جامعة بنهـــا

كلية الحاسبات و الذكاء الاصطناعي

Orthopedic classification.

Machine Learning Project.

Team members:

Sherif Ashraf Ahmed Roshdy.

Abdelrahman Mohamed Khalil.

Mohamed Hosny Mosad.

Yahia Zakria Ebrahim.

Sobeh Salah Sobeh.

CONTENTS

Introduction

Domain of problem.

- Orthopedic relating to the branch of medicine dealing with the correction of deformities of bones or muscles.

Introduction

Conn.

- If you suffer from pain in the bones, the bone examination will show whether this pain is normal or abnormal.
- This is important because, in the event of abnormal pain, you will have to go to a specialist. After all, it may be very dangerous, such as Joints Dislocation or Fractures.
- Bone, normal or abnormal, is an important tool for detecting cancer spread in the bones from the site of the original tumor, such as breast or prostate cancer,
- It helps in repairing and preventing deformities in children at an early stage, and these deformities result from some factors that may affect the body while it is still at the embryo stage.

Introduction

Software Details and Project Objective.

Software Details:

Python (libraries) >> Pandas, SKLearn, Seaborn, Matplotlib, numpy.

Dataset >> Kaggle.

Project Objective

Used machine learning algorithms to classify a patient’s condition as normal or abnormal based on various orthopedic parameters

Dataset.

Biomechanical features of orthopedic patients | Kaggle

Dataset.

Information about dataset

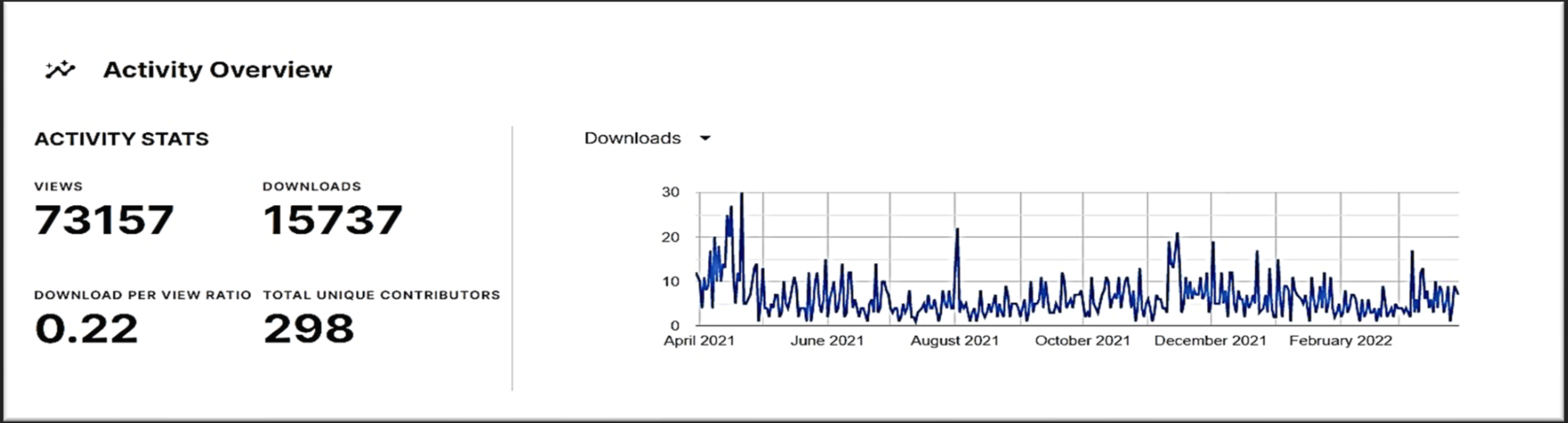
- Biomechanical features of orthopedic patients

Kaggle image.

Dataset.

Conn.

- The task consists in classifying patients as belonging to one out of two categories: Normal or Abnormal.
- Activity Overview of dataset



Dataset.

Conn.

Each patient is represented in the data set by six biomechanical attributes derived from the shape and orientation of the pelvis and lumbar spine (each one is a column)

Pelvic incidence حدوث الحوض	pelvic tilt إمالة الحوض	lumbar lordosis angle زاوية قعس قطني	sacral slope منحدر عجزي	pelvic radius نصف قطر الحوض	grade of spondylolis thesis درجة انزلاق الفقار

Dataset.

Exploratory Dataset.

- Show the first 10 rows.

	pelvic_incidence	pelvic_tilt numeric	lumbar_lordosis_angle	sacral_slope	pelvic_radius	degree_spondylolisthesis	class
0	63.027817	22.552586	39.609117	40.475232	98.672917	-0.254400	Abnormal
1	39.056951	10.060991	25.015378	28.995960	114.405425	4.564259	Abnormal
2	68.832021	22.218482	50.092194	46.613539	105.985135	-3.530317	Abnormal
3	69.297008	24.652878	44.311238	44.644130	101.868495	11.211523	Abnormal
4	49.712859	9.652075	28.317406	40.060784	108.168725	7.918501	Abnormal
5	40.250200	13.921907	25.124950	26.328293	130.327871	2.230652	Abnormal
6	53.432928	15.864336	37.165934	37.568592	120.567523	5.988551	Abnormal
7	45.366754	10.755611	29.038349	34.611142	117.270067	-10.675871	Abnormal
8	43.790190	13.533753	42.690814	30.256437	125.002893	13.289018	Abnormal
9	36.686353	5.010884	41.948751	31.675469	84.241415	0.664437	Abnormal

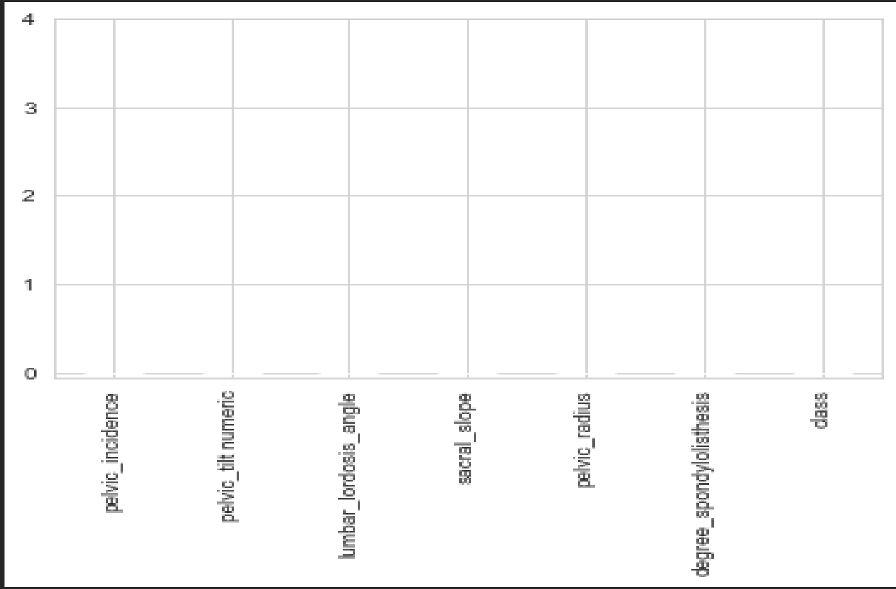
Dataset.

Conn.

- Print a concise summary of a DataFrame.
- The shape >> (1180,7)

