Jeevan Kumar Banoth - 02105145

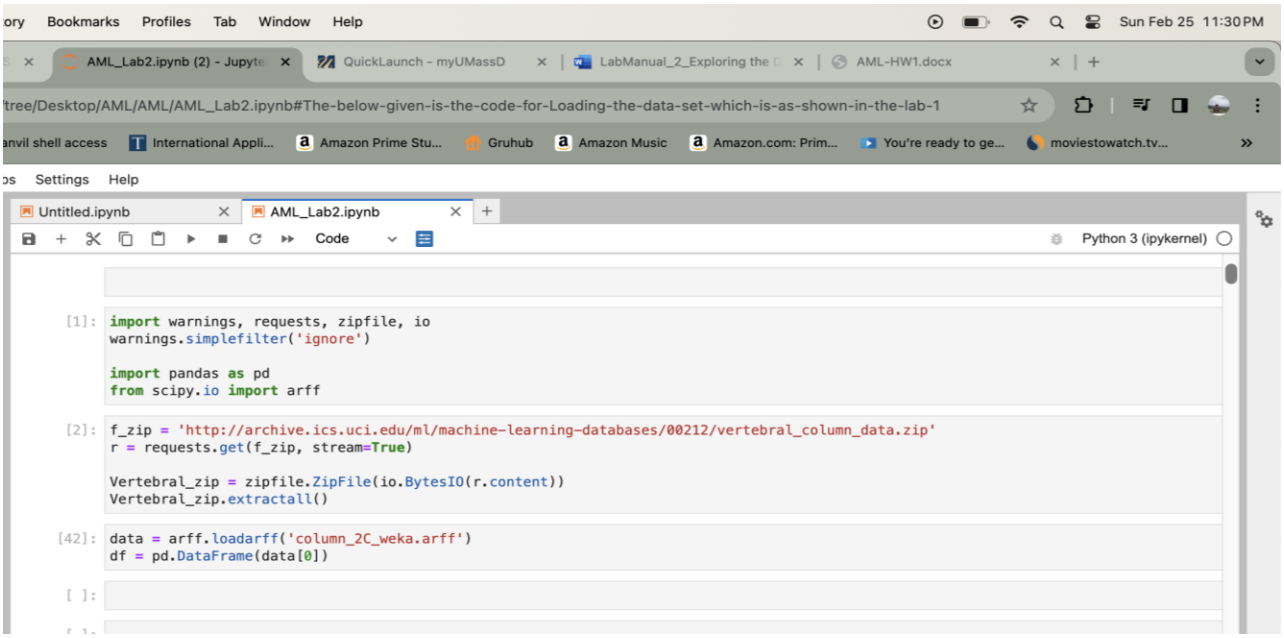
Advanced machine learning

Homework –2:

Lab setup:

The below given is the code for Loading the data set, which is as shown in the lab 1. Because the solution is split across several labs in this module, we have imported that data set I have imported that as instructed in the Lab manual

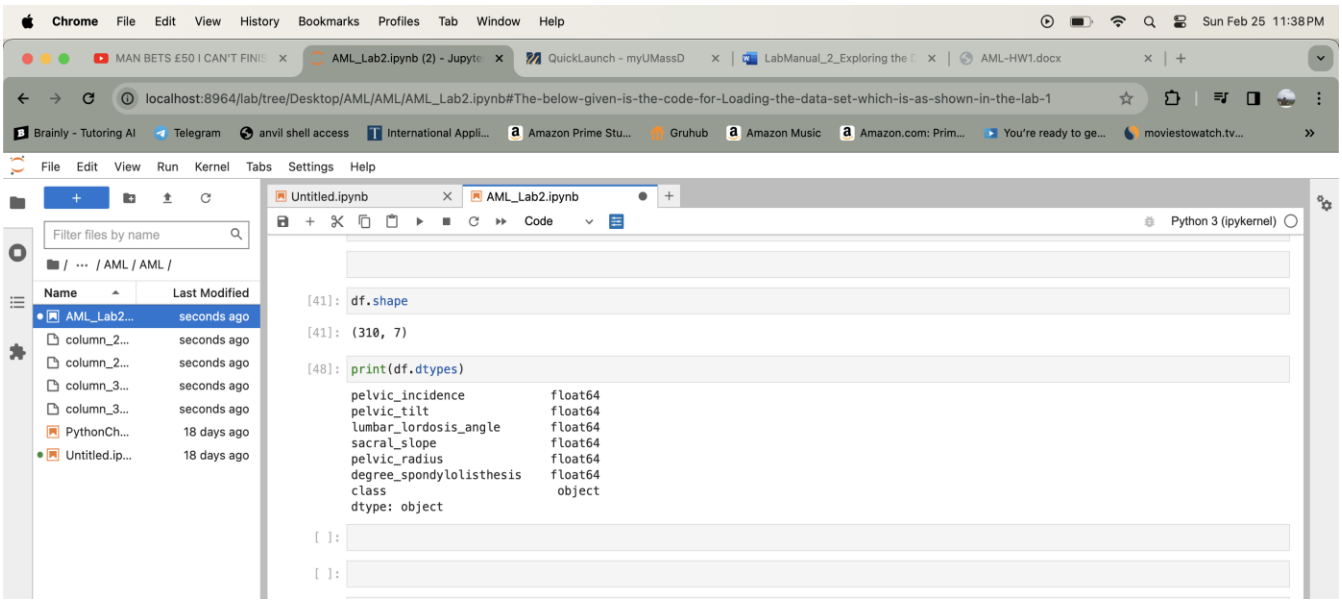
Importing the data:



So now the data has been imported we can start Lab-2 from the steps below.

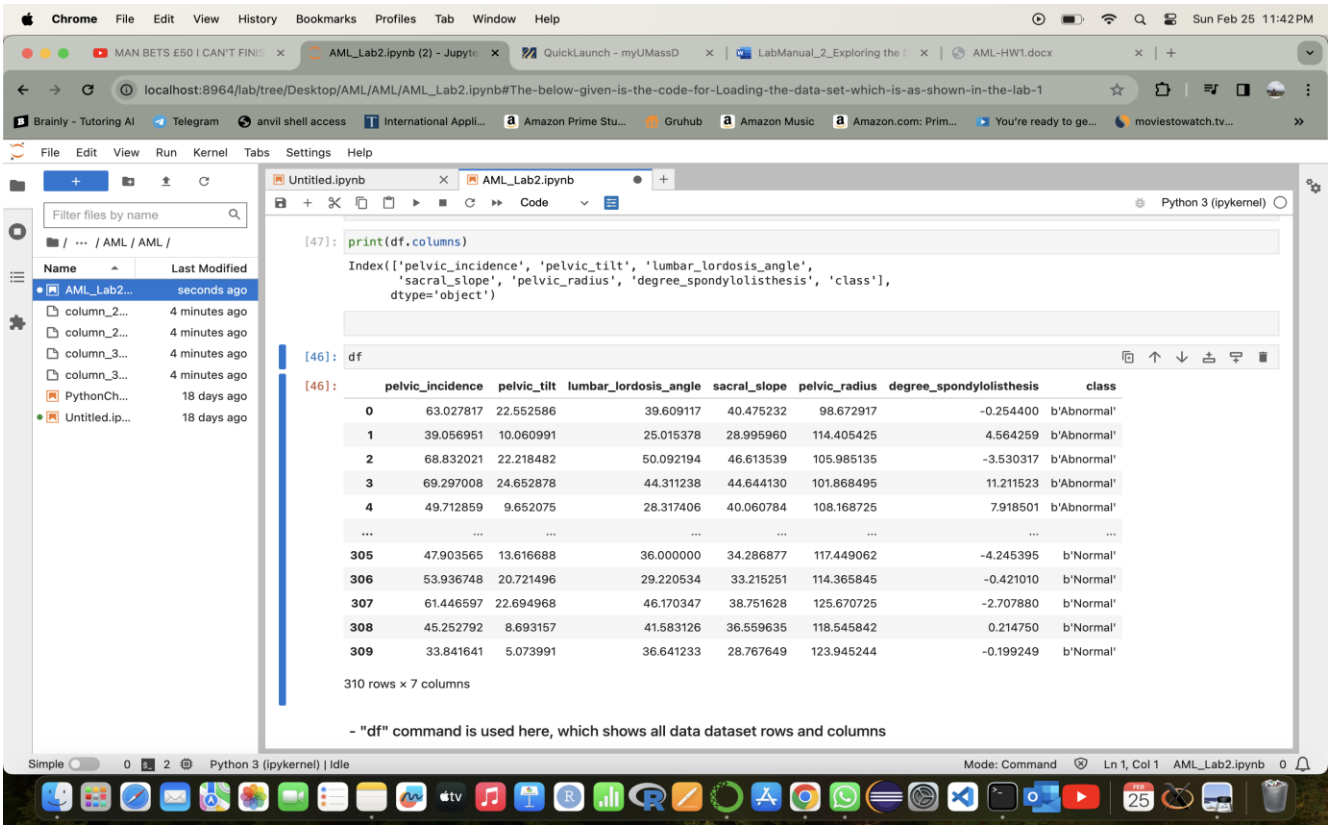
Step 1: Exploring the data

First, we are using “shape” to determine the number of rows and columns in the dataset. As we have given “df” to the name of dataset, we are using “df.shape” command for determining the number of rows and columns in the dataset.



