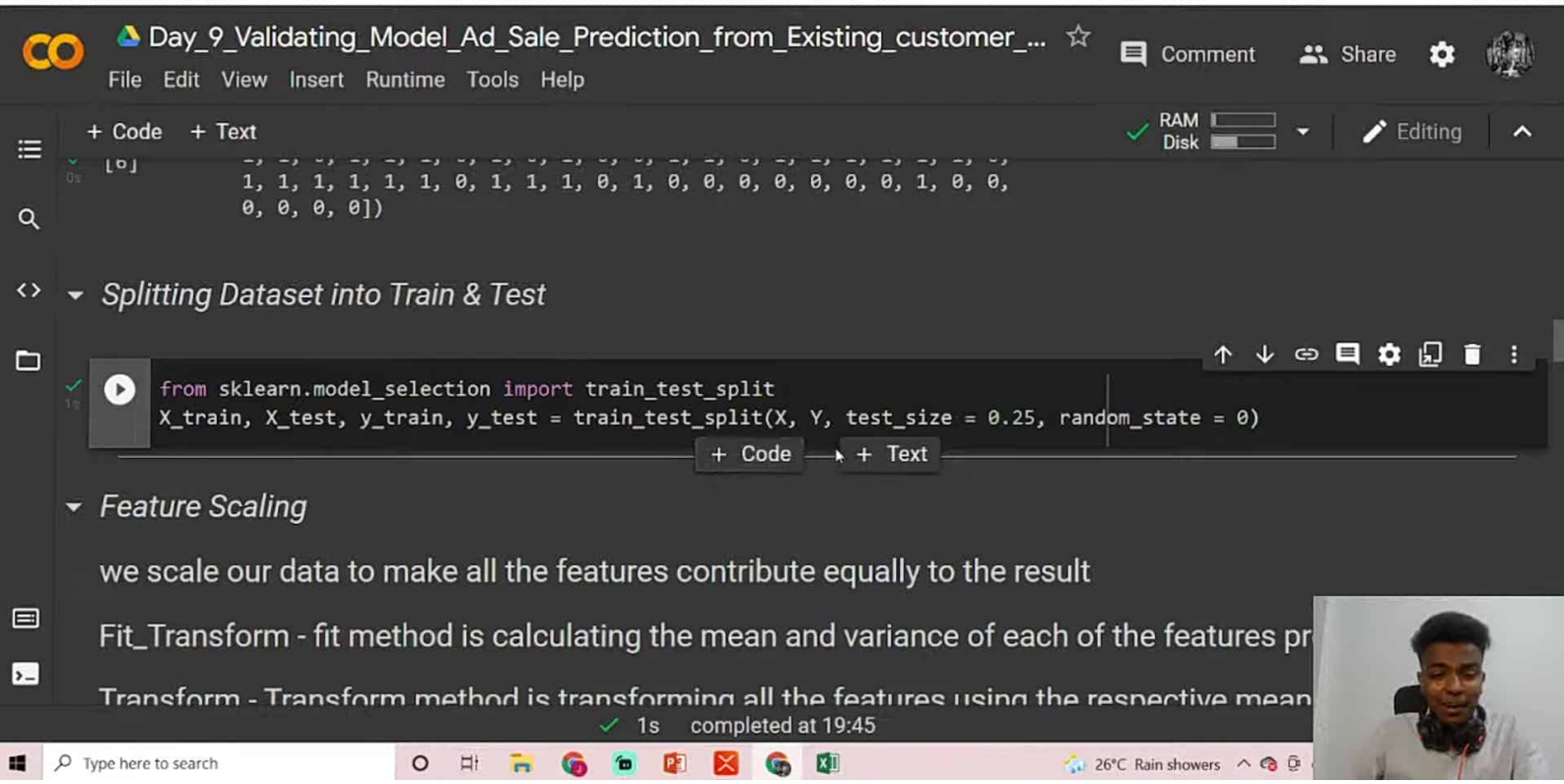
RAM Disk Editing

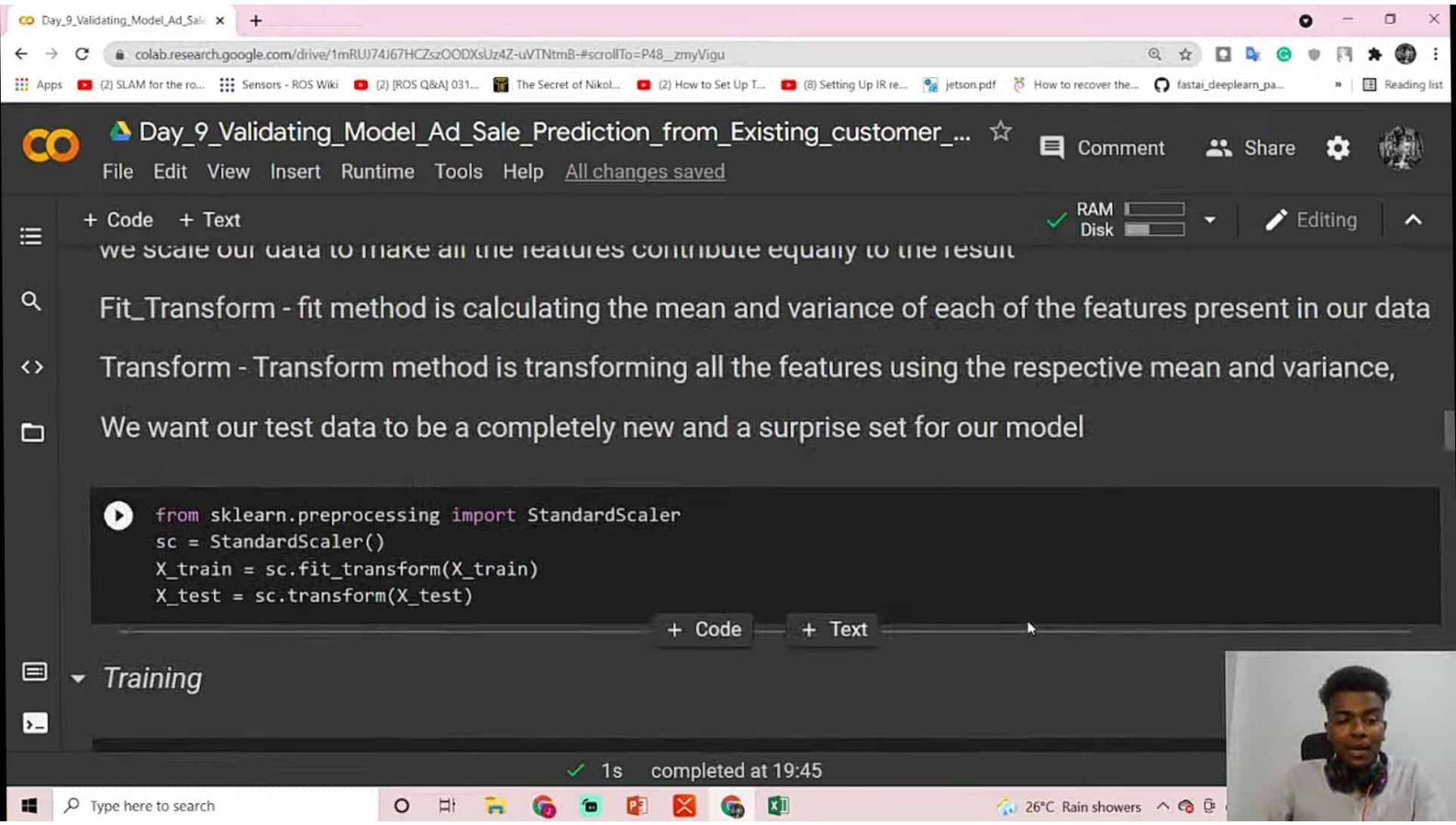
```
[5]          [    25,  33000],
             [    35,  65000],
             [    26,  80000],
             [    26,  52000],
             [    20,  86000],
             [    32,  18000]])
```

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

```
array([0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0,
       0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0,
       1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 0, 1, 0, 1, 1,
       0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 1, 1,
```