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1180 entries, 0 to 1179
Data columns (total 7 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   pelvic_incidence          1180 non-null   float64
 1   pelvic_tilt numeric       1180 non-null   float64
 2   lumbar_lordosis_angle     1180 non-null   float64
 3   sacral_slope              1180 non-null   float64
 4   pelvic_radius             1180 non-null   float64
 5   degree_spondylolisthesis  1180 non-null   float64
 6   class                     1180 non-null   object
dtypes: float64(6), object(1)
memory usage: 64.7+ KB
```

Check the non-null values



Dataset.

Conn.

- Show descriptive statistics include those that summarize the central tendency, dispersion and shape of a dataset's distribution.

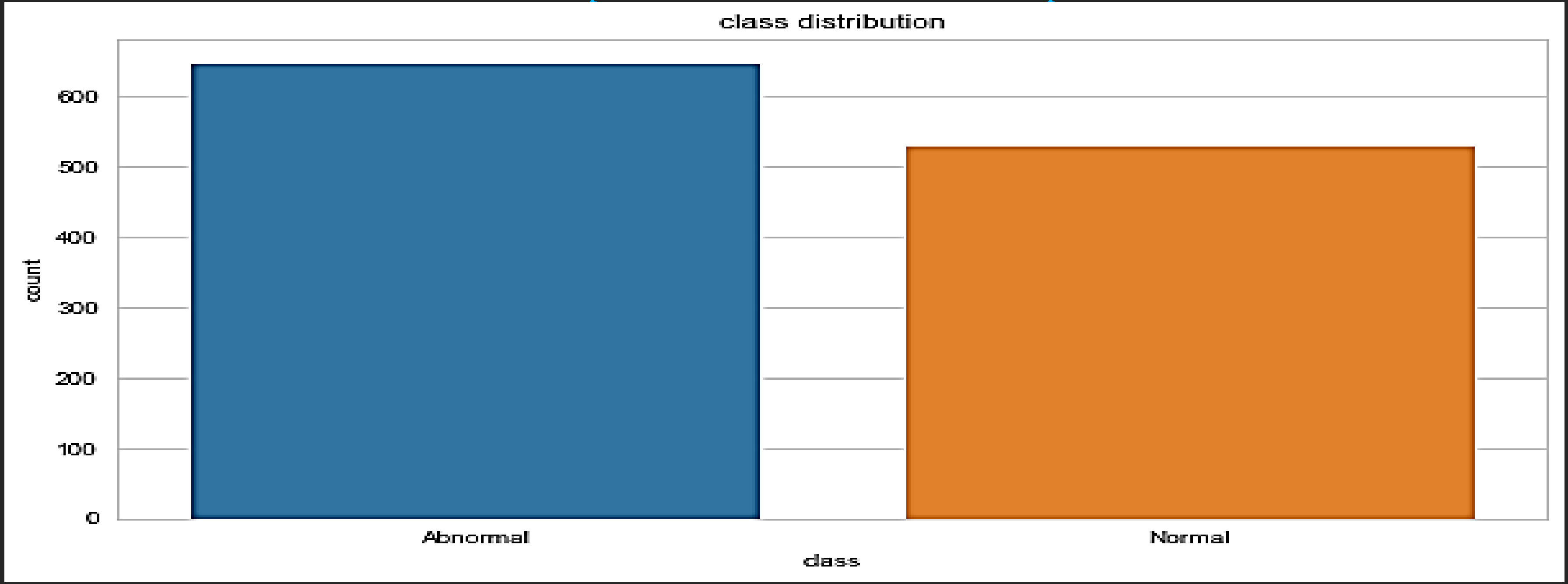
	pelvic_incidence	pelvic_tilt numeric	lumbar_lordosis_angle	sacral_slope	pelvic_radius	degree_spondylolisthesis
count	1180.000000	1180.000000	1180.000000	1180.000000	1180.000000	1180.000000