As we can see here it shows the number of rows as 310 and the number of columns 7.

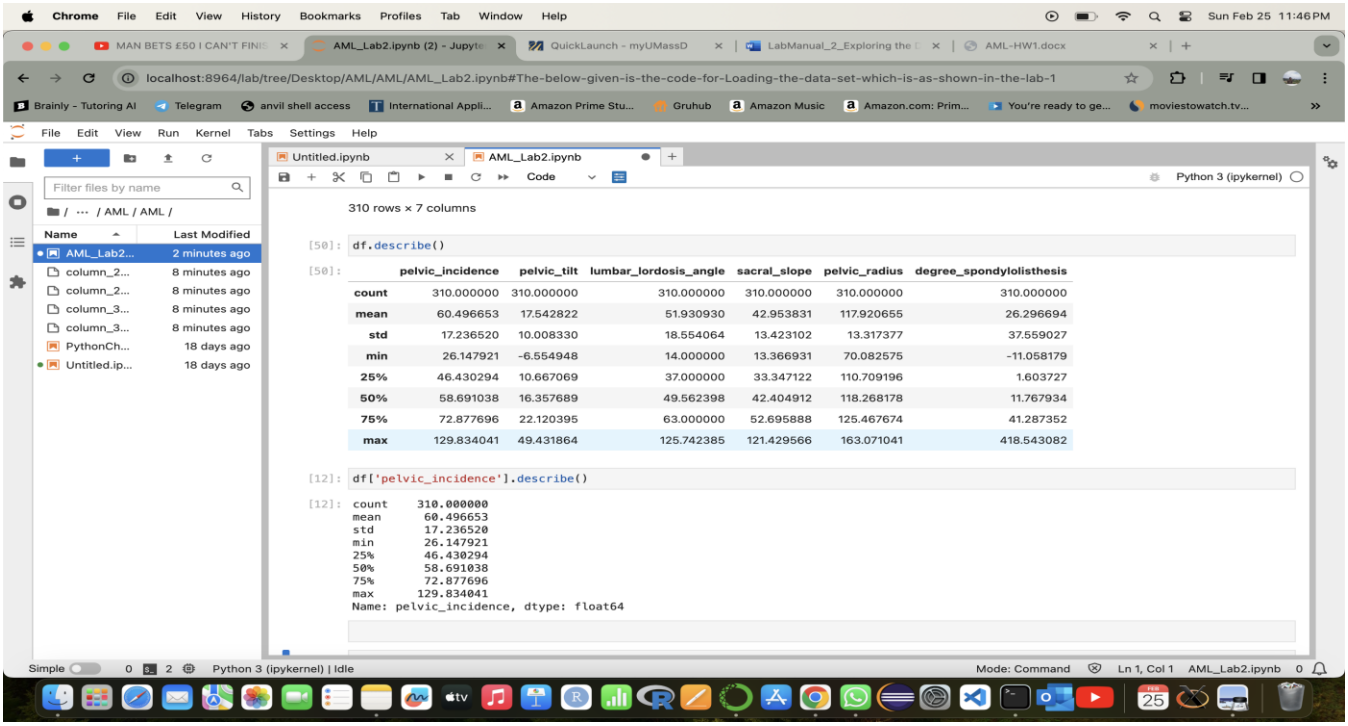
Here, dtypes shows the types of data that we have in the dataset, Here we are using the df.dtypes command, which shows types of data present in that.



→ Columns command is used for showing the number of columns present in the dataset.

→ So, here we are using “df.columns” command which will represent the number of columns.

→ “df” command is used here, which shows all data dataset rows and columns.

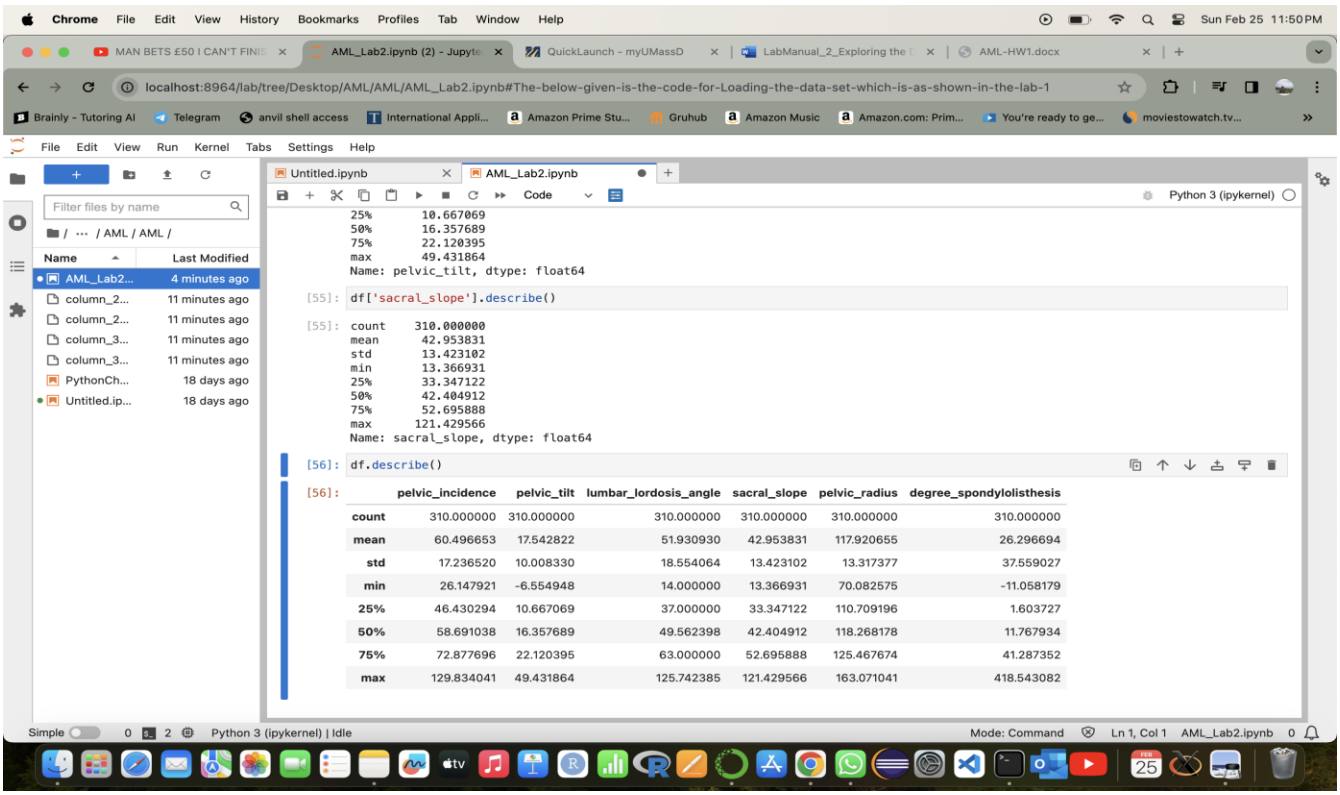


	pelvic_incidence	pelvic_tilt	lumbar_lordosis_angle	sacral_slope	pelvic_radius	degree_spondylolisthesis
count	310.000000	310.000000	310.000000	310.000000	310.000000	310.000000
mean	60.496653	17.542822	51.930930	42.953831	117.920655	26.296694
std	17.236520	10.008330	18.554064	13.423102	13.317377	37.559027
min	26.147921	-6.554948	14.000000	13.366931	70.082575	-11.058179
25%	46.430294	10.667069	37.000000	33.347122	110.709196	1.603727
50%	58.691038	16.357689	49.562398	42.404912	118.268178	11.767934
75%	72.877696	22.120395	63.000000	52.695888	125.467674	41.287352
max	129.834041	49.431864	125.742385	121.429566	163.071041	418.543082

→ “df.describe()” command used to describe their statistics of entire data set.

→ For looking at their statistics of each row we can use the name of that row and the describe () function so that it will display the actual functionalities of that respective row.

→ Below are the few different codes which describe their statistics of respective rows.



→ By changing the code as follows we can get their statistics.

Features with potential issues:

degree_spondylolisthesis:

Large standard deviation (37.5) compared to the mean (2.6) suggests a wide spread of values.

Minimum value (-11.06) and maximum value (418.54) indicate potential outliers.

Features with possible correlations:

pelvic_incidence, pelvic_tilt, sacral_slope: These features are all related to the pelvic alignment and might show correlations.

pelvic_tilt, lumbar_lordosis_angle: Pelvic tilt and lumbar lordosis often compensate for each other, potentially showing negative correlation.