0s completed at 19:44


Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...
File Edit View Insert Runtime Tools Help
Comment
Share
+ Code + Text
RAM
Disk
Editing
[6]
1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0,
0, 0, 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)
+ Code
+ Text
Feature Scaling
we scale our data to make all the features contribute equally to the result
Fit_Transform - fit method is calculating the mean and variance of each of the features pr
Transform - Transform method is transforming all the features using the respective mean
1s completed at 19:45
Type here to search
26°C Rain showers


Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
we scale our data to make all the features contribute equally to the result
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)
+ Code + Text
Training
1s completed at 19:45
Type here to search
26°C Rain showers

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[9] from sklearn.linear_model import LogisticRegression
    model = LogisticRegression(random_state = 0)
    model.fit(X_train, y_train)
```

```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, l1_ratio=None, max_iter=100,
                   multi_class='auto', n_jobs=None, penalty='l2',
                   random_state=0, solver='lbfgs', tol=0.0001, verbose=0,
                   warm_start=False)
```

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 1]
```

0s completed at 19:45

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
[0 1]
```

Evaluating Model

Confusion Matrix

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
print("Confusion Matrix: ")
print(cm)
```

```
Confusion Matrix:
[[61  0]
 [20 19]]
```

0s completed at 19:45

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

Accuracy_Score

```
[13] from sklearn.metrics import accuracy_score
     print("Accuracy: {0}%".format(accuracy_score(y_test, y_pred)*100))
```

```
Accuracy: 80.0%
```

Receiver Operating Curve - ROC Curve

```
[ ] from sklearn.metrics import roc_auc_score, roc_curve
    import matplotlib.pyplot as plt

    nsProbability = [0 for _ in range(len(y_test))]
```

0s completed at 19:50

Day_9_Validating_Model_Ad_Sale

colab.research.google.com/drive/1mRUJ74J67HCZszOODXsUz4Z-uVTNtmB-#scrollTo=DcsG6RJqacbw

Apps (2) SLAM for the ro... Sensors - ROS Wiki (2) [ROS Q&A] 031... The Secret of Nikol... (2) How to Set Up T... (8) Setting Up IR re... jetson.pdf How to recover the... fastai_deeplearn_pa... Reading list

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
nsProbability = [0 for _ in range(len(y_test))]
lsProbability = model.predict_proba(X_test)
# keep probabilities for the positive outcome only
lsProbability = lsProbability[:, 1]
# calculate scores
nsAUC = roc_auc_score(y_test, nsProbability)
lrAUC = roc_auc_score(y_test, lsProbability)
# summarize scores
print('No Skill: ROC AUC=%.3f' % (nsAUC*100))
print('Logistic Skill: ROC AUC=%.3f' % (lrAUC*100))
# calculate roc curves
nsFP, nsTP, _ = roc_curve(y_test, nsProbability)
lrFP, lrTP, _ = roc_curve(y_test, lsProbability)
# plot the roc curve for the model
plt.plot(nsFP, nsTP, linestyle='--', label='No Skill')
plt.plot(lrFP, lrTP, marker='*', label='Logistic')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# show the legend
```

0s completed at 19:50

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

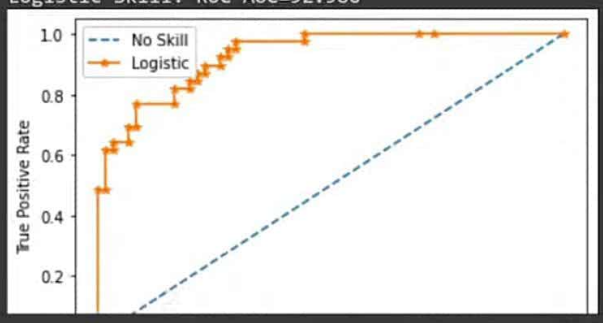
Comment Share

+ Code + Text

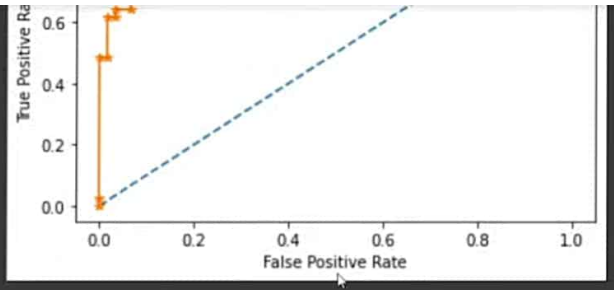
RAM Disk Editing