mean	60.628995	19.534941	42.966891	34.621027	115.654789	39.428121
std	12.212527	10.779453	21.270700	12.197810	14.427055	40.814087
min	26.147921	-6.554948	14.000000	13.366931	70.082575	-11.058179
25%	50.804924	12.537992	28.957819	25.065929	107.690466	11.400298
50%	61.542890	17.977784	38.926371	33.918037	116.250917	33.157646
75%	69.658921	24.822631	48.426306	43.163549	124.118877	55.995454
max	129.834041	49.431864	125.742385	121.429566	163.071041	418.543082

Dataset.

Conn.

- Represent the number of output:

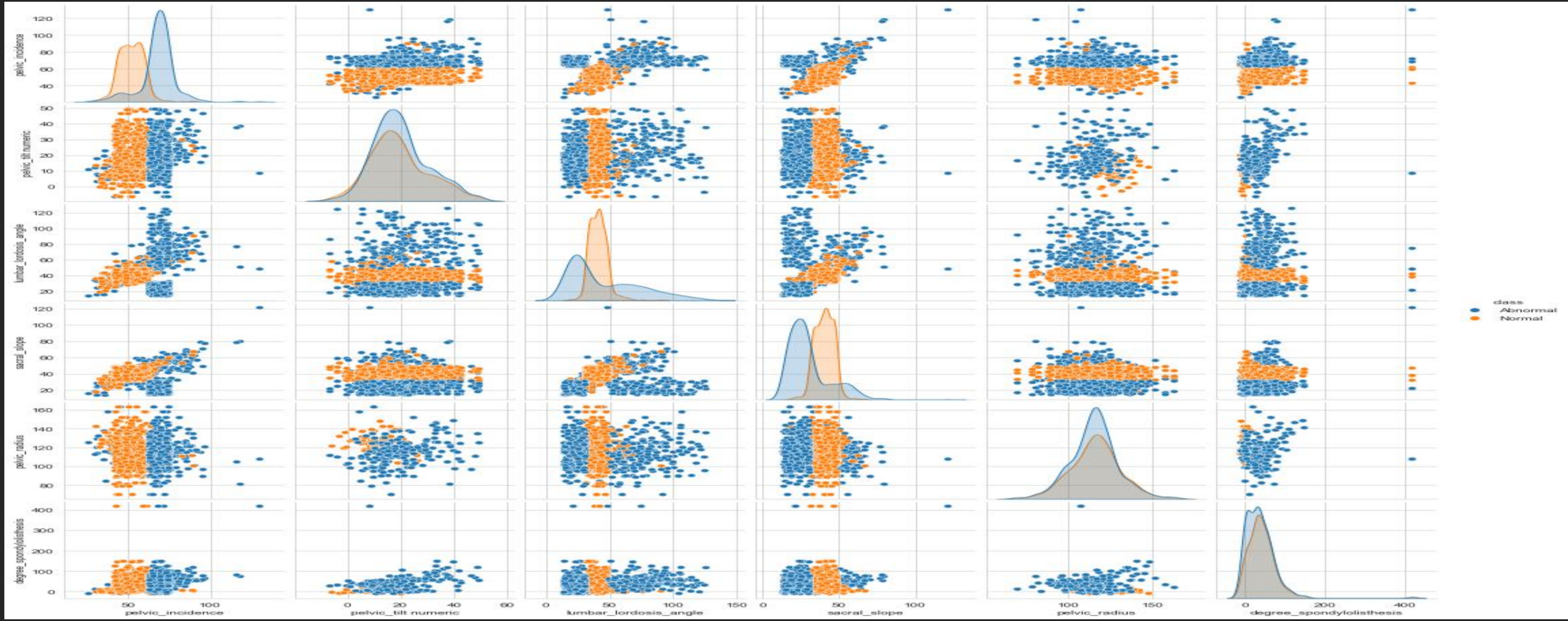
Normal (649 patients) or Abnormal (513 patients).



Dataset.

Conn.

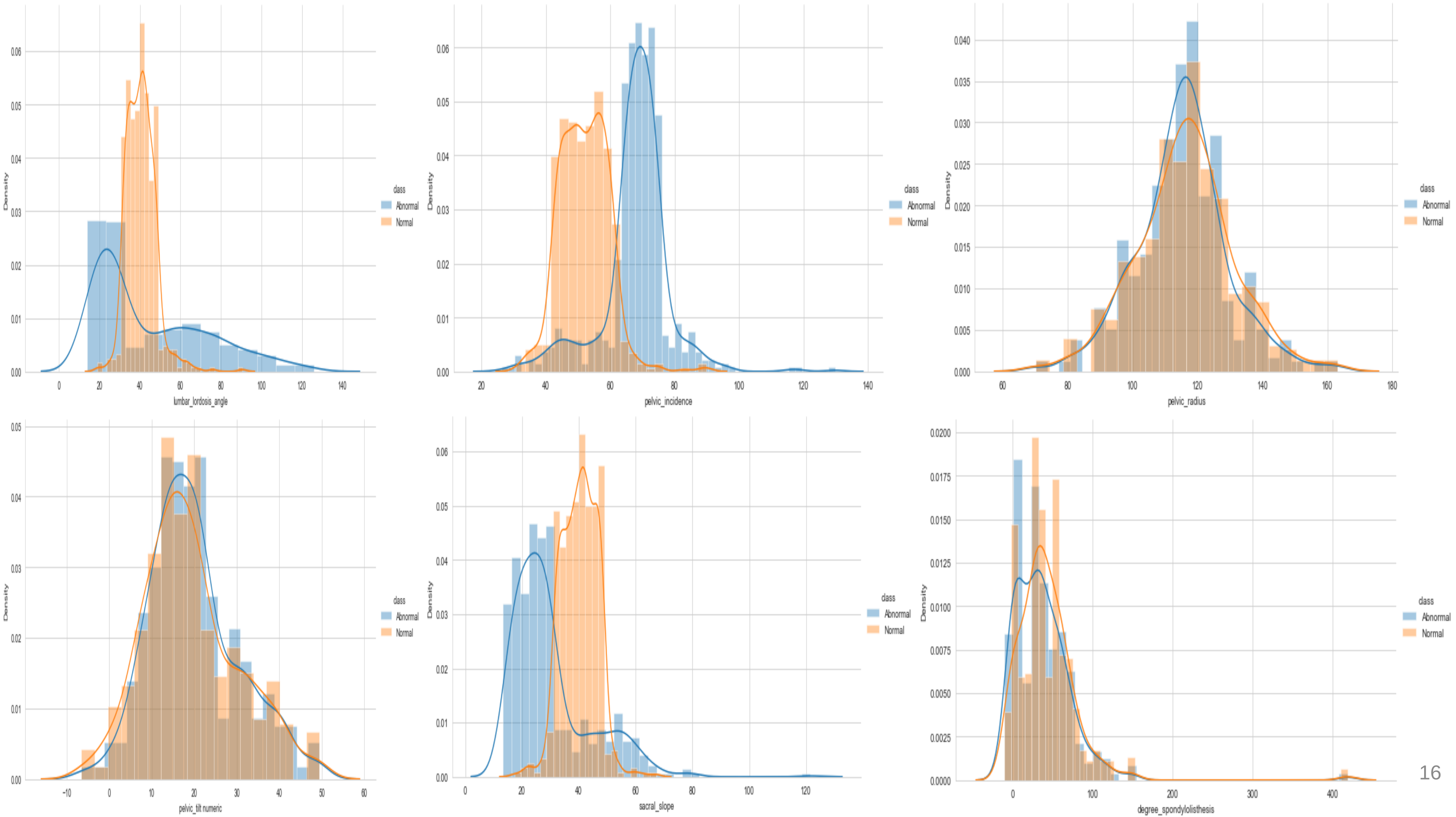
- Plot pairwise relationships in a dataset using seaborn.pairplot function.



Dataset.

Conn.

- FacetGrid class helps in visualizing distribution of one variable as well as the relationship between multiple variables separately within subsets of your dataset using multiple panels.

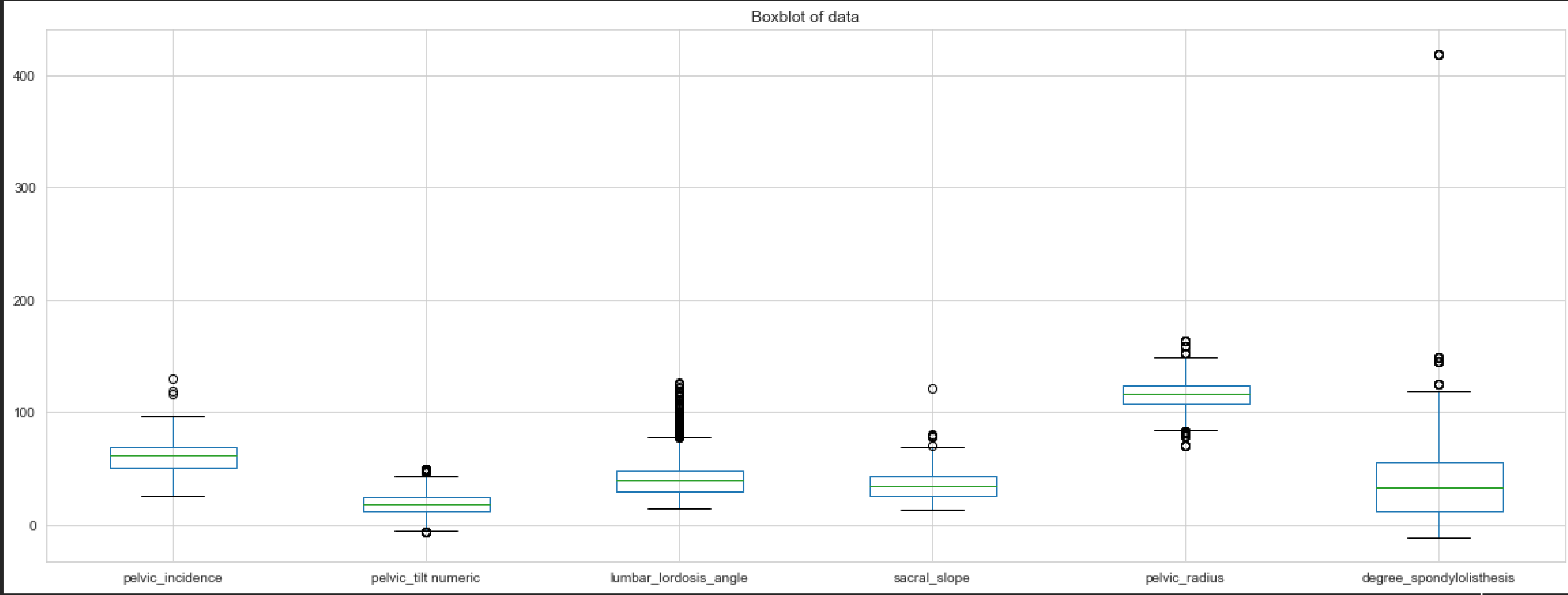


Density
lumbar_lordosis_angle
class
Abnormal
Normal
Density
pelvic_incidence