Distribution visualizations:

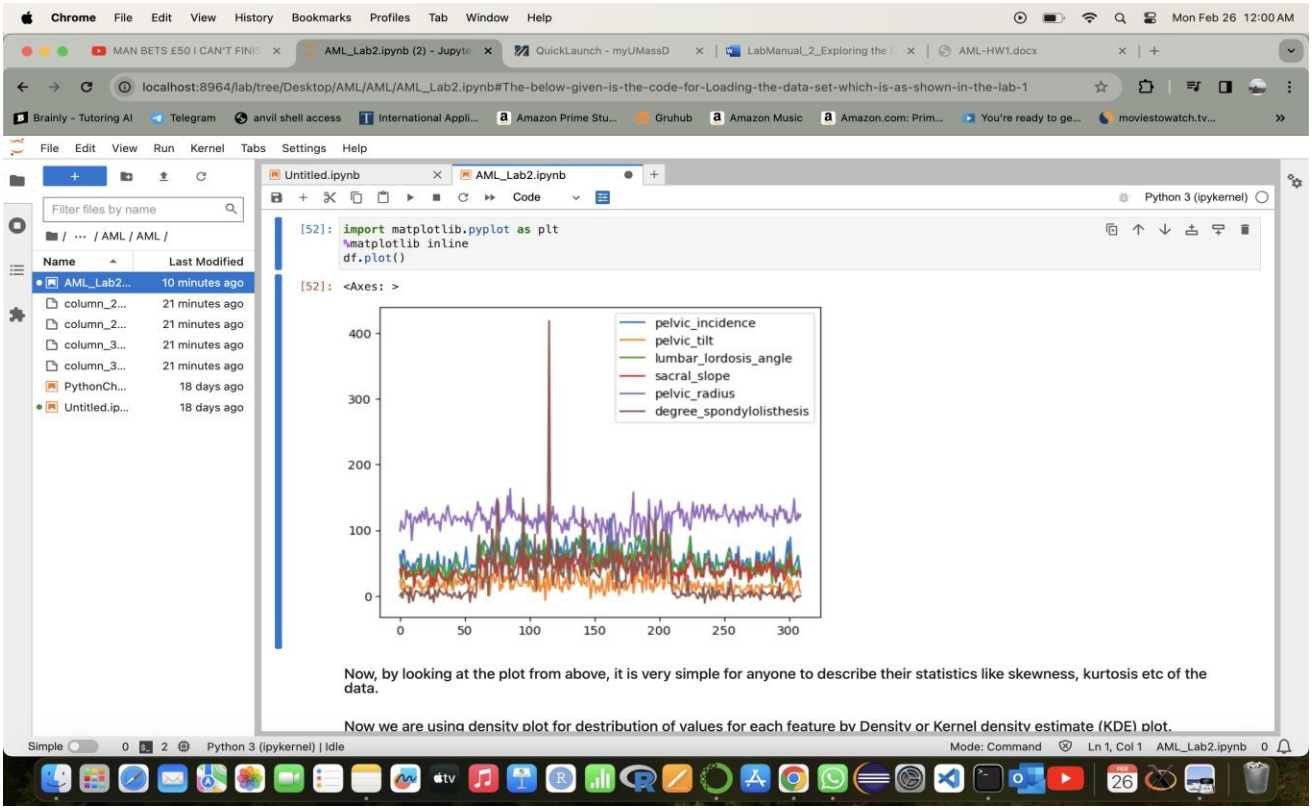
Histograms or boxplots can better visualize distributions and identify outliers.

Correlation matrix:

Calculating and visualizing the correlation matrix would reveal potential relationships between features.

As it is not always easy to make the observations based on their numbers, so here we are going with plotting them based on the given data.

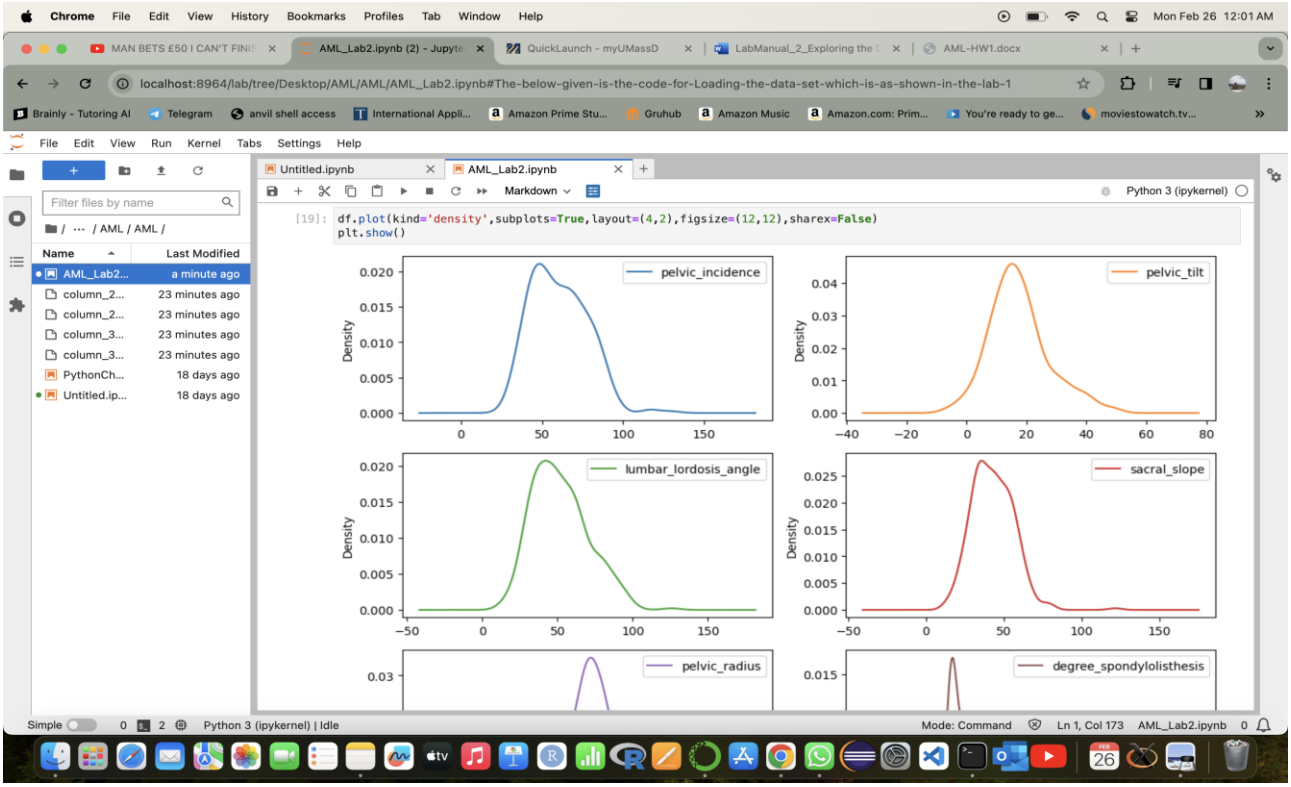
The below code is for plotting the dataset:

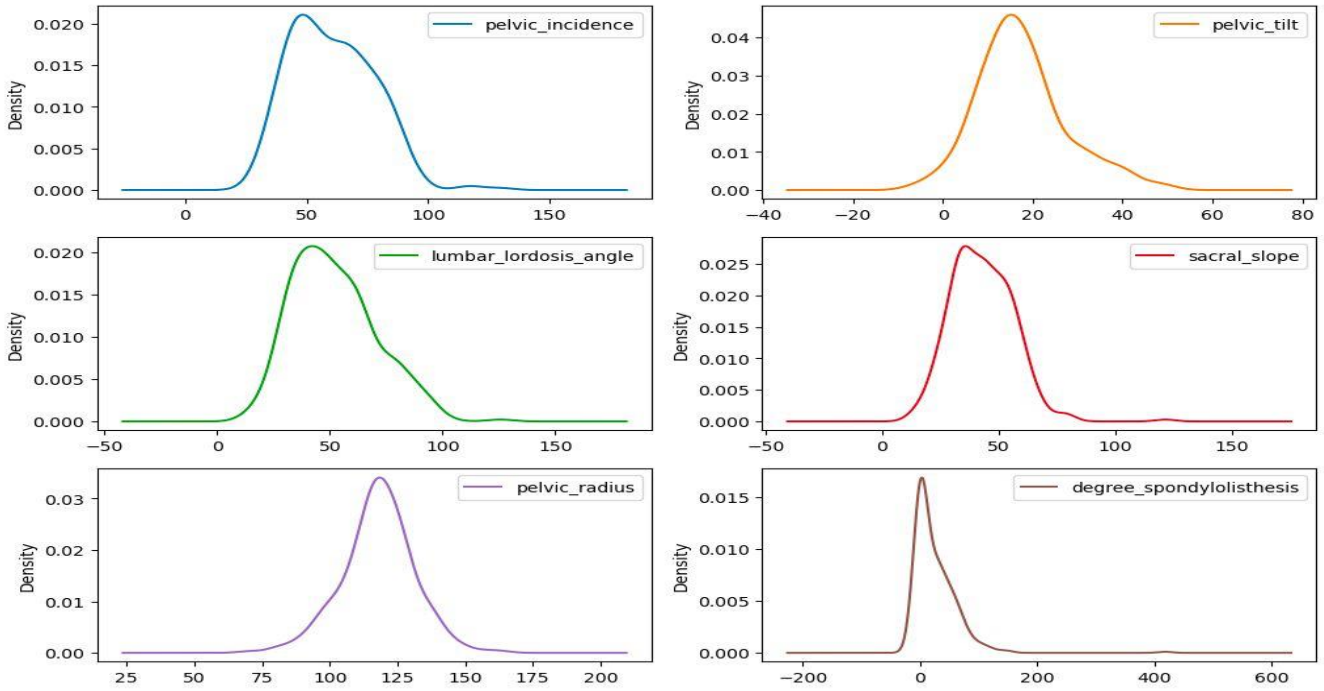


Now, by looking at the plot from above, it is very simple for anyone to describe their statistics like skewness, kurtosis etc of the data.

