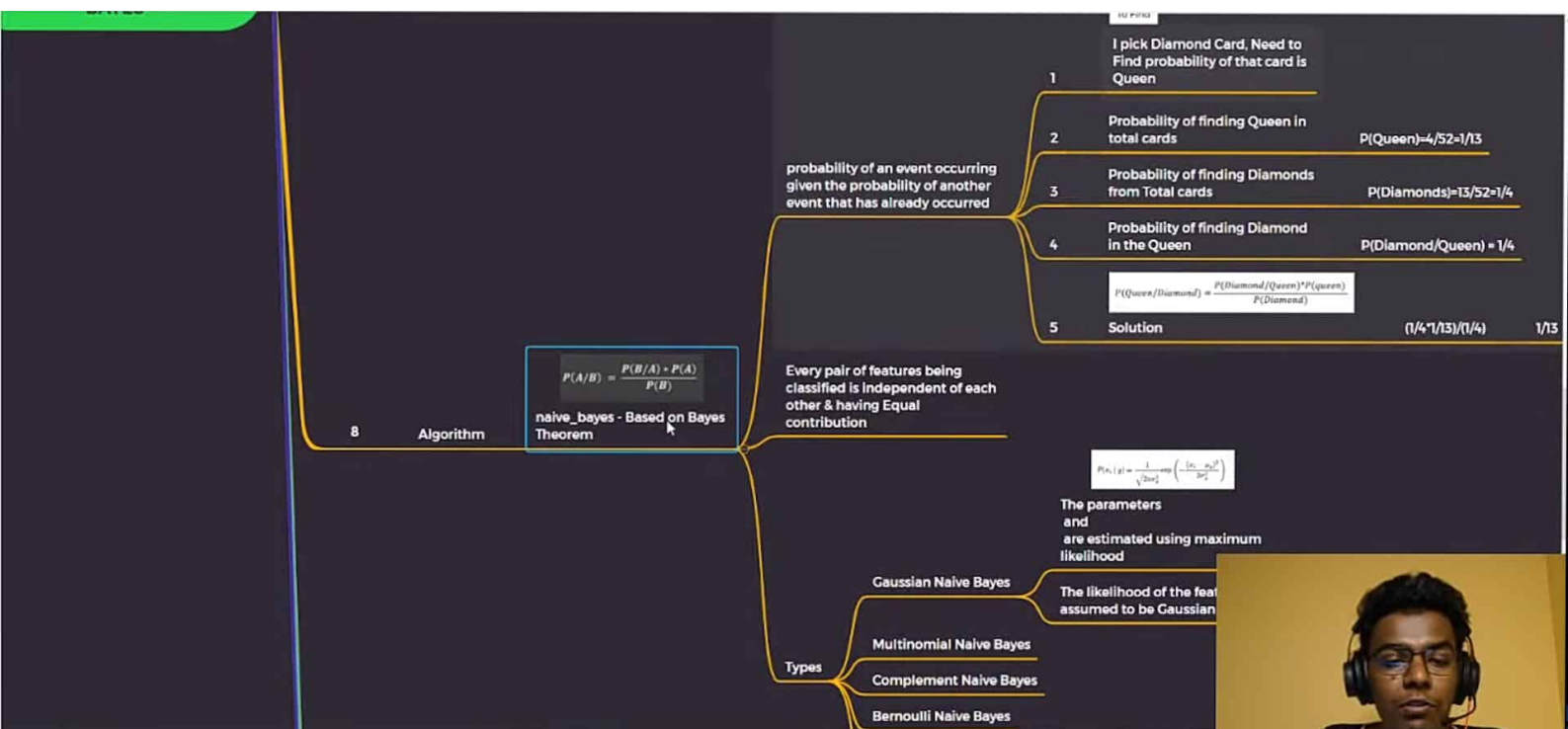


Microsoft Excel interface showing a spreadsheet with data from the Titanic survival dataset. The ribbon includes tabs for Clipboard, Font, Alignment, Number, Styles, Cells, and Editing. The spreadsheet has columns labeled Pclass, Sex, Age, Fare, Survived, and others. The data is organized into rows, with the first row (row 1) containing headers. The spreadsheet is titled "titanicsurvival".

	Pclass	Sex	Age	Fare	Survived
2	3	male	22	7.25	0
3	1	female	38	71.2833	1
4	3	female	26	7.925	1
5	1	female	35	53.1	1
6	3	male	35	8.05	0
7	3	male		8.4583	0
8	1	male	54	51.8625	0
9	3	male	2	21.075	0
10	3	female	27	11.1333	1
11	2	female	14	30.0708	1
12	3	female	4	16.7	1
13	1	female	58	26.55	1
14	3	male	20	8.05	0
15	3	male	39	31.275	0
16	3	female	14	7.8542	0
17	2	female	55	16	1
18	3	male	2	29.125	0
19	2	male		13	1
20	3	female	31	18	0
21	3	female		7.225	1
22	2	male	35	26	0
23	2	male	34	13	1
24	3	female	15	8.0292	1
25	1	male	28	35.5	1
26	3	female	8	21.075	0
27	3	female	38	31.3875	1
28	3	male		7.225	0
29	1	male	19	263	0
30	3	female		7.8792	1