class
Abnormal
Normal
Density
pelvic_radius
class
Abnormal
Normal
Density
pelvic_tilt numeric
class
Abnormal
Normal
Density
sacral_slope
class
Abnormal
Normal
Density
degree_spondylolisthesis
class
Abnormal
Normal

Dataset.

Conn.

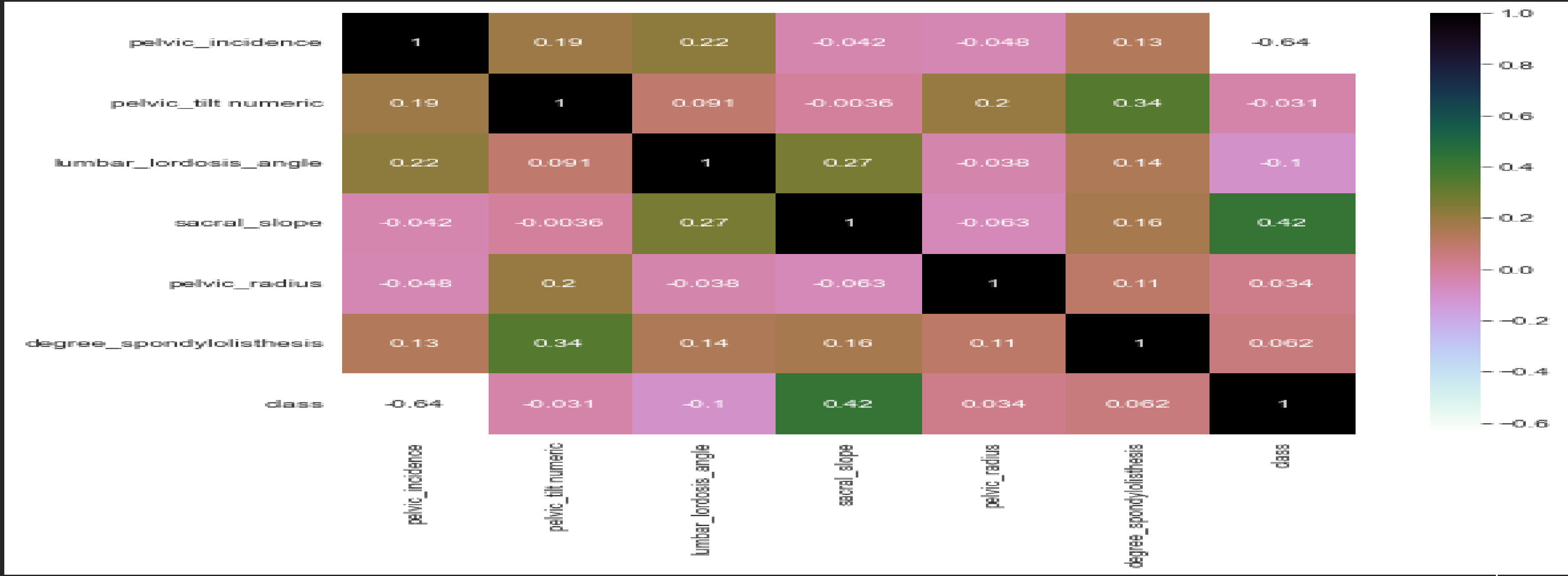
- Box plots are used to show distributions of numeric data values.



Dataset.

Conn.

- Representing the correlation between features and some of them using heatmap function



Data preprocessing for machine learning

Splitting dataset

Input data.

	pelvic_incidence	pelvic_tilt numeric	lumbar_lordosis_angle	sacral_slope	pelvic_radius	degree_spondylolisthesis
0	63.027817	22.552586	39.609117	40.475232	98.672917	-0.254400
1	39.056951	10.060991	25.015378	28.995960	114.405425	4.564259
2	68.832021	22.218482	50.092194	46.613539	105.985135	-3.530317
3	69.297008	24.652878	44.311238	44.644130	101.868495	11.211523
4	49.712859	9.652075	28.317406	40.060784	108.168725	7.918501

Output data

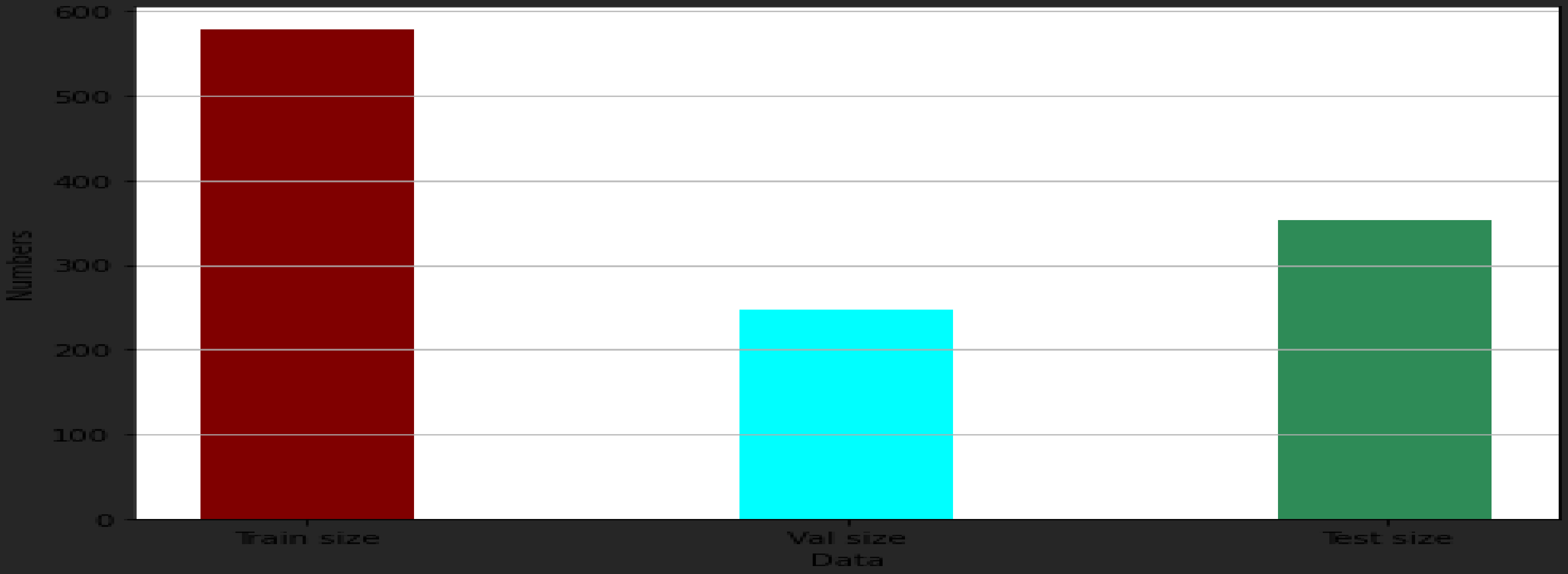
```
data_output.head()
```

```
0     0
1     0
2     0
3     0
4     0
Name: class, dtype: int64
```

Data preprocessing for machine learning

Splitting Datasets

- We need to split our dataset to 3 sets: Training, Validation, and Test data subsets.




```
X_train: (578, 6)
y_train: (578,)
-------------------------
```

```
-------------------------
X_val: (248, 6)
y_val: (248,)
-------------------------
```

```
-------------------------
X_test: (354, 6)
y_test: (354,)
```

The Algorithms used to classification

(KNN, Naive Bayes, Decision Tree)

The Algorithms used to classification

Hyper-parameter tuning

- When we build a machine learning model ,we have a number of Hyper-parameters that will determine how our model looks like.
- Changing the values of these variables will affect the accuracy of our model.
- We need to find the best hyper-parameters for our model to achieve the best accuracy.

KNN Algorithm.

The Algorithms used to classification

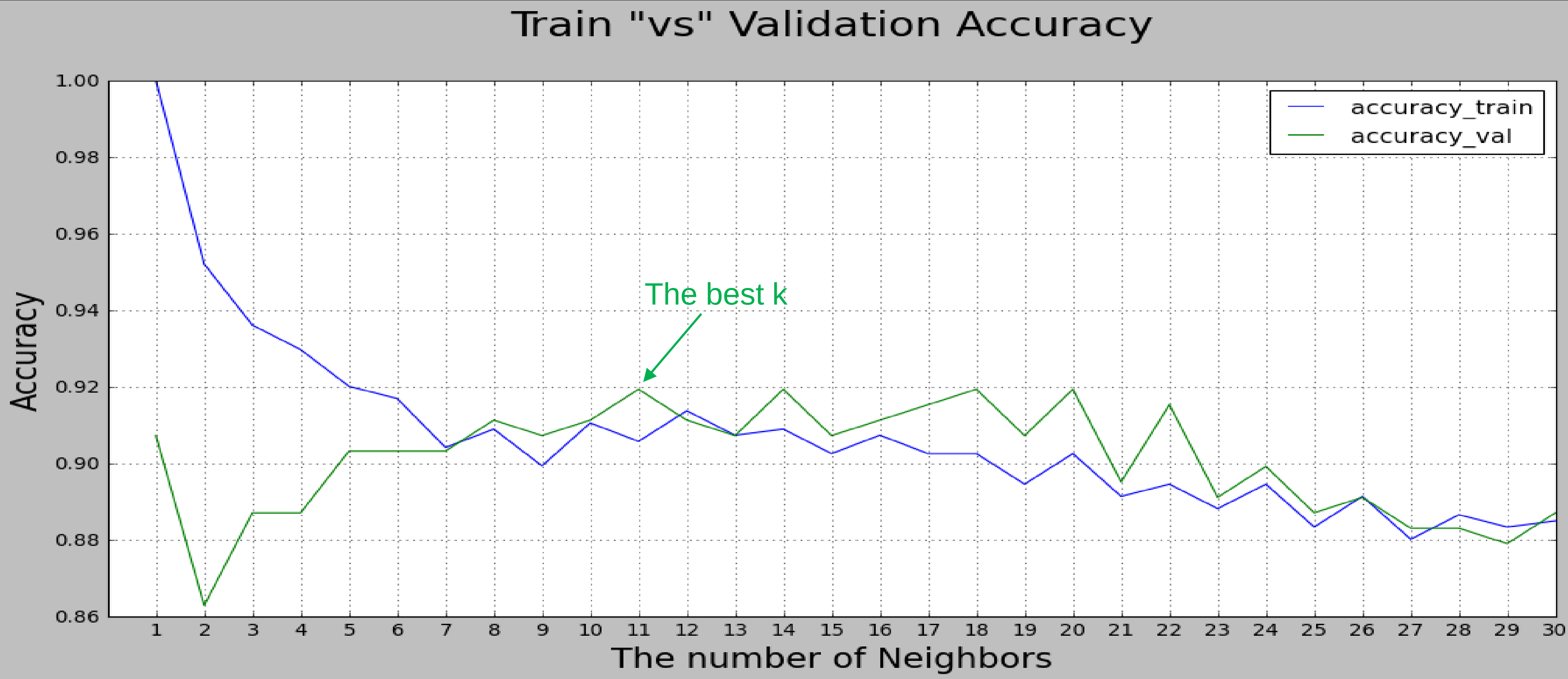
KNN Algorithm.

Hyper-parameter in KNN

- the number of neighbors
- The weights

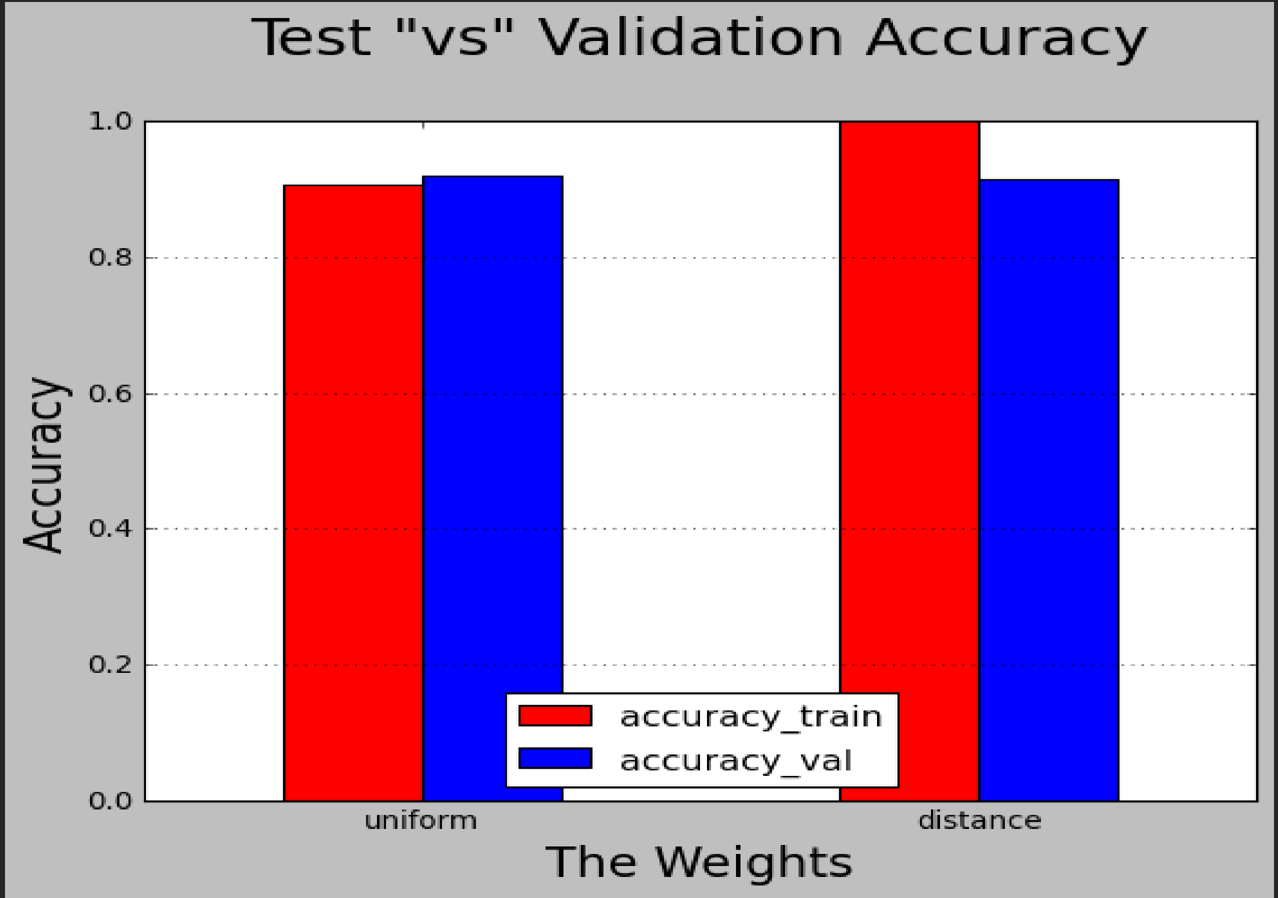
The Algorithms used to classification

KNN Algorithm.



The Algorithms used to classification

KNN Algorithm.



The best weights

	weights	accuracy_train	accuracy_val
0	uniform	0.905751	0.919355
1	distance	1.000000	0.915323

The Algorithms used to classification

KNN Algorithm.

- Then we applied the KNeighborsClassifier using the best K (11) ,weight="uniform"

```
best_knn_model=KNeighborsClassifier(n_neighbors=11,weights='uniform')
best_knn_model.fit(X_train_balanced,y_train_balaned)
y_pred_test=best_knn_model.predict(X_test)
knn_acc_final=accuracy_score(y_test,y_pred_test)
print(round(knn_acc_final,3)*100,"%")
```

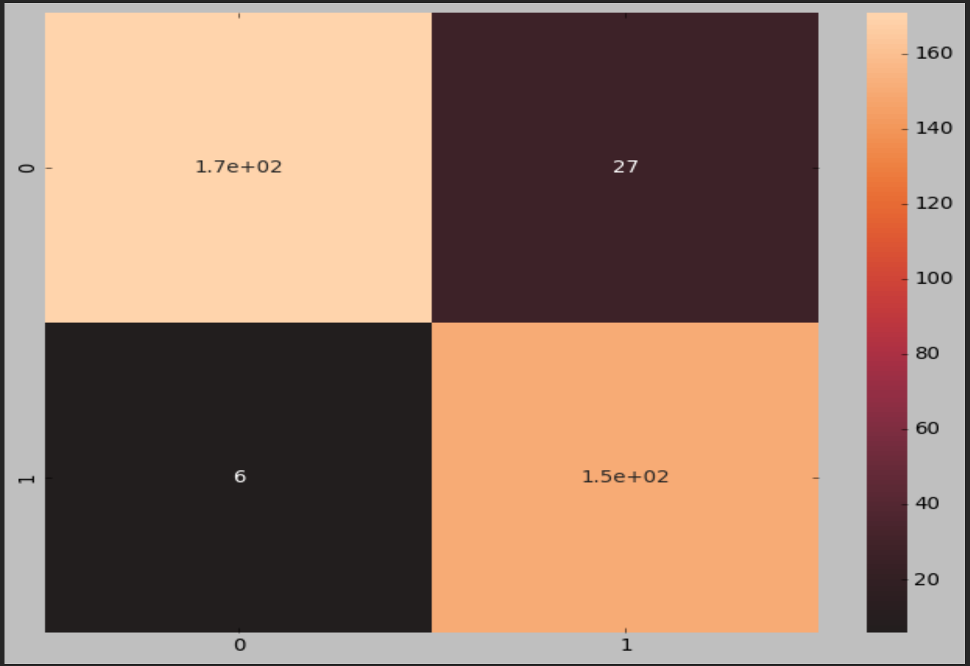
```
90.7 %
```

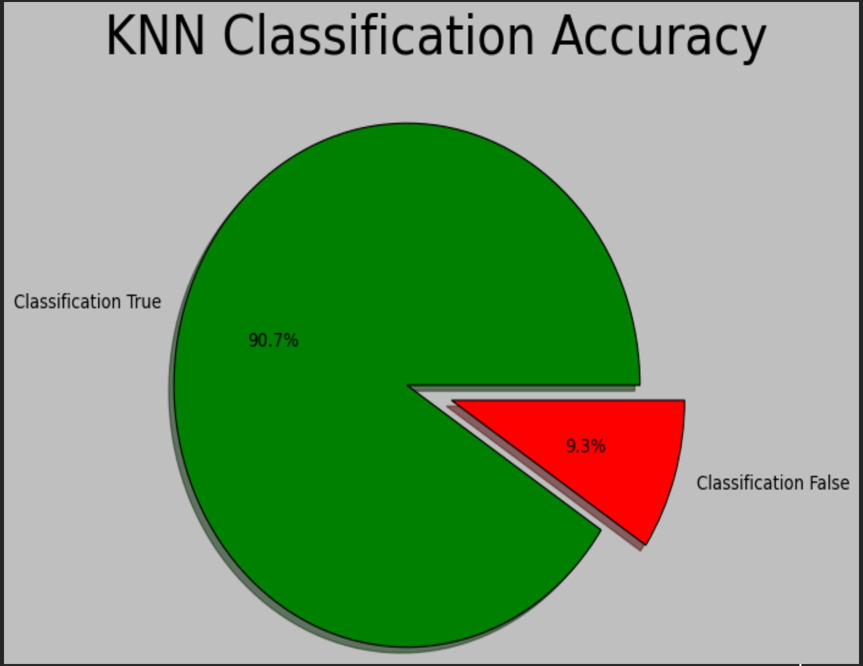
The Algorithms used to classification

KNN Algorithm.

plotted the confusion matrix of KNN

KNN Accuracy = 91%

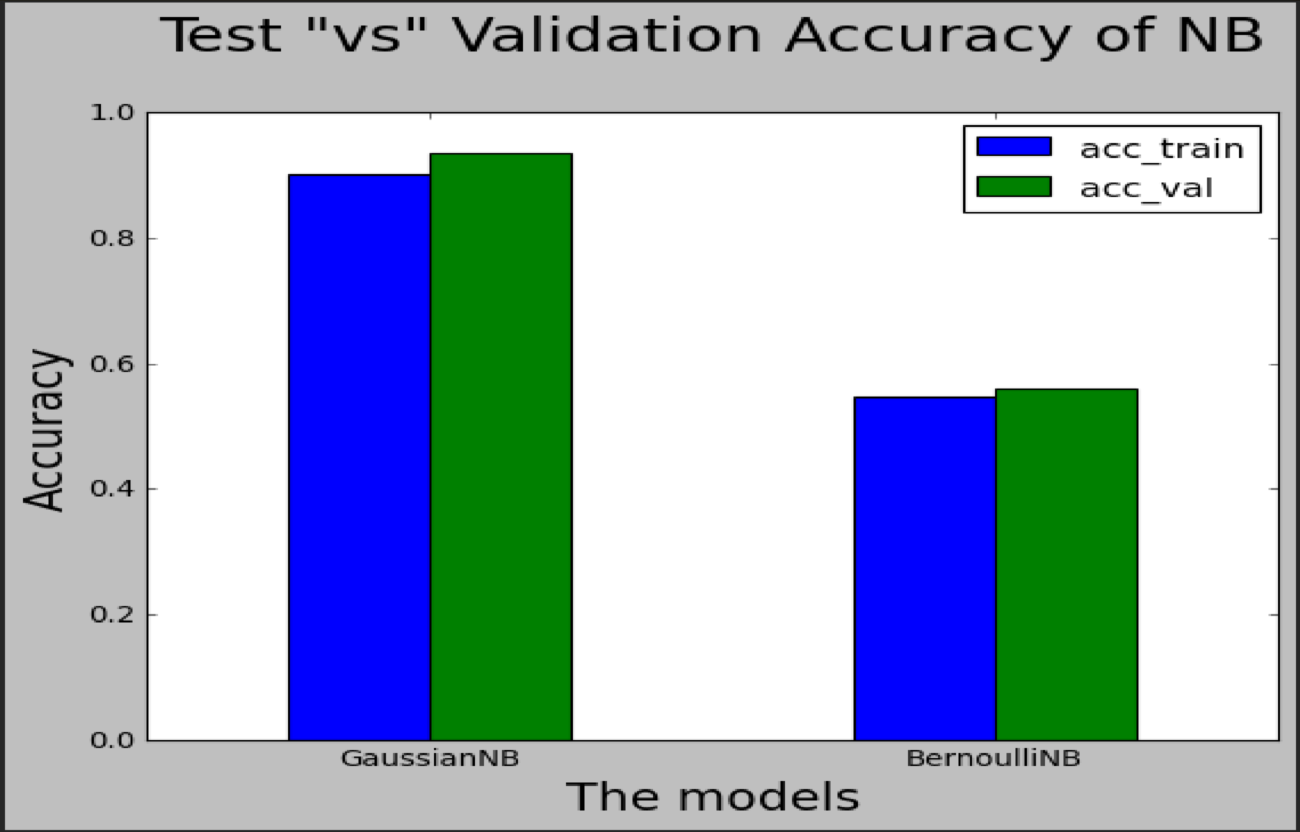




Naive Bayes Algorithm.

The Algorithms used to classification

Naive Bayes Algorithm.



GaussianNB model win

	model	acc_train	acc_val
0	GaussianNB	0.901384	0.935484
1	BernoulliNB	0.546713	0.560484

The Algorithms used to classification

Naive Bayes Algorithm.

- We applied GaussianNB() in the data

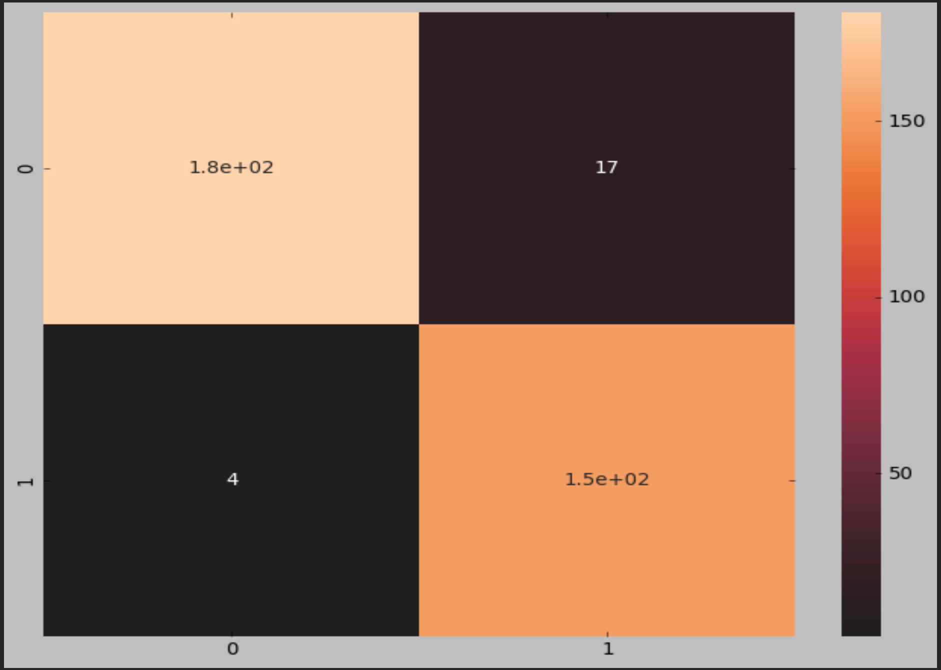
```
best_nb_model=GaussianNB()
best_nb_model.fit(X_train,y_train)
y_pred_test=best_nb_model.predict(X_test)
model_nb_final=accuracy_score(y_test,y_pred_test)
print(round(model_nb_final,3)*100,"%")
```

```
94.1 %
```

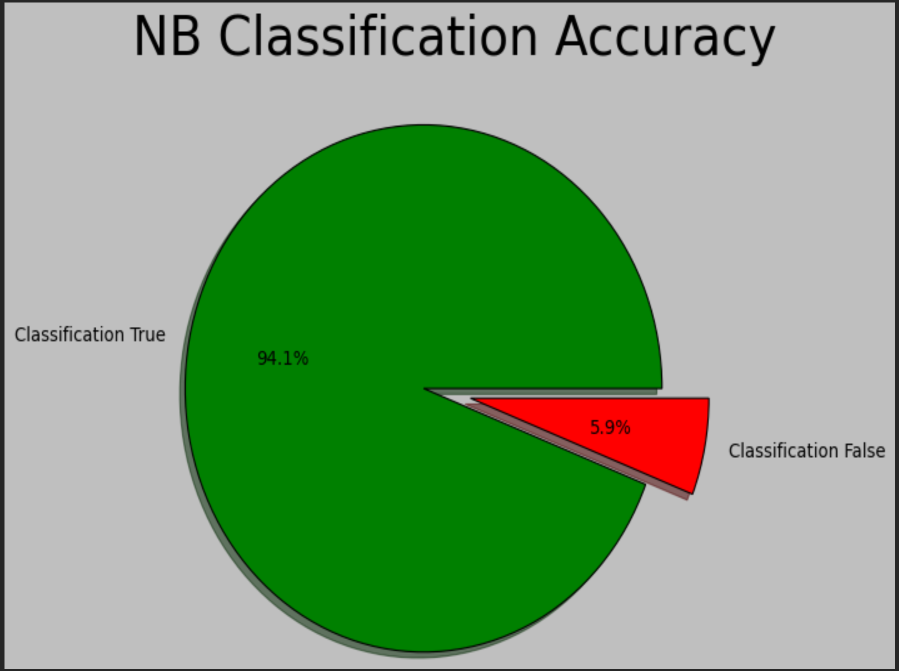
The Algorithms used to classification

Naive Bayes Algorithm.

plotted the confusion matrix of NB



NB Accuracy = 94%



Decision Tree Algorithm

The Algorithms used to classification

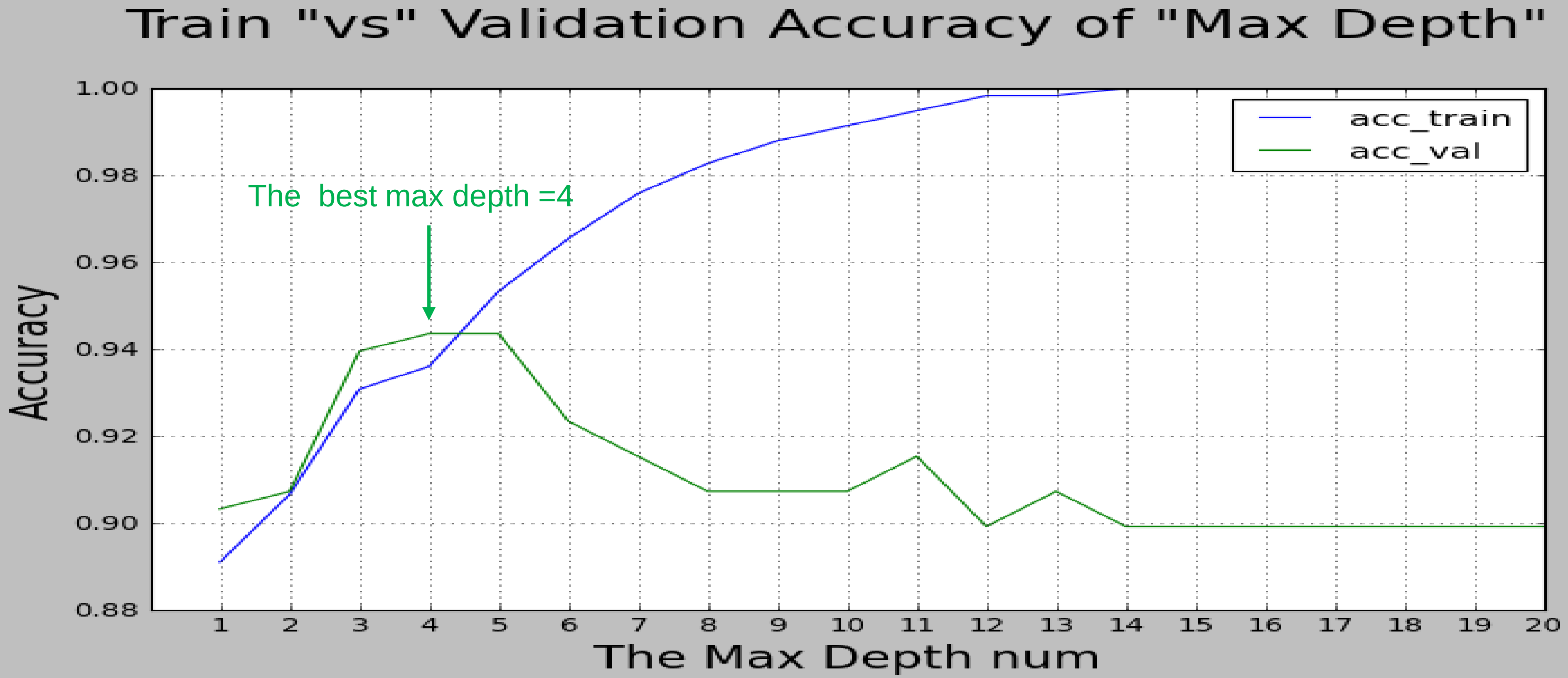
DT Algorithm.

Hyper-parameter in DT

- The max depth of tree
- The Criterion

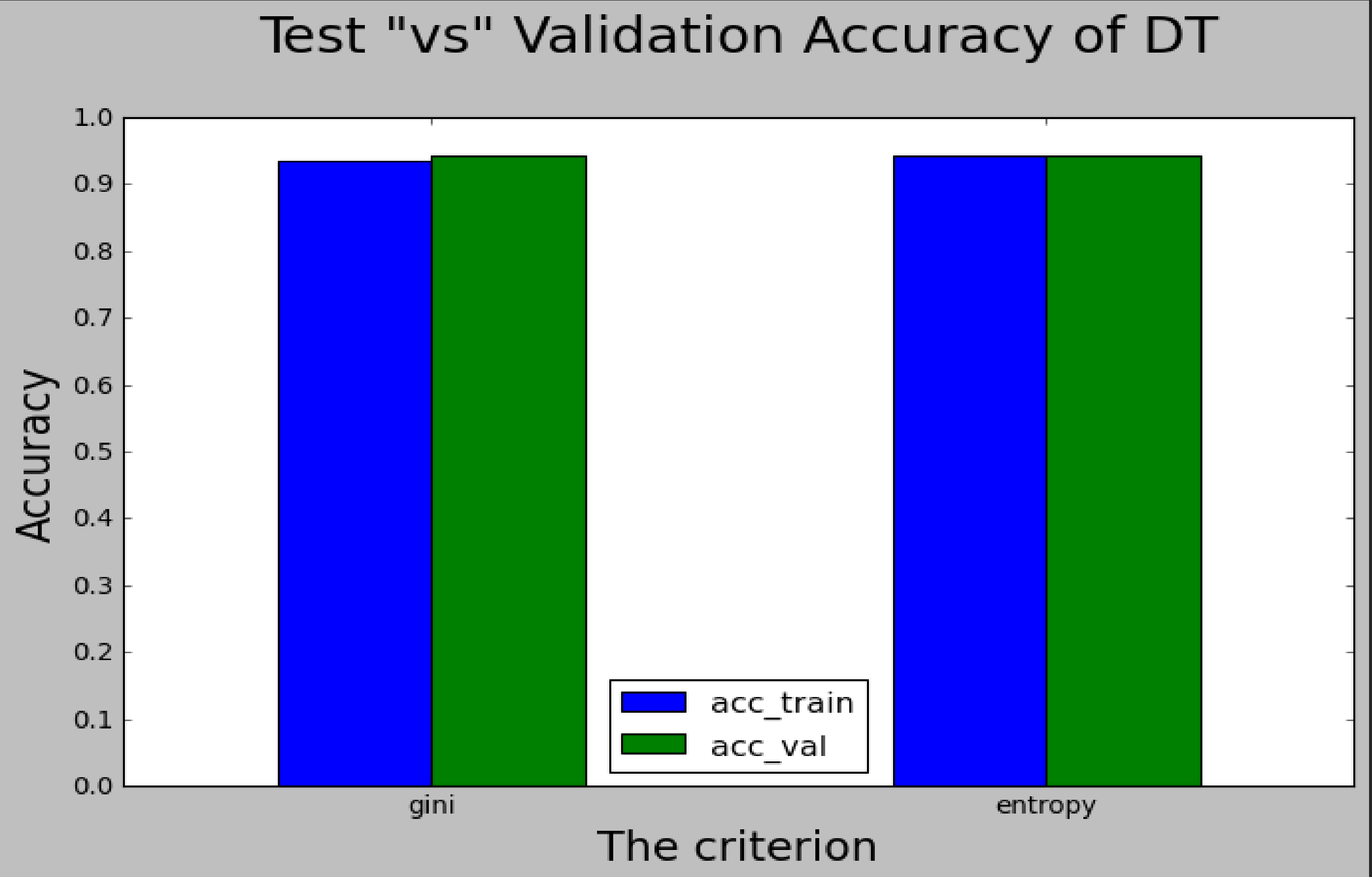
The Algorithms used to classification

Decision Tree Algorithm



The Algorithms used to classification

Decision Tree Algorithm



	criterion	acc_train	acc_val
0	gini	0.935986	0.943548
1	entropy	0.942907	0.943548

The Algorithms used to classification

Decision Tree Algorithm

- We applied the DecisionTreeClassifier with max depth=4, criterion="entropy"

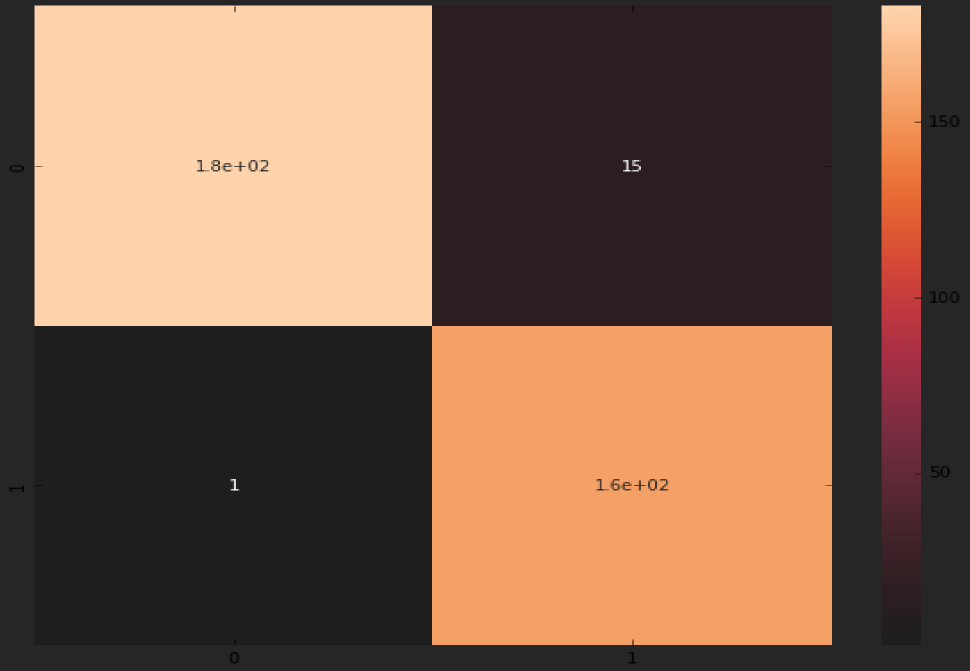
```
best_dt_model=DecisionTreeClassifier(max_depth=4,criterion='entropy',random_state=0)
best_dt_model.fit(X_train,y_train)
y_pred_test=best_dt_model.predict(X_test)
model_dt_final=accuracy_score(y_test,y_pred_test)
print(round(model_dt_final,3)*100,"%")
```

