

```
1 !gdown 1ogeP_G780UMvHgZUMAX0ggshuR6067CF
```

Downloading...

From: [https://drive.google.com/uc?id=1ogeP\\_G780UMvHgZUMAX0ggshuR6067CF](https://drive.google.com/uc?id=1ogeP_G780UMvHgZUMAX0ggshuR6067CF)

To: /content/pbp-2021.csv

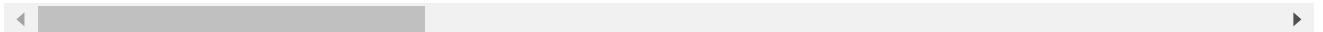
100% 5.49M/5.49M [00:00<00:00, 133MB/s]

```
1 #Looking at the data
2 import pandas as pd
3 df = pd.read_csv('pbp-2021.csv')
4 df.head()
```

↗

	GameId	GameDate	Quarter	Minute	Second	OffenseTeam	DefenseTeam	Down	ToGo
0	2021100306	2021-10-03	2	2	29	NO	NYG	3	
1	2021100306	2021-10-03	2	2	0	NaN	NaN	0	
2	2021100306	2021-10-03	2	2	0	NO	NYG	1	1
3	2021100310	2021-10-03	4	15	0	SEA	SF	2	
4	2021100310	2021-10-03	4	14	57	NaN	NaN	0	

5 rows × 45 columns



## Naive Bayes

<https://www.datacamp.com/tutorial/naive-bayes-scikit-learn>

<https://medium.com/hugo-ferreiras-blog/confusion-matrix-and-other-metrics-in-machine-learning-894688cb1c0a>

```
1 #importing libraries
2 from sklearn.naive_bayes import GaussianNB
3 from sklearn.metrics import confusion_matrix
4 from sklearn.metrics import classification_report
5 from sklearn.model_selection import train_test_split
6 from sklearn.model_selection import validation_curve
7 from sklearn.pipeline import make_pipeline
8 from sklearn import metrics
9 def Naive_bayes(X, y, test_x, test_y):
10     """
11     Description: This method performs training and prints the results of Naive Bayes
12     Input:
13     X: the training dataset,
14     y: the training labels,
15     test_x: test dataset,
```

```

16     test_y: test labels
17     Returns:
18     predictions: list or array of output predictions
19     """
20     #training
21     gaussian = GaussianNB()
22     gaussian.fit(X,y)
23     #prediction
24     predictions= gaussian.predict(test_x)
25     print("Accuracy:", metrics.accuracy_score(test_y, predictions))
26     print(confusion_matrix(test_y, predictions))
27     print(classification_report(test_y, predictions))
28     return predictions
29

```

## Decision Trees

<https://datagy.io/sklearn-decision-tree-classifier/>

```

1 #importing libraries
2 from sklearn.tree import DecisionTreeClassifier
3 from sklearn.metrics import confusion_matrix
4 from sklearn.metrics import classification_report
5 from sklearn.model_selection import train_test_split
6 from sklearn.model_selection import validation_curve
7 from sklearn.pipeline import make_pipeline
8 from sklearn import metrics
9
10 def Decision_tree(X, y, test_x, test_y):
11     """
12     Description: This method performs training and prints the results of Decision Tree
13     Input:
14         X: the training dataset,
15         y: the training labels,
16         test_x: test dataset,
17         test_y: test labels
18     Returns:
19     predictions: list or array of output predictions
20     """
21     #training
22     dt = DecisionTreeClassifier( criterion='gini',
23                                 splitter='best',
24                                 max_depth=None,
25                                 min_samples_split=2,
26                                 min_samples_leaf=1,
27                                 min_weight_fraction_leaf=0.0,
28                                 max_features=None,
29                                 random_state=None,
30                                 max_leaf_nodes=None,
31                                 min_impurity_decrease=0.0,
32                                 class_weight=None,
33                                 ccp_alpha=0.0)
34     dt.fit(X,y)
35     #prediction

```

```
36 predictions = dt.predict(test_x)
37 print("Accuracy:",metrics.accuracy_score(test_y, predictions))
38 print(confusion_matrix(test_y, predictions))
39 print(classification_report(test_y, predictions))
40 return predictions
41
```

[Colab paid products](#) - [Cancel contracts here](#)