```
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# show the legend
plt.legend()
plt.show()
```

```
No Skill: ROC AUC=50.000
Logistic Skill: ROC AUC=92.980
```



0s completed at 19:50



Cross Validation Score

```
[ ]  from sklearn.model_selection import cross_val_score
     from sklearn.model_selection import KFold
     kfold = KFold(n_splits=10, random_state=100)
     result = cross_val_score(model, X, Y, cv=kfold)
```

✓ 0s completed at 19:50

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

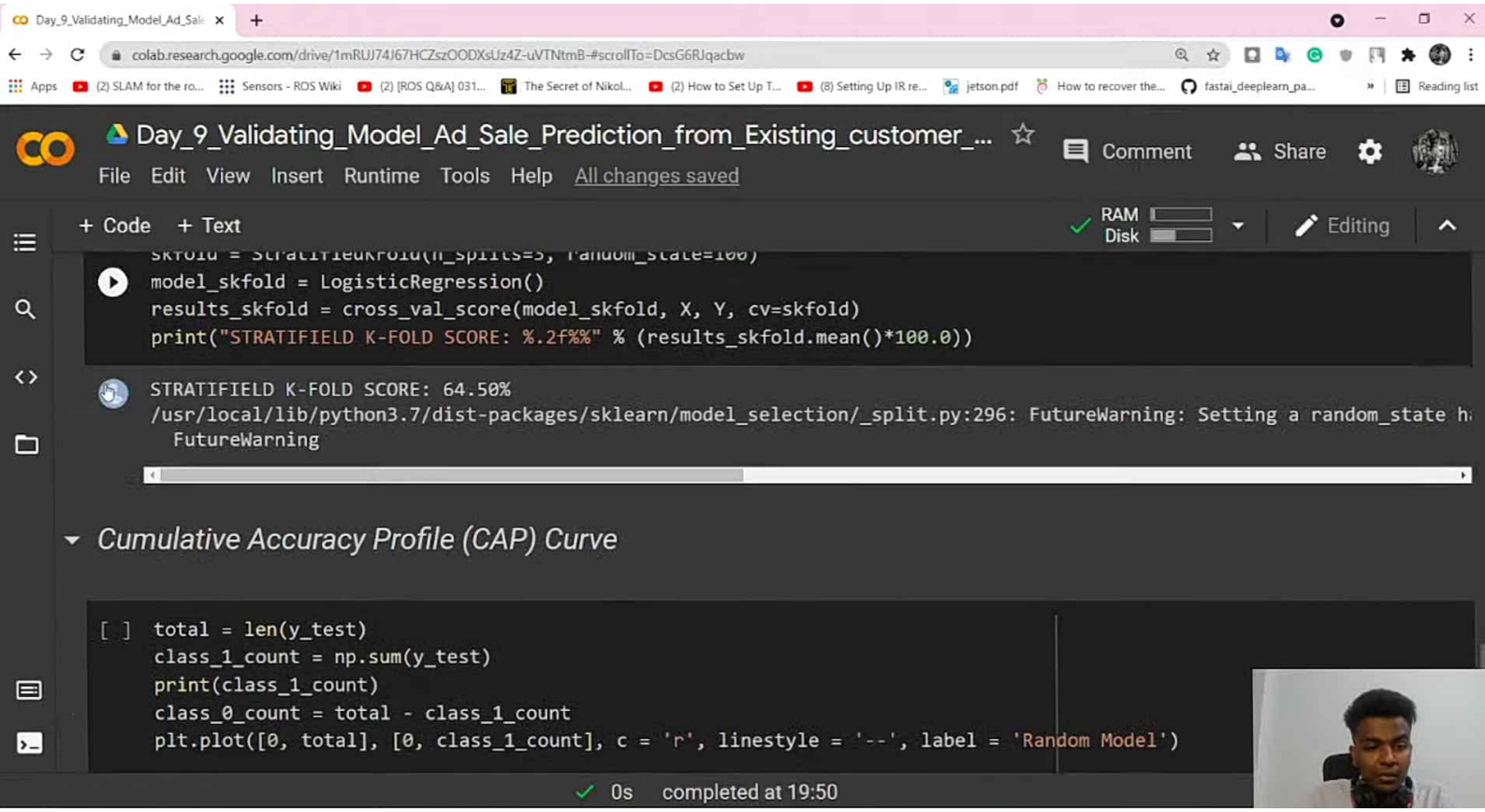
```
kfold = KFold(n_splits=10, random_state=100)
result = cross_val_score(model, X, Y, cv=kfold)
print("CROSS VALIDATION SCORE: %.2f%%" % (result.mean()*100.0))
```