Day6_TitanicSurvivalPrediction_NAIVEBAYES.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Editing

0s


[1] import pandas as pd
import numpy as np

Choose Dataset file from Local Directory

from google.colab import files
uploaded = files.upload()

Load Dataset

0s completed at 19:32



+ Code + Text

✓ RAM Disk Editing ^

0s dataset = pd.read_csv('titanicsurvival.csv')

↑ ↓ ↻ ⌨ ⚙ 📄 🗑 ⋮

Summarize Dataset

print(dataset.shape)
print(dataset.head(5))

Mapping Text Data to Binary Value

[1] female = 1 - male

✓ 0s completed at 19:33



Day6_TitanicSurvivalPrediction_ NAIVEBAYES.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text


RAM Disk

Editing

```
income_set = set(dataset['Sex'])
dataset['Sex'] = dataset['Sex'].map({'female': 0, 'male': 1}).astype(int)
print(dataset.head)
```

	<bound method NDFrame.head of	Pclass	Sex	Age	Fare	Survived
0	3 1 22.0 7.2500	0				
1	1 0 38.0 71.2833	1				
2	3 0 26.0 7.9250	1				
3	1 0 35.0 53.1000	1				
4	3 1 35.0 8.0500	0				
...				
886	2 1 27.0 13.0000	0				
887	1 0 19.0 30.0000	1				
888	3 0 NaN 23.4500	0				
889	1 1 26.0 30.0000	1				

0s completed at 19:34



+ Code + Text

✓ RAM Disk Editing

↑ ↓ ↻ ⌨ ⚙ 📄 🗑 ⋮

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

```
X = dataset.drop('Survived',axis='columns')  
X
```

```
[ ] Y = dataset.Survived  
Y
```

Finding & Removing NA values from our Features X

✓ 0s completed at 19:34



Day6_TitanicSurvivalPrediction_NAIVEBAYES.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

RAM Disk

Editing

Finding & Removing NA values from our Features X

```
[ ] X.columns[X.isna().any()]
```

```
X.Age = X.Age.fillna(X.Age.mean())
```

Test again to check any na value

```
[ ] X.columns[X.isna().any()]
```

0s completed at 19:36

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)
```

Training

```
[12] from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train, y_train)
```

0s completed at 19:43



Day6_TitanicSurvivalPrediction_NAIVEBAYES.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

0s

model.fit(X_train, y_train)

GaussianNB(priors=None, var_smoothing=1e-09)

RAM

Disk

Editing

Predicting, wheather Person Survived or Not

19s

pclassNo = int(input("Enter Person's Pclass number: "))
gender = int(input("Enter Person's Gender 0-female 1-male(0 or 1): "))
age = int(input("Enter Person's Age: "))
fare = float(input("Enter Person's Fare: "))
person = [[pclassNo,gender,age,fare]]
result = model.predict(person)
print(result)

0s completed at 19:43

Comment

Share

Settings

Profile

Day6_TitanicSurvivalPrediction_NAIVEBAYES.ipynb

☆

Colab

colab.research.google.com/drive/164Zt9UI-ulYobMU5O2iC-wDg5c-R2zv#scrollTo=tPSuaammYz_4

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

A small video feed in the bottom right corner of the JupyterLab interface, showing a person wearing a headset and speaking.

Day6_TitanicSurvivalPrediction_NAIVEBAYES.ipynb

File Edit View Insert Runtime Tools Help

+ Code + Text

✓ 19s

```
if result == 1:
    print("Person might be Survived")
else:
    print("Person might not be Survived")
```

Enter Person's Pclass number: 2

Enter Person's Gender 0-female 1-male(0 or 1): 0

Enter Person's Age: 45

Enter Person's Fare: 7.885

[1]

Person might be Survived

RAM


Disk

Editing

↑ ↓ ↶ ↷ ⚙ 🗑 ⋮

▼ Prediction for all Test Data

✓ 18s completed at 19:42



+ Code + Text

RAM Disk Editing

Prediction for all Test Data

```
y_pred = model.predict(X_test)
print(np.column_stack((y_pred,y_test)))
```

```
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[0 0]
[1 1]
[0 0]
[0 0]
```

0s completed at 19:42



Day6_TitanicSurvivalPrediction_NAIVEBAYES.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

✓ 0s [14]

1	0
0	0
0	0
0	0

▼ Accuracy of our Model

✓ 0s

▶

```
from sklearn.metrics import accuracy_score
print("Accuracy of the Model: {0}%".format(accuracy_score(y_test, y_pred)*100))
```

Accuracy of the Model: 77.57847533632287%

✓ 0s completed at 19:43

RAM

Disk

Editing

Comment

Share

⚙️

👤

colab.research.google.com/drive/164Zt9Uf-_uIYobMU5O2iC-wDg5c-R2zv#scrollTo=HDOFRQ0PdZQS

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

A small video inset in the bottom right corner of the Jupyter Notebook interface. It shows a person with dark hair, wearing glasses and large black headphones, with their hand near their face. They are wearing a white shirt with black details on the sleeves. The background is a plain, light-colored wall.