Now we are using density plot for destribution of values for each feature by Density or Kernel density estimate (KDE) plot.

By performing this code, we can now get the different plots for each function or row as below. This makes it easier for one to describe or differentiate them.





Here are a few observations:

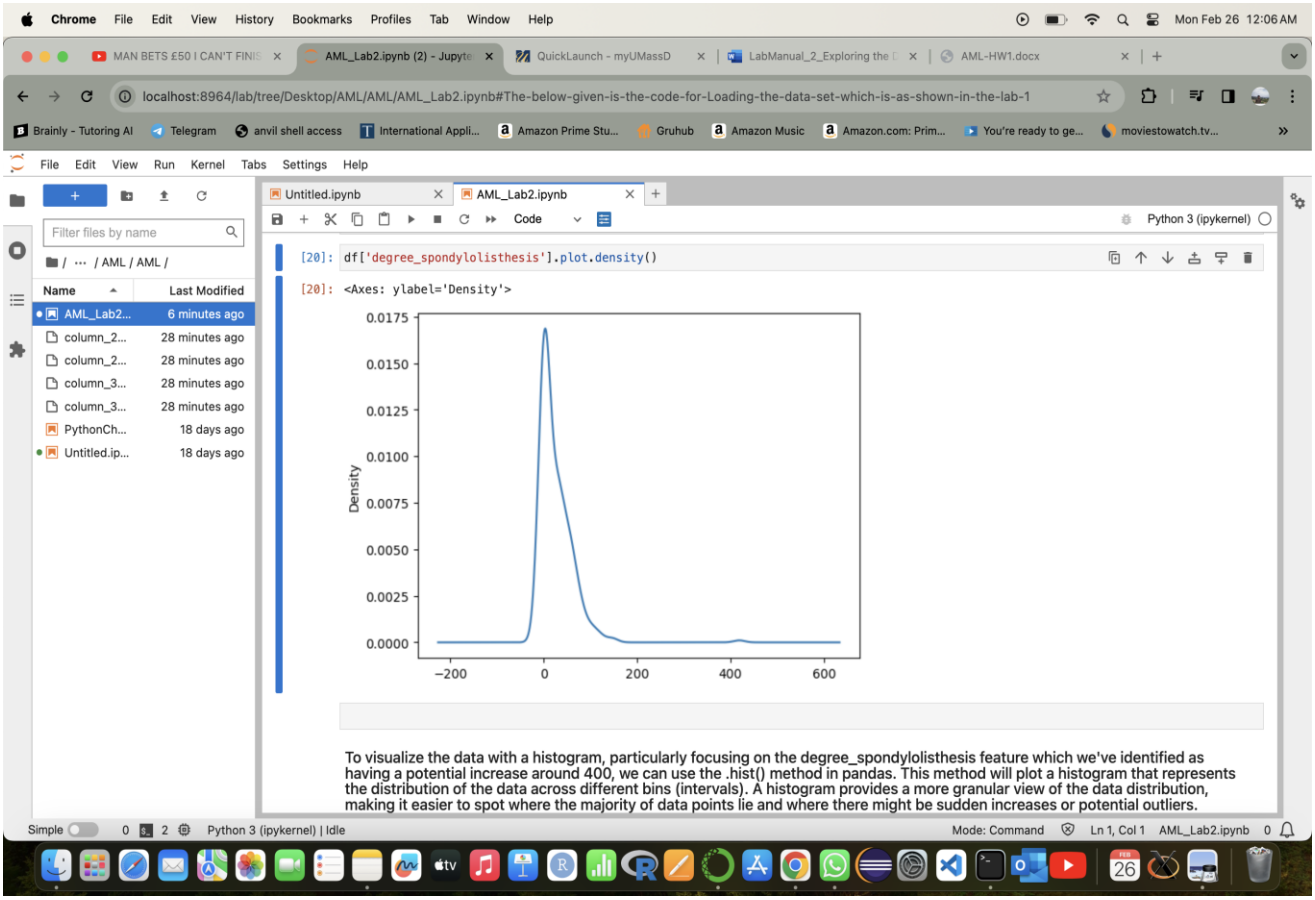
The distribution of pelvic tilt appears to be bimodal, with two peaks at around -10 and 20 degrees. This suggests that there may be two distinct groups of people in the dataset, with one group having a more anteriorly tilted pelvis and the other group having a more posteriorly tilted pelvis.

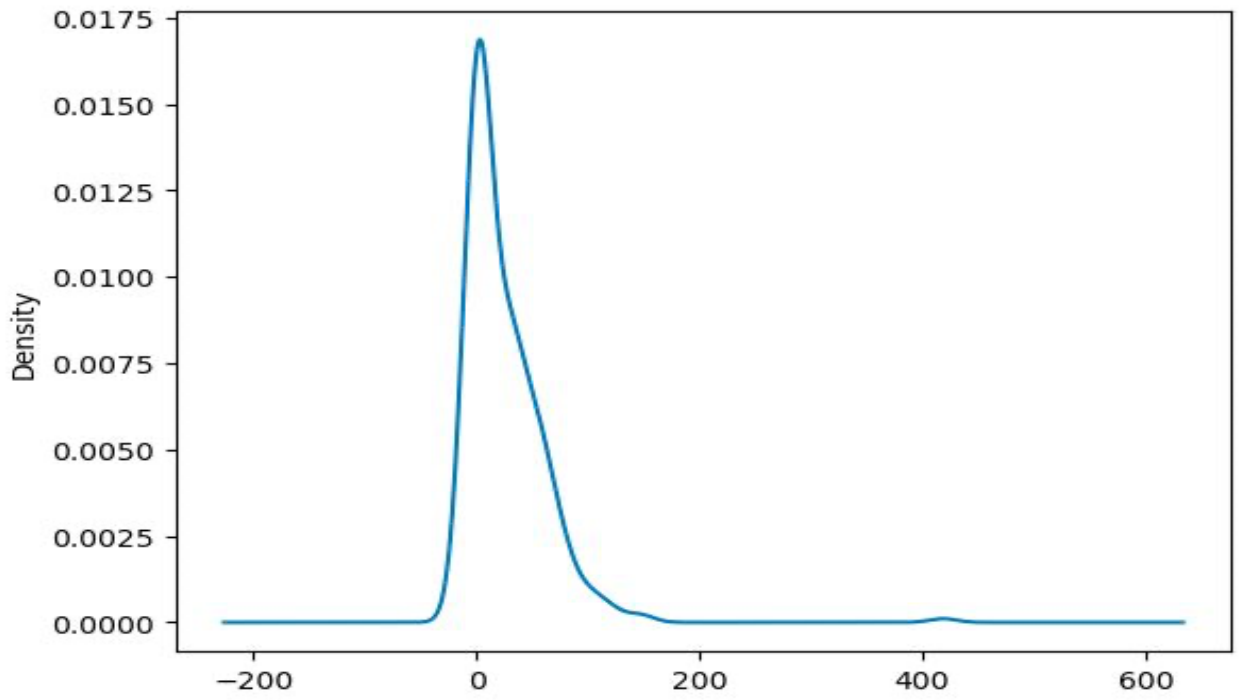
The distribution of lumbar lordosis angle appears to be skewed to the right, with a tail extending towards higher values. This suggests that most people have a moderate amount of lordosis, but some people have a very large amount of lordosis.

The distribution of degree of spondylolisthesis appears to be centered around zero, with a few outliers on either side. This suggests that most people do not have spondylolisthesis, but a small number of people do have it.

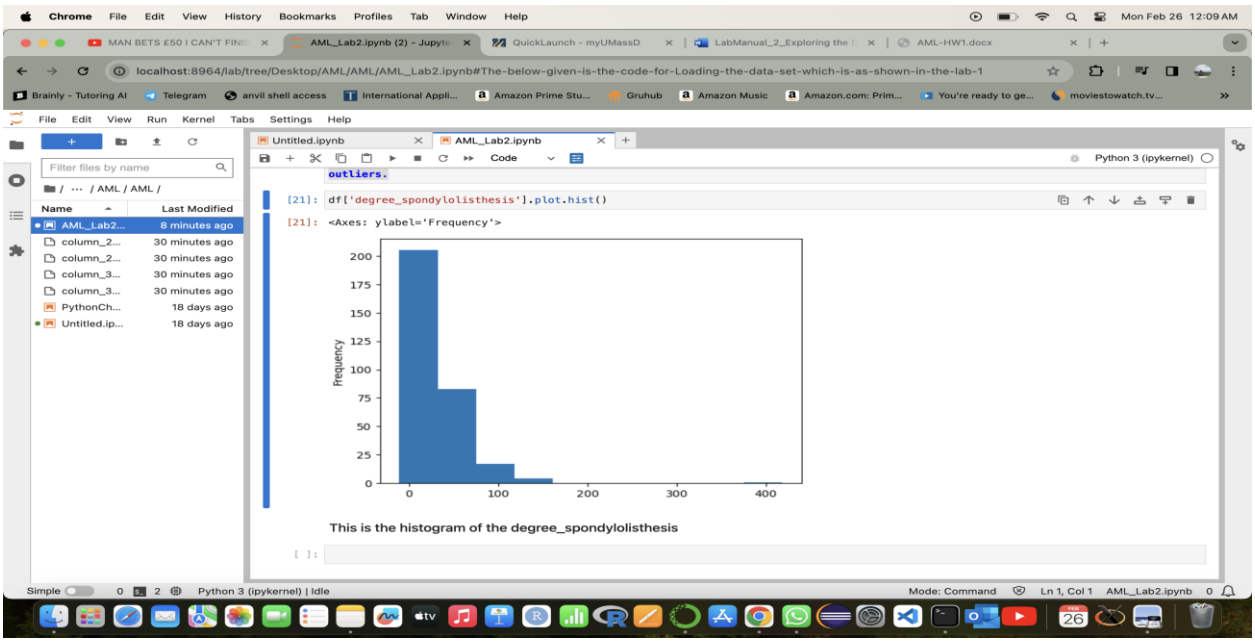
Investigating degree_spondylolisthesis:

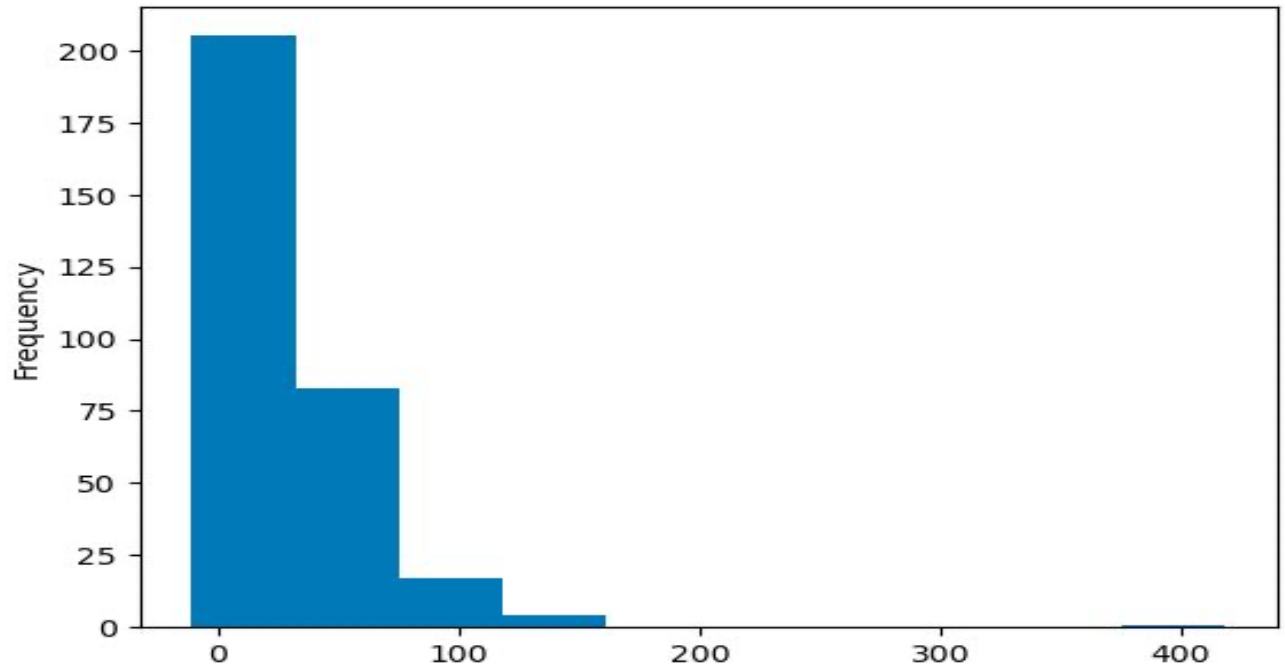
Now we are investigating degree spondylolisthesis, we are starting with density plot which shows the distribution of values





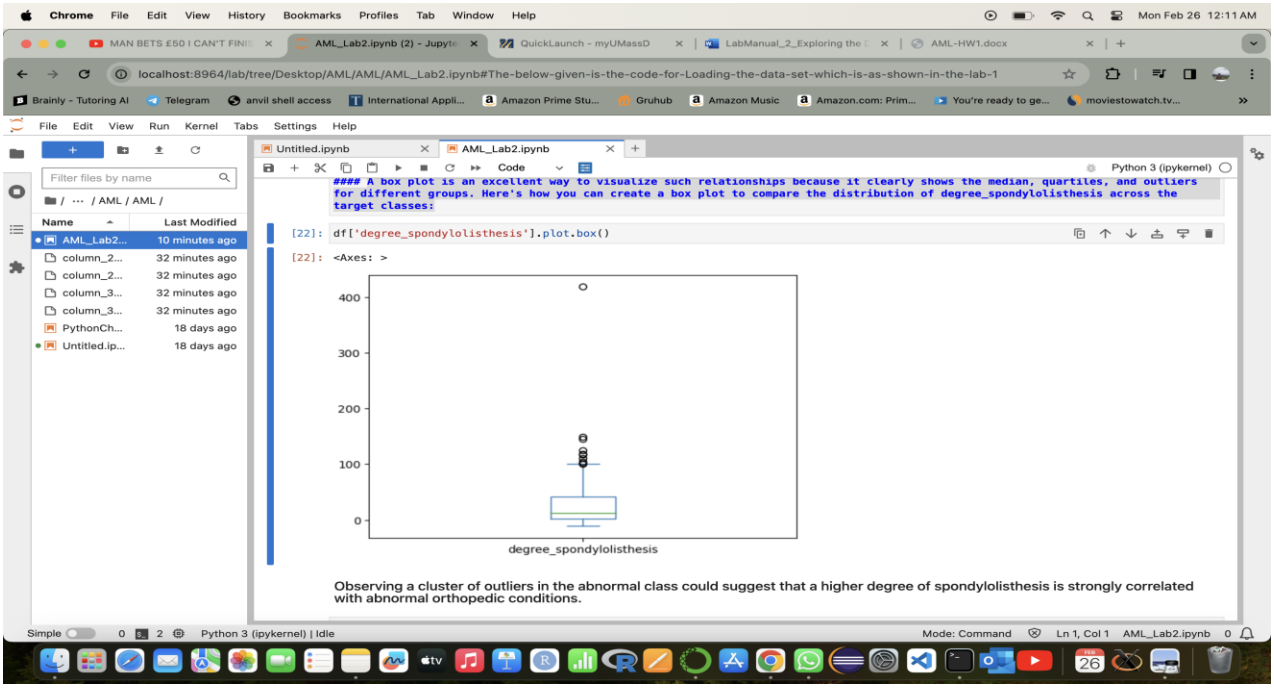
To visualize the data with a histogram, particularly focusing on the degree spondylolisthesis feature which we've identified as having a potential increase around 400, we can use the. hist() method in pandas. This method will plot a histogram that represents the distribution of the data across different bins (intervals). A histogram provides a more granular view of the data distribution, making it easier to spot where most data points lie and where there might be sudden increases or potential outliers.





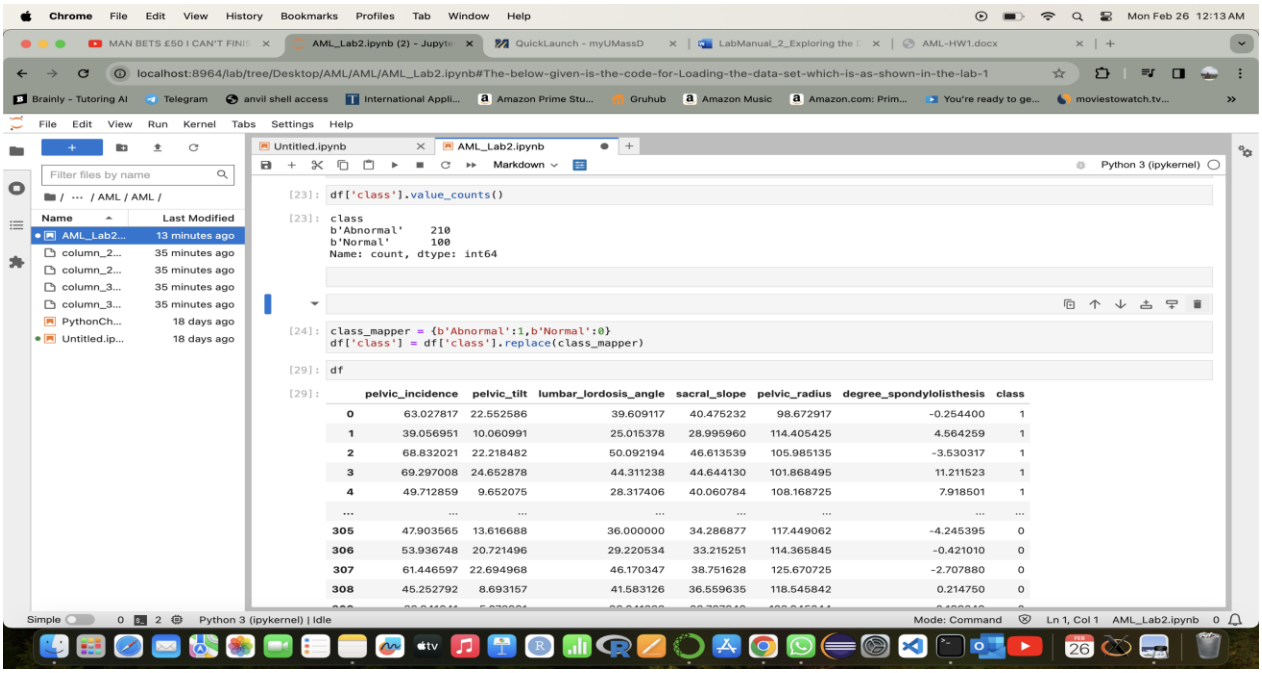
→ This is the histogram of the degree_spondylolisthesis

A box plot is an excellent way to visualize such relationships because it clearly shows the median, quartiles, and outliers for different groups. Here's how you can create a box plot to compare the distribution of degree_spondylolisthesis across the target classes:



Observing a cluster of outliers in the abnormal class could suggest that a higher degree of spondylolisthesis is strongly correlated with abnormal orthopedic conditions.

Analyzing the target:



```
[23]: df['class'].value_counts()

[23]: class
      b'Abnormal'    210
      b'Normal'      100
      Name: count, dtype: int64

[24]: class_mapper = {b'Abnormal':1,b'Normal':0}
      df['class'] = df['class'].replace(class_mapper)

[29]: df
```

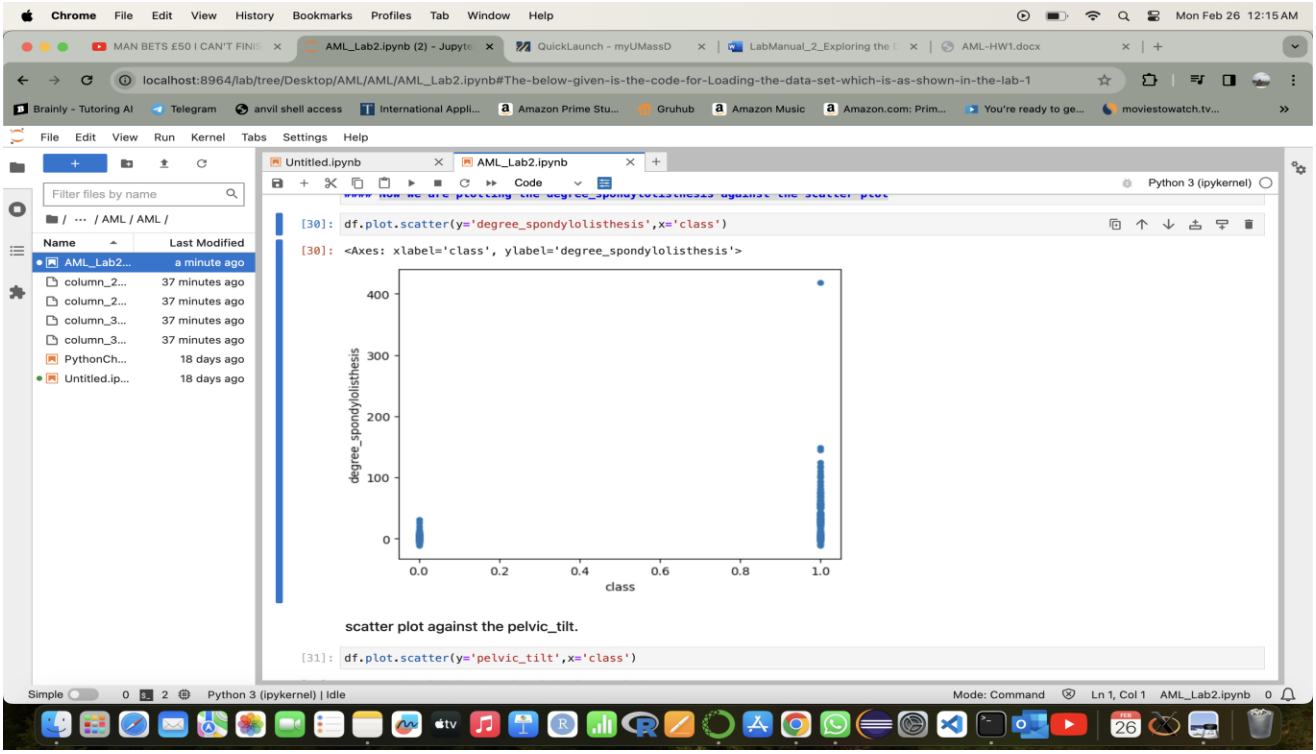
[29]:

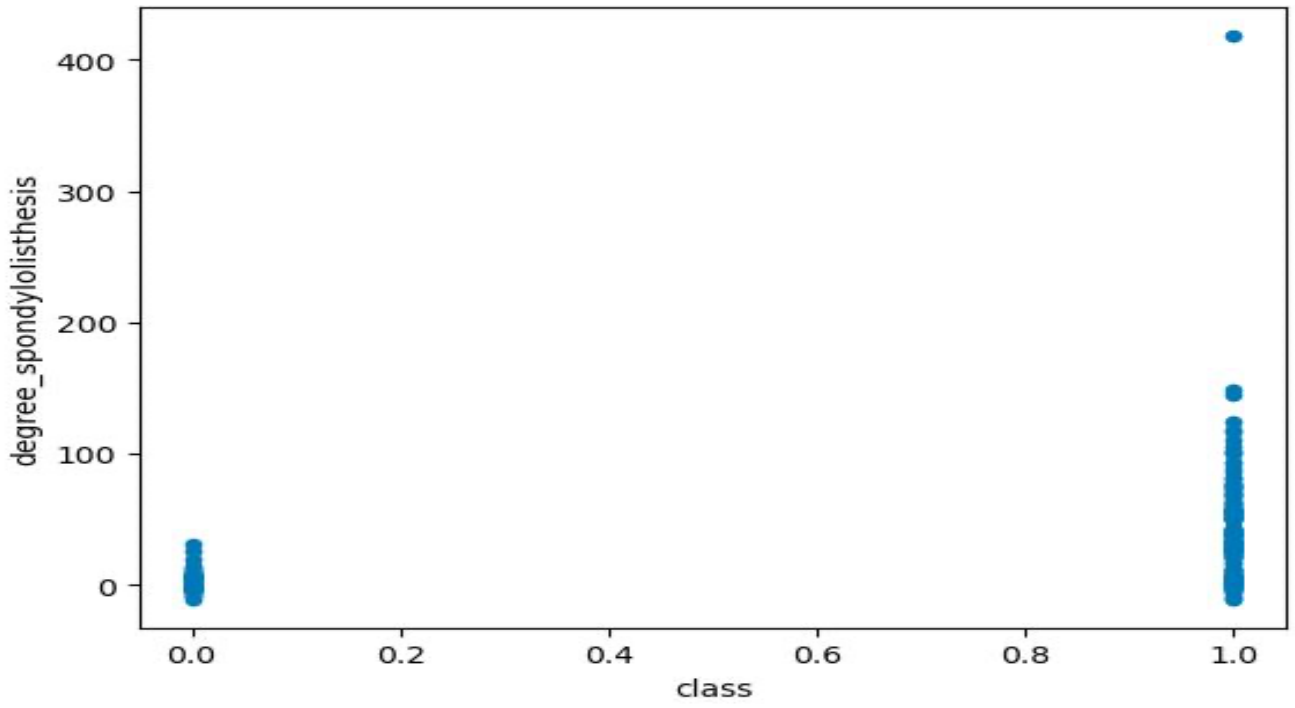
	pelvic_incidence	pelvic_tilt	lumbar_lordosis_angle	sacral_slope	pelvic_radius	degree_spondylolisthesis	class
0	63.027817	22.552586	39.609117	40.475232	98.672917	-0.254400	1
1	39.056951	10.060991	25.015378	28.995960	114.405425	4.564259	1
2	68.832021	22.218482	50.092194	46.613539	105.985135	-3.530317	1
3	69.297008	24.652878	44.311238	44.644130	101.868495	11.211523	1
4	49.712859	9.652075	28.317406	40.060784	108.168725	7.918501	1
...	...	...	...	...	...	...	...
305	47.903565	13.616688	36.000000	34.286877	117.449062	-4.245395	0
306	53.936748	20.721496	29.220534	33.215251	114.365845	-0.421010	0
307	61.446597	22.694968	46.170347	38.751628	125.670725	-2.707880	0
308	45.252792	8.693157	41.583126	36.559635	118.545842	0.214750	0

The output of df['class'].value_counts() shows the distribution of classes in your dataset, indicating there are 210 cases labeled as b'Abnormal' and 100 cases labeled as b'Normal'. This information is useful for understanding the balance of your dataset.

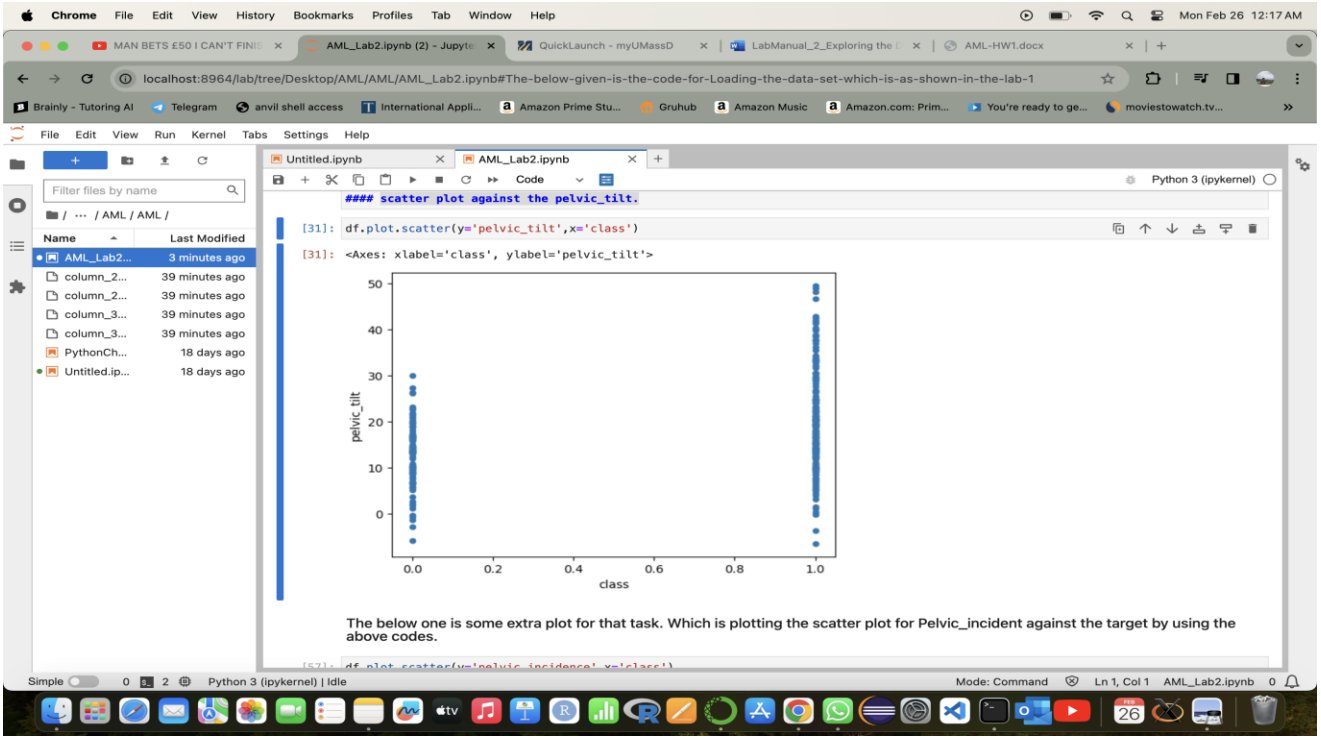
To convert the class column from binary strings to numeric values, we've defined a mapping dictionary named class_mapper where b'Abnormal' is mapped to 1 and b'Normal' is mapped to 0. This conversion is necessary because most machine learning algorithms work with numeric data, and having your target variable in a numeric format will facilitate model training and evaluation.

Now we are plotting the degree_spondylolisthesis against the scatter plot

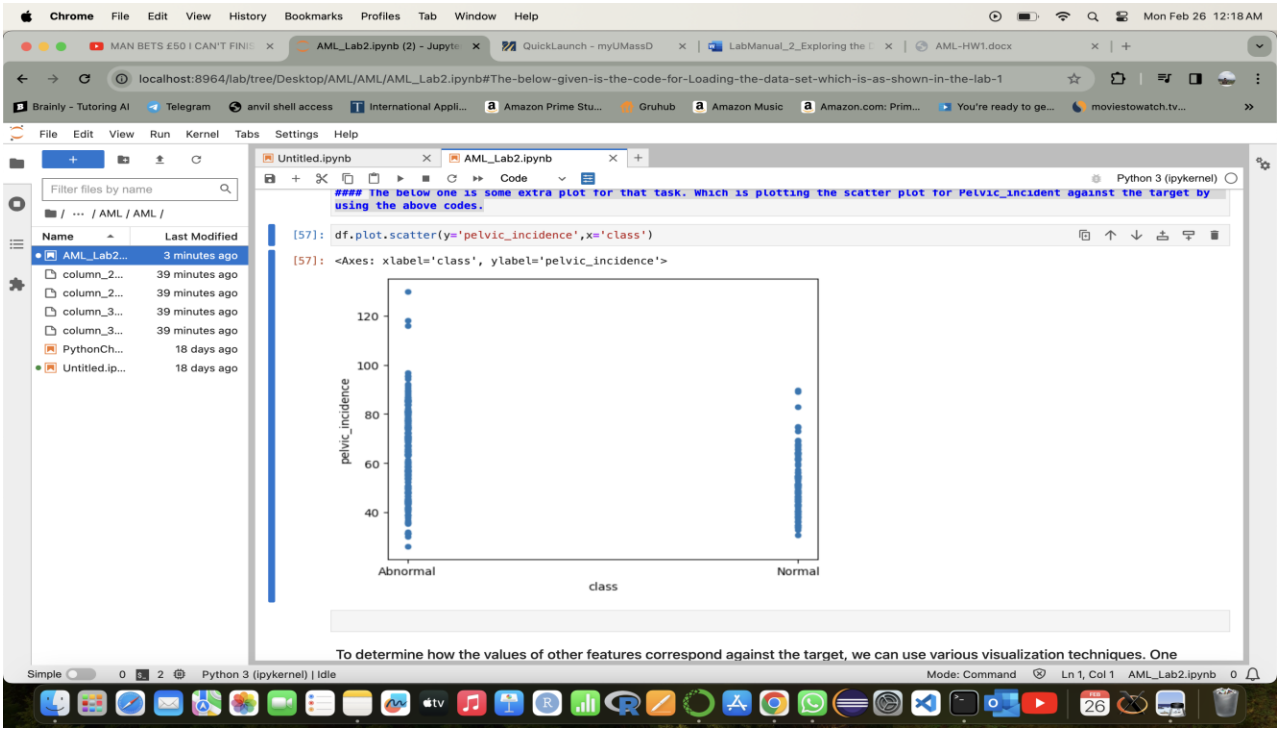




scatter plot against the pelvic_tilt.



The below one is some extra plot for that task. Which is plotting the scatter plot for Pelvic_incident against the target by using the above codes.

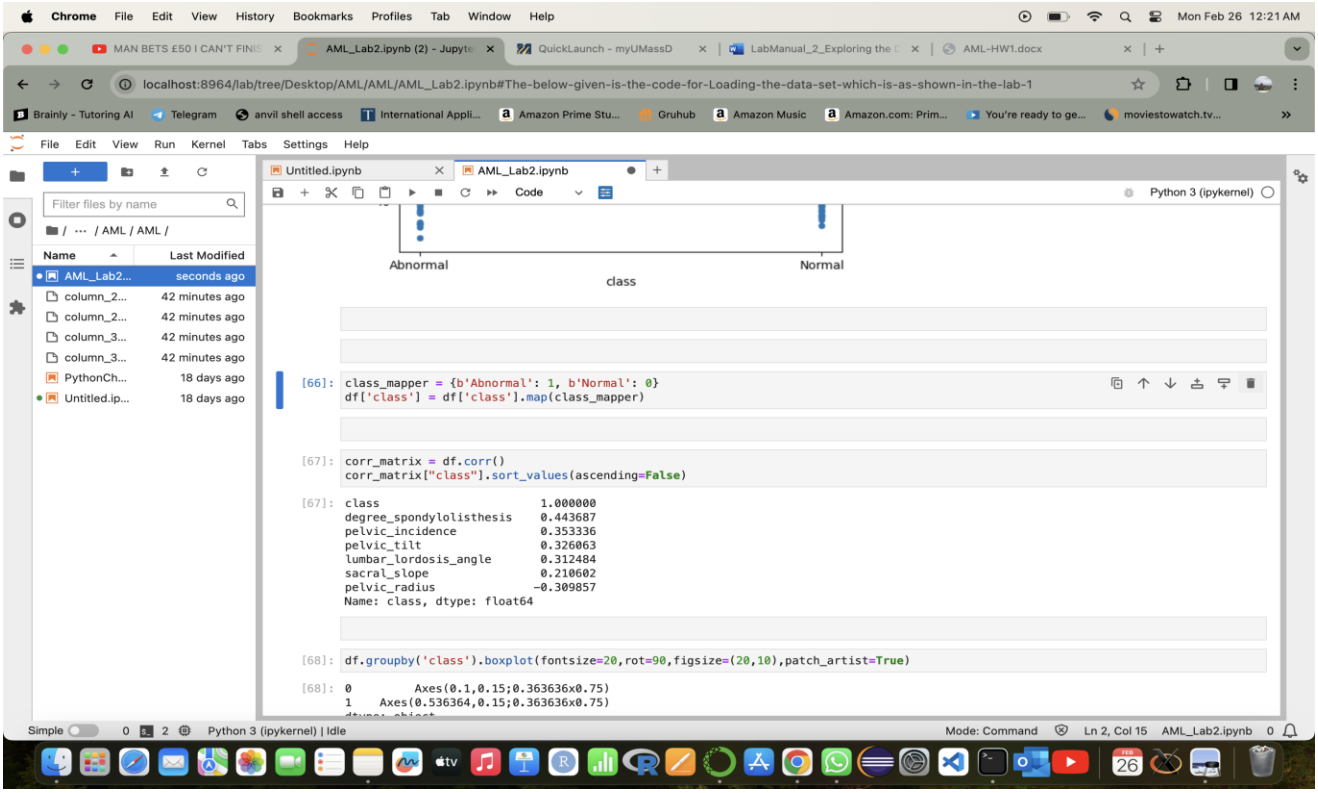


To determine how the values of other features correspond against the target, we can use various visualization techniques. One effective approach is to create scatter plots for each feature against

the target, and use color coding to differentiate between the classes. Additionally, box plots can offer insights into the distribution of feature values across different classes.