```
CROSS VALIDATION SCORE: 64.25%
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_split.py:296: FutureWarning: Setting a random_state h
  FutureWarning
```

Stratifield K-fold Cross Validation

```
[ ] from sklearn.model_selection import cross_val_score
    from sklearn.model_selection import StratifiedKFold
    skfold = StratifiedKFold(n_splits=3, random_state=100)
    model_skfold = LogisticRegression()
    results_skfold = cross_val_score(model_skfold, X, Y, cv=skfold)
    print("STRATIFIELD K-FOLD SCORE: %.2f%%" % (results_skfold.mean()*100.0))
```

0s completed at 19:50


Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
Editing
skfold = StratifiedKFold(n_splits=5, random_state=100)
model_skfold = LogisticRegression()
results_skfold = cross_val_score(model_skfold, X, Y, cv=skfold)
print("STRATIFIELD K-FOLD SCORE: %.2f%%" % (results_skfold.mean()*100.0))
STRATIFIELD K-FOLD SCORE: 64.50%
/usr/local/lib/python3.7/dist-packages/sklearn/model_selection/_split.py:296: FutureWarning: Setting a random_state h
FutureWarning
Cumulative Accuracy Profile (CAP) Curve
[ ] total = len(y_test)
class_1_count = np.sum(y_test)
print(class_1_count)
class_0_count = total - class_1_count
plt.plot([0, total], [0, class_1_count], c = 'r', linestyle = '--', label = 'Random Model')
0s completed at 19:50

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

colab.research.google.com/drive/1mRUJ74J67HCZszOODXsUz4Z-uVTNtmB-#scrollTo=DcsG6RJqacbw

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk

Editing

```
plt.plot([0, class_1_count, total],
         [0, class_1_count, class_1_count],
         c = 'grey',
         linewidth = 2,
         label = 'Perfect Model')

probs = model.predict_proba(X_test)
probs = probs[:, 1]
model_y = [y for _, y in sorted(zip(probs, y_test), reverse = True)]
y_values = np.append([0], np.cumsum(model_y))
x_values = np.arange(0, total + 1)

plt.plot(x_values,
         y_values,
         c = 'b',
         label = 'LR Classifier',
         linewidth = 4)
```

0s completed at 19:50

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

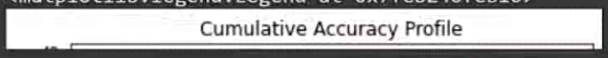
```
index = int((50*total / 100))

## 50% Verticcal line from x-axis
plt.plot([index, index], [0, y_values[index]], c ='g', linestyle = '--')

## Horizontal line to y-axis from prediction model
plt.plot([0, index], [y_values[index], y_values[index]], c = 'g', linestyle = '--')

class_1_observed = y_values[index] * 100 / max(y_values)
plt.xlabel('Total observations')
plt.ylabel('Class 1 observations')
plt.title('Cumulative Accuracy Profile')
plt.legend(loc = 'lower right')
```

```
100
39
<matplotlib.legend.Legend at 0x7feb240feb10>
```



0s completed at 19:50

Day_9_Validating_Model_Ad_Sale_Prediction_from_Existing_customer_...

File Edit View Insert Runtime Tools Help All changes saved

Comment Share

+ Code + Text

RAM Disk Editing

```
<matplotlib.legend.Legend at 0x7feb240feb10>
```