



Data Analysis 2 Project

Duration: 1st Semester

Task 1 (Naive Bayes Classifier)

Submitted By:

<u><i>Reem Ghazi Alosaimi</i></u>	<i>444001268</i>
<i>Remas Majed Almalki</i>	<i>444002871</i>



Introduction:

Naive Bayes classifiers are a collection of classification algorithms based on [Bayes' Theorem](#). It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other. To start with, let us consider a dataset.

One of the most simple and effective classification algorithms, the Naïve Bayes classifier aids in the rapid development of machine learning models with rapid prediction capabilities.

Naïve Bayes algorithm is used for classification problems. It is highly used in text classification. In text classification tasks, data contains high dimension (as each word represent one feature in the data). It is used in spam filtering, sentiment detection, rating classification etc. The advantage of using naïve Bayes is its speed. It is fast and making prediction is easy with high dimension of data.

This model predicts the probability of an instance belongs to a class with a given set of feature value. It is a probabilistic classifier. It is because it assumes that one feature in the model is independent of existence of another feature. In other words, each feature contributes to the predictions with no relation between each other. In real world, this condition satisfies rarely. It uses Bayes theorem in the algorithm for training and prediction



Data Features:

Variable Name	Role	Type	Variable Name	Role	Type
ID	ID	Categorical	symmetry1	Feature	Continuous
Diagnosis	Target	Categorical	fractal_dimension1	Feature	Continuous
radius1	Feature	Continuous	radius2	Feature	Continuous
texture1	Feature	Continuous	texture2	Feature	Continuous
perimeter1	Feature	Continuous	perimeter2	Feature	Continuous
area1	Feature	Continuous	area2	Feature	Continuous
smoothness1	Feature	Continuous	smoothness2	Feature	Continuous
compactness1	Feature	Continuous	compactness2	Feature	Continuous
concavity1	Feature	Continuous	concavity2	Feature	Continuous
concave_points1	Feature	Continuous	concave_points2	Feature	Continuous



Explore the Dataset:

Loading the Dataset, View 25 Rows”`df.head(25)`” + “`df.info()`” for understanding the structure (Column Count, Column Name, Data types, And the number of non-null entries in each column) there is no Missing value!

Also, we use “`df.shape()`” for check the size.

```
Data Analysis Project.ipynb ☆
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text
df = pd.read_csv('/content/breast_cancer_wisconsin_diagnostic.csv')
df.head(25)
```

	ID	Diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	smoothness_mean	compactness_mean	concavity_mean
0	842302	M	17.990	10.38	122.80	1001.0	0.11840	0.27760	0.3001
1	842517	M	20.570	17.77	132.90	1326.0	0.08474	0.07864	0.0869
2	84300903	M	19.690	21.25	130.00	1203.0	0.10960	0.15990	0.1974
3	84346301	M	11.420	20.38	77.58	386.1	0.14250	0.28390	0.2414
4	84358402	M	20.290	14.34	135.10	1297.0	0.10030	0.13280	0.1980
5	843786	M	12.450	15.70	82.57	477.1	0.12780	0.17000	0.1578
6	844359	M	18.250	19.98	119.60	1040.0	0.09463	0.10900	0.1127
7	84458202	M	13.710	20.83	90.20	577.9	0.11890	0.16450	0.0936
8	844981	M	13.000	21.82	87.50	519.8	0.12730	0.19320	0.1859

```
+ Code + Text
[5]
9 concave_points_mean 569 non-null float64
10 symmetry_mean 569 non-null float64
11 fractal_dimension_mean 569 non-null float64
12 radius_se 569 non-null float64
13 texture_se 569 non-null float64
14 perimeter_se 569 non-null float64
15 area_se 569 non-null float64
16 smoothness_se 569 non-null float64
17 compactness_se 569 non-null float64
18 concavity_se 569 non-null float64
19 concave_points_se 569 non-null float64
20 symmetry_se 569 non-null float64
21 fractal_dimension_se 569 non-null float64
22 radius_worst 569 non-null float64
23 texture_worst 569 non-null float64
24 perimeter_worst 569 non-null float64
25 area_worst 569 non-null float64
26 smoothness_worst 569 non-null float64
27 compactness_worst 569 non-null float64
28 concavity_worst 569 non-null float64
29 concave_points_worst 569 non-null float64
30 symmetry_worst 569 non-null float64
31 fractal_dimension_worst 569 non-null float64
```

```
df.info()
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 569 entries, 0 to 568
Data columns (total 32 columns):
# Column Non-Null Count Dtype
---
0 ID 569 non-null int64
1 Diagnosis 569 non-null object
2 radius_mean 569 non-null float64
3 texture_mean 569 non-null float64
4 perimeter_mean 569 non-null float64
5 area_mean 569 non-null float64
6 smoothness_mean 569 non-null float64
7 compactness_mean 569 non-null float64
8 concavity_mean 569 non-null float64
9 concave_points_mean 569 non-null float64
10 symmetry_mean 569 non-null float64
11 fractal_dimension_mean 569 non-null float64
12 radius_se 569 non-null float64
13 texture_se 569 non-null float64
14 perimeter_se 569 non-null float64
```



Modification:

The Diagnosis column was converted from letters ('A', 'B') to numbers (1, 0) using the .map() function, indicating that 1 represents a malignant tumor and 0 represents a benign tumor.

```
[15] import pandas as pd
      from sklearn.model_selection import train_test_split
      from sklearn.naive_bayes import GaussianNB
      from sklearn.metrics import accuracy_score, classification_report

      # "B" للحميد 0، "M" 1 للخبث (إلى قيم رقمية Diagnosis تحويل عمود)
      df['Diagnosis'] = df['Diagnosis'].map({'M': 1, 'B': 0})

      # تجهيز البيانات 'ID' إسقاط عمود
      X = df.drop(columns=['ID', 'Diagnosis'])
      y = df['Diagnosis']
```



Before And After Converting from Categorical to Continuous Variable:

ID	Diagnosis	radius_m
842302	M	17.9
842517	M	20.5
84300903	M	19.6
84348301	M	11.4
84358402	M	20.2
843786	M	12.4
844359	M	18.2
84458202	M	13.7
844981	M	13.0

ID	Diagnosis
842302	1
842517	1
84300903	1
84348301	1
84358402	1
843786	1
844359	1
84458202	1
844981	1
84501001	1
845636	1
84610002	1
846226	1
846381	1
84667401	1
84799002	1
848406	1
84862001	1
849014	1
8510426	0



Analysis:

splitting the dataset into training set and test set

We evaluate the performance of a trained classification model (Naive bayes) using “accuracy_score” and “classification_report” for precision, recall and f-score.

The results show that the accuracy 97.37% Which is a good performance!

Accuracy: 97.37%
Classification Report:
precision recall f1-score support
0 0.96 1.00 0.98 71
1 1.00 0.93 0.96 43
accuracy 0.97 114
macro avg 0.98 0.97 0.97 114
weighted avg 0.97 0.97 0.97 114



Analysis:

here we Calculate the minimum and maximum values of 'radius_mean' in the dataset that we use it on the next steps.

based on the "radius_mean" using logistic regression to predict whether a tumor is cancerous or benign based on the "radius_mean" and "Diagnosis" features.

```
{x} # Calculate the minimum and maximum values of 'radius_mean' in the dataset
min_radius = df['radius_mean'].min()
max_radius = df['radius_mean'].max()

print(f"Minimum radius_mean: {min_radius}")
print(f"Maximum radius_mean: {max_radius}")

Minimum radius_mean: 6.981
Maximum radius_mean: 28.11
```

```
# Use 'radius_mean' as the feature (X) and 'Diagnosis' as the target (y)
X = df['radius_mean'].values.reshape(-1, 1)
y = df['Diagnosis'].values

# Import the logistic regression model
from sklearn.linear_model import LogisticRegression

# Create a logistic regression model and fit it to the data
logr = LogisticRegression() # Removed linear_model. from this line
logr.fit(X, y)

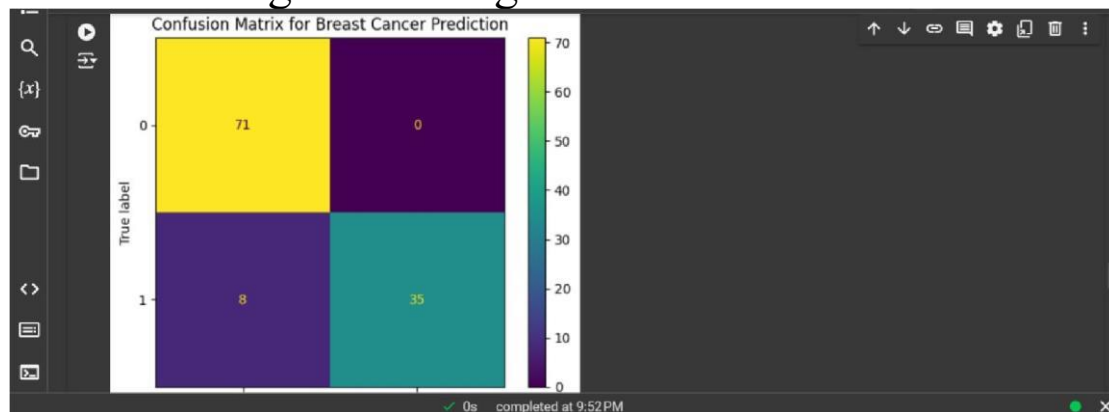
# Predict if a tumor is cancerous (1) or benign (0) for a given size, e.g., 15.0 mm (1), 14.0 mm(0)
predicted = logr.predict(np.array([14.0]).reshape(-1, 1))
print(predicted)
```

[0]



Analysis:

visualize the performance of a classification model through a confusion matrix. which is can quickly assess how well the model is performing in distinguishing between benign and malignant tumors.



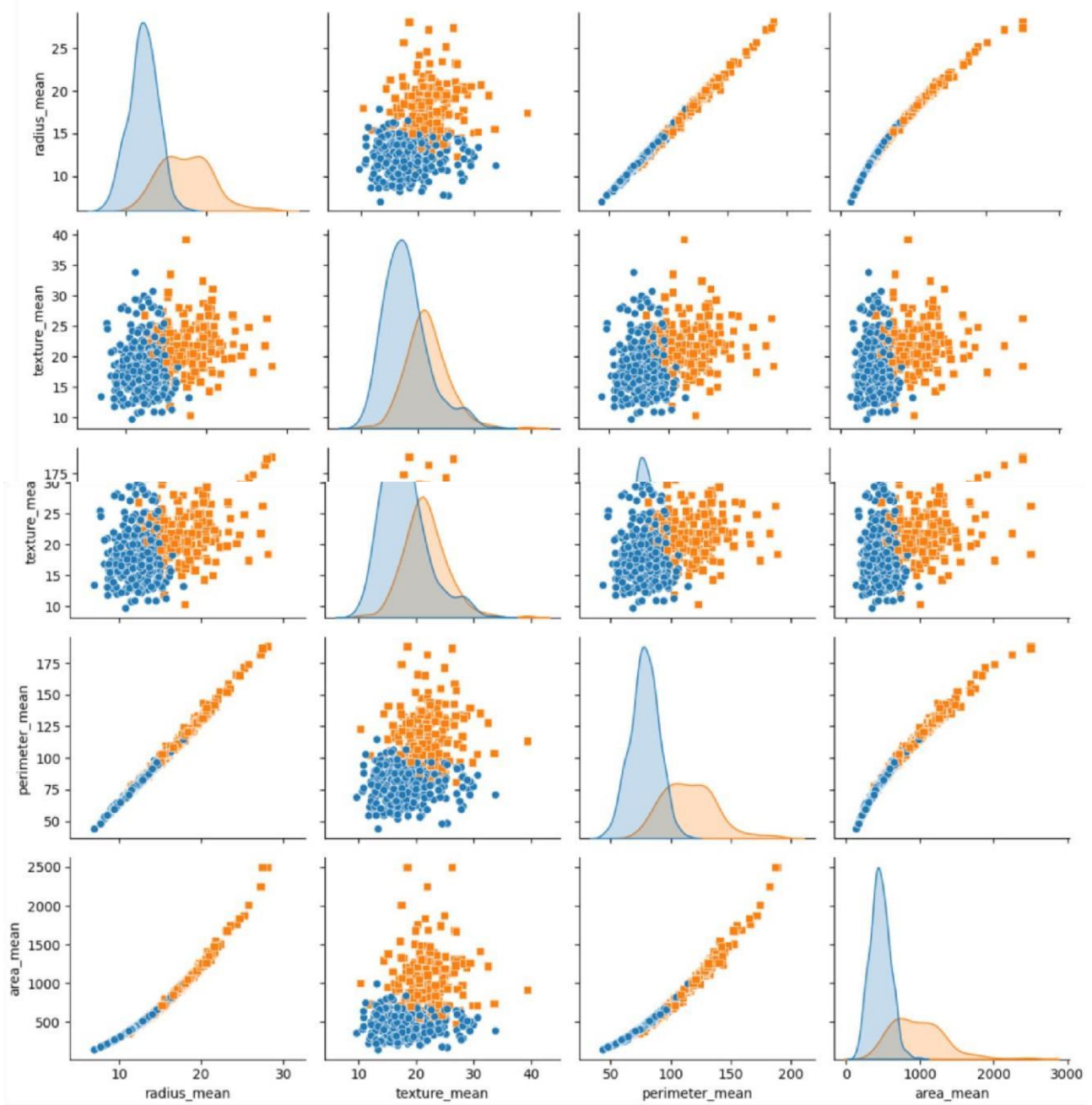


Analysis:

generates a pair plot to visualize key features in our Dataset, enabling easy examination of potential relationships and trends between features, which is essential for understanding the data and guiding further analysis or model development.



Pair Plot of Breast Cancer Dataset





جامعة أم القرى
UMM AL-QURA UNIVERSITY



جامعة أم القرى
UMM AL-QURA UNIVERSITY