```
95.5 %
```

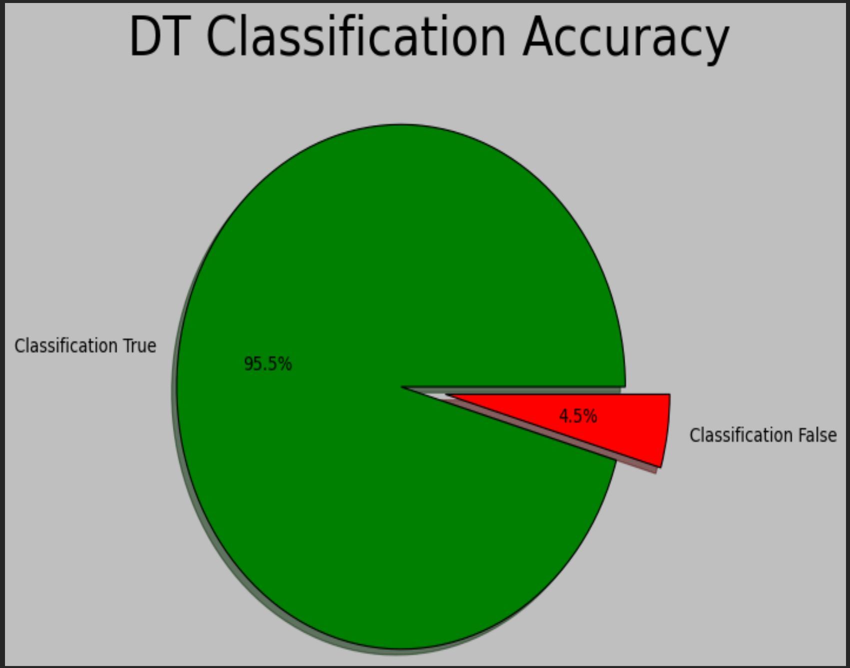
The Algorithms used to classification

Decision Tree Algorithm

plotted the confusion matrix of DT

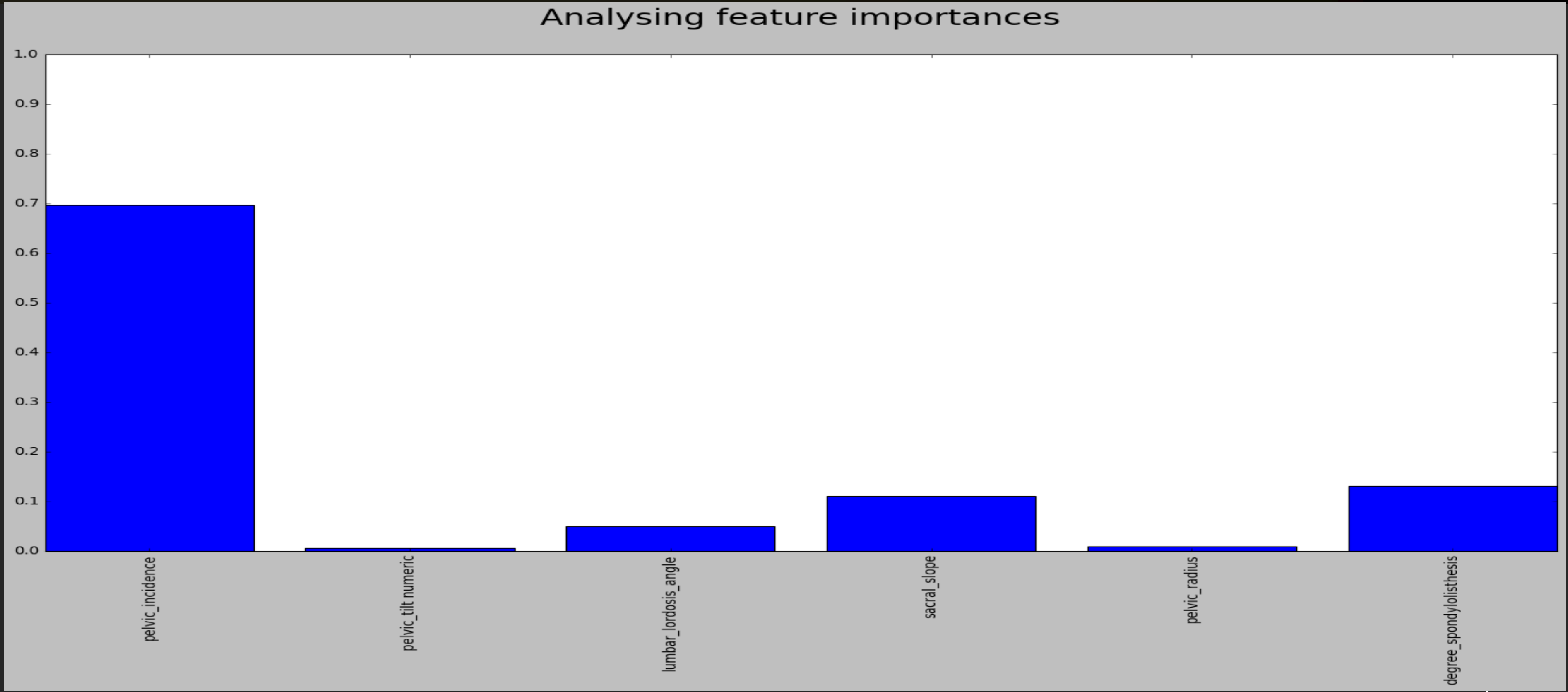


DT Accuracy = 95.5%



The Algorithms used to classification

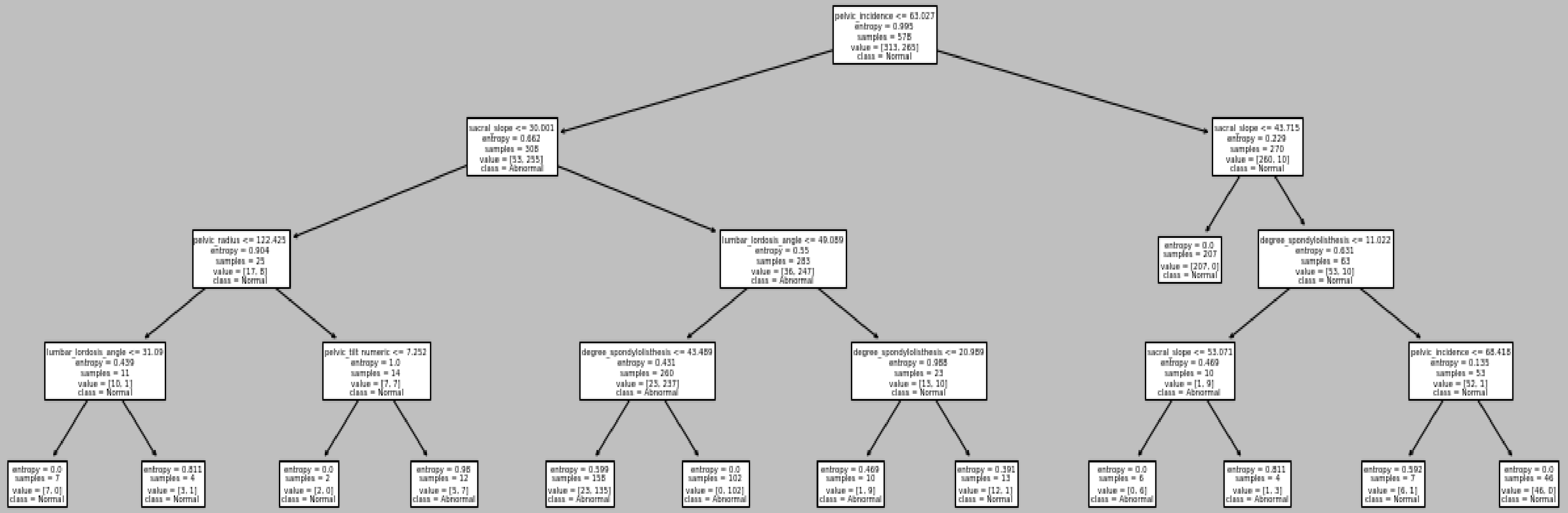
Decision Tree Algorithm



The Algorithms used to classification

Decision Tree Algorithm

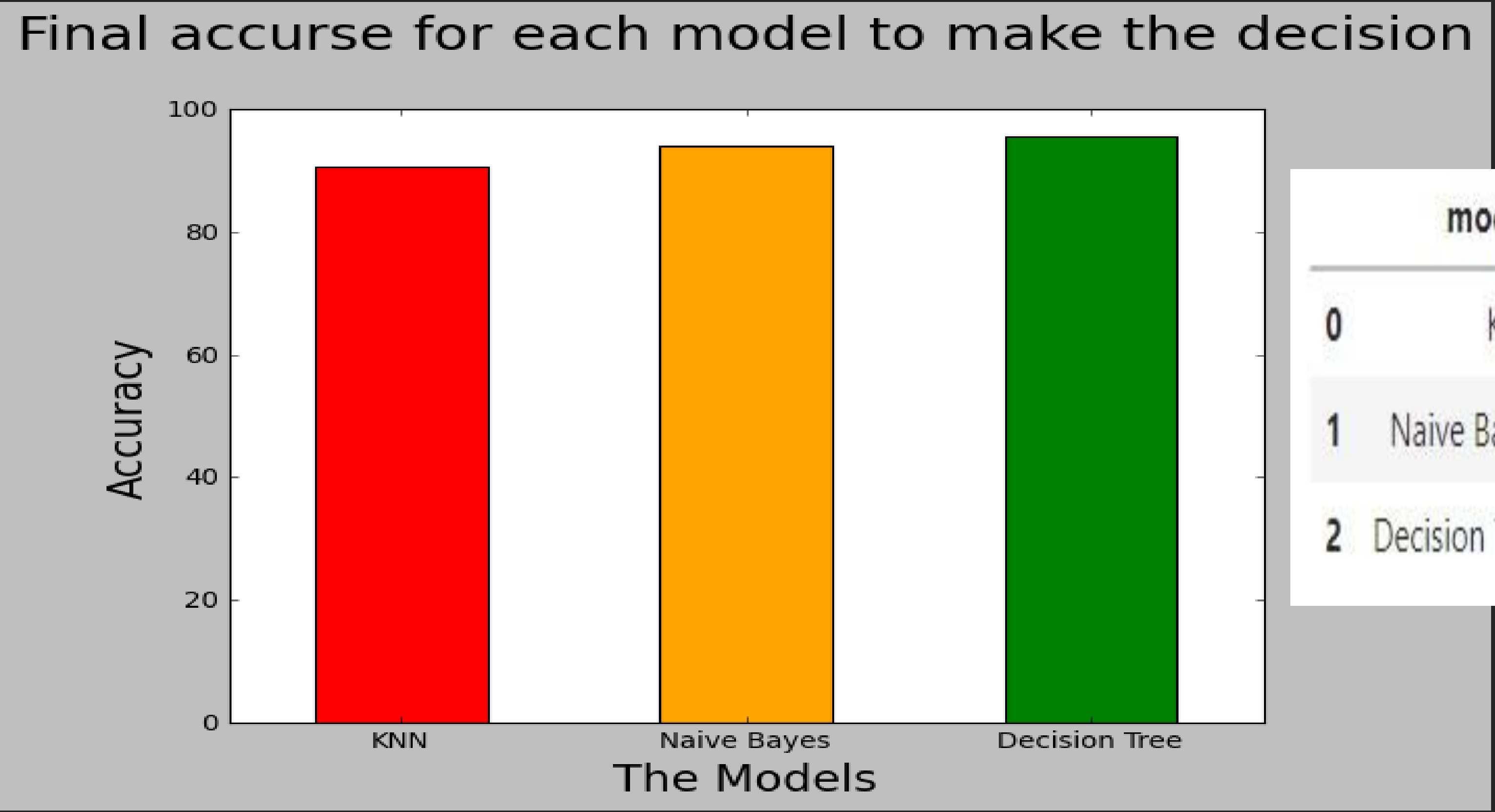
Visualizing the model



Conclusion.

Conclusion.

Compare The Accuracy Of Each Model



	models	final accurase
0	KNN	90.7
1	Naive Bayes	94.1
2	Decision Tree	95.5

Conclusion.

The Decision of solving problem.

- When a new orthopedic patient comes to the hospital, we will be able to largely classify if he is normal or abnormal by adding some new biomechanical features to the model.

- We can conclude that "Decision Tree Algorithm" works well for data because it has a high accuracy of 95.5%.

❤❤ THANK YOU! ❤❤

Any Questions?