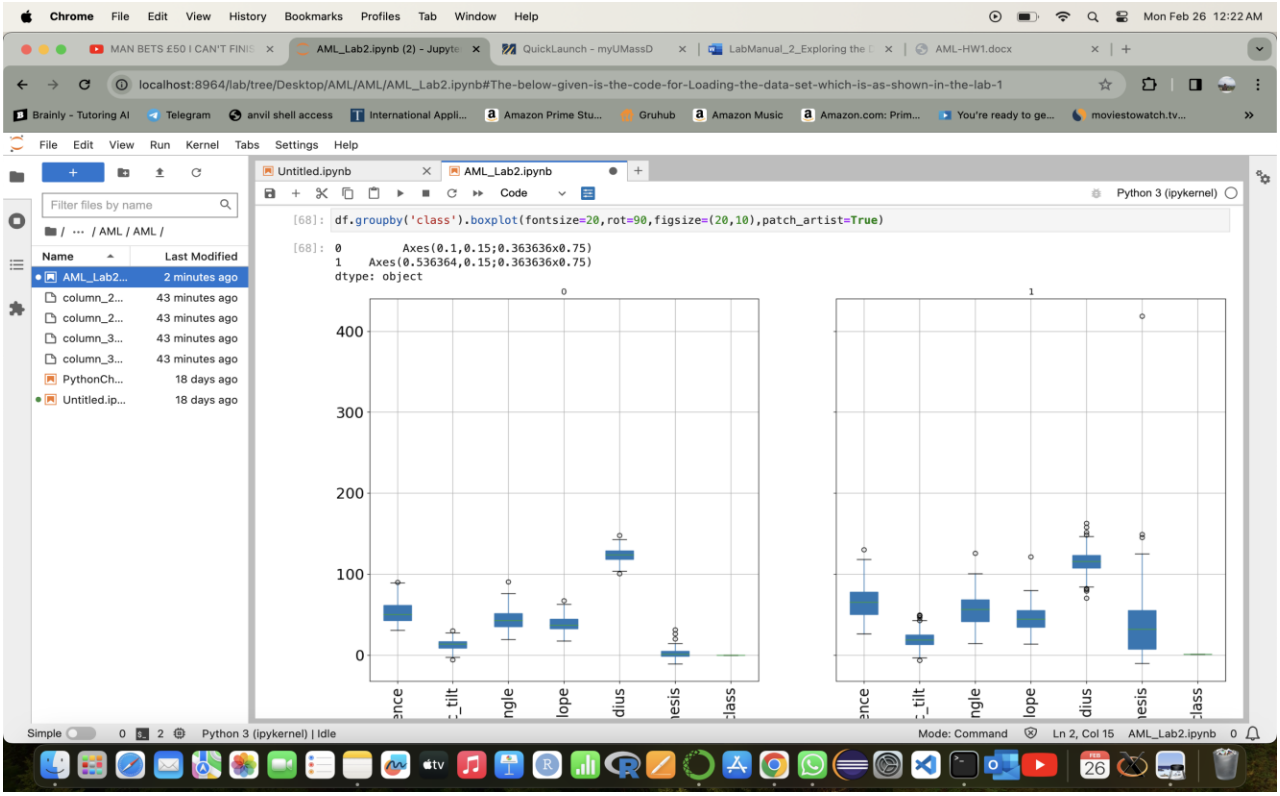
Visualizing multiple variables:

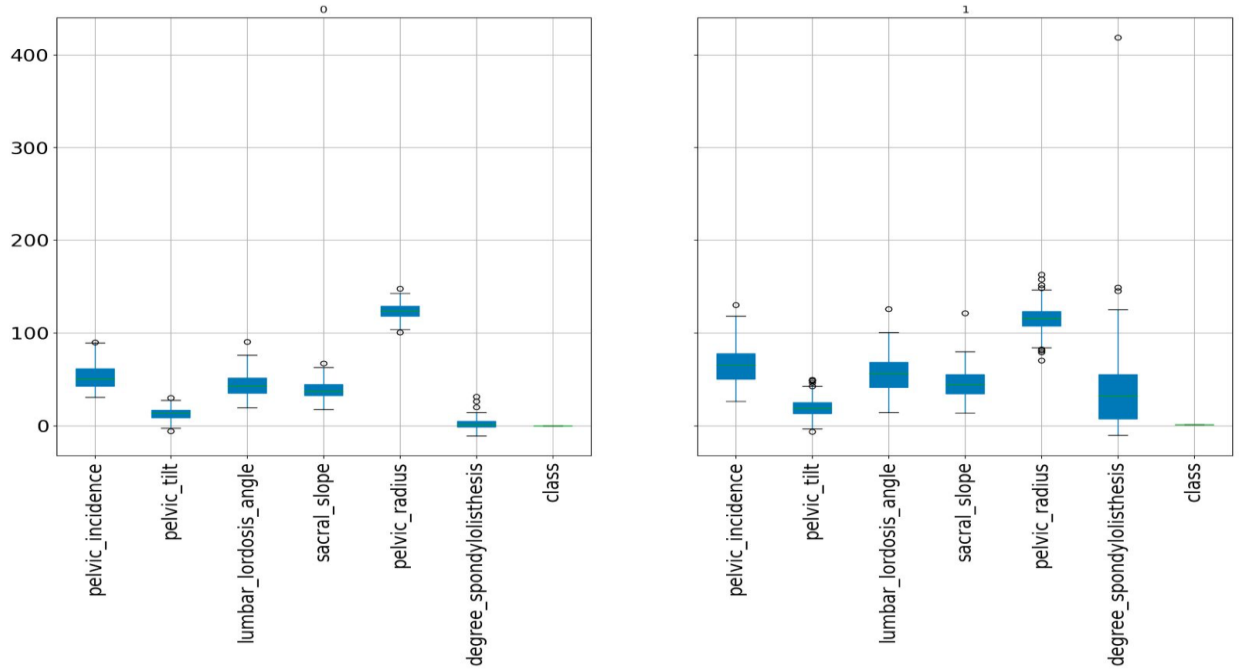
First, let's create a correlation matrix and look at how each feature correlates with the 'class'. This will give us a numerical understanding of relationships.



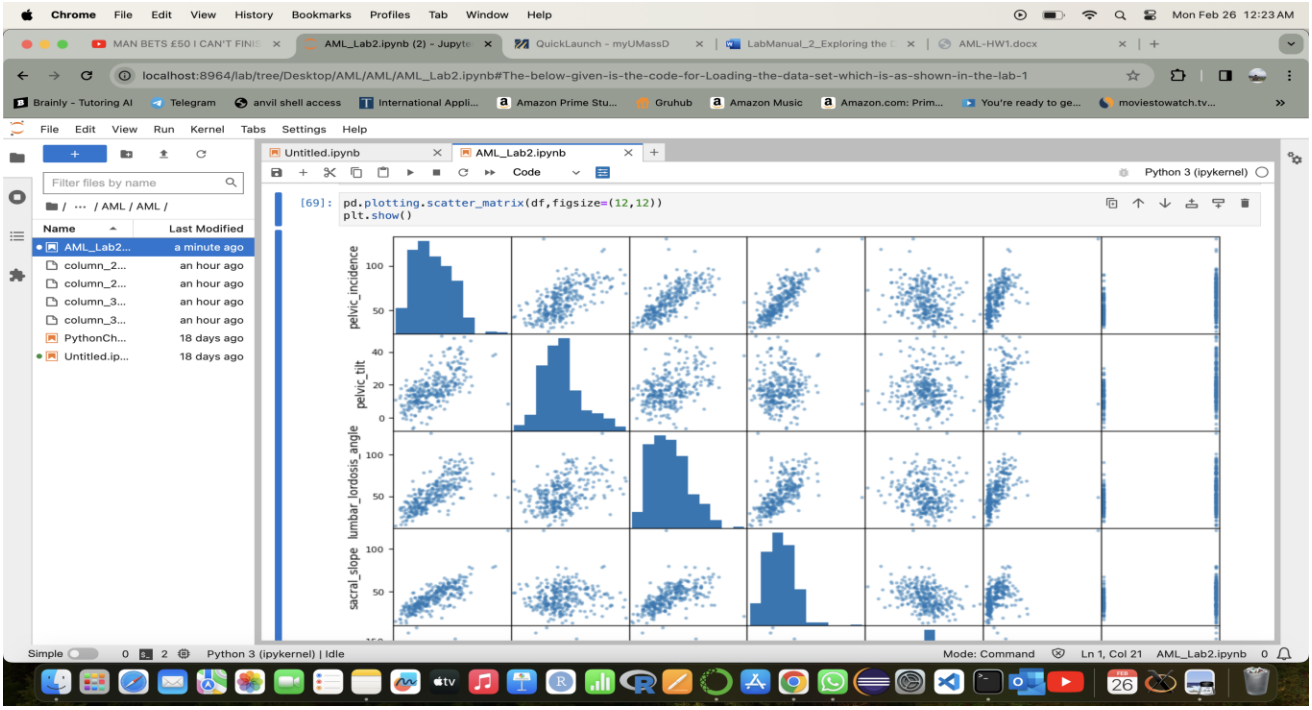
This code snippet prints out the correlation coefficients between each feature and the 'class', sorted in descending order. A positive value indicates a positive correlation, meaning as the feature value increases, the likelihood of the class being 'Abnormal' (1) increases. Conversely, a negative value indicates an inverse relationship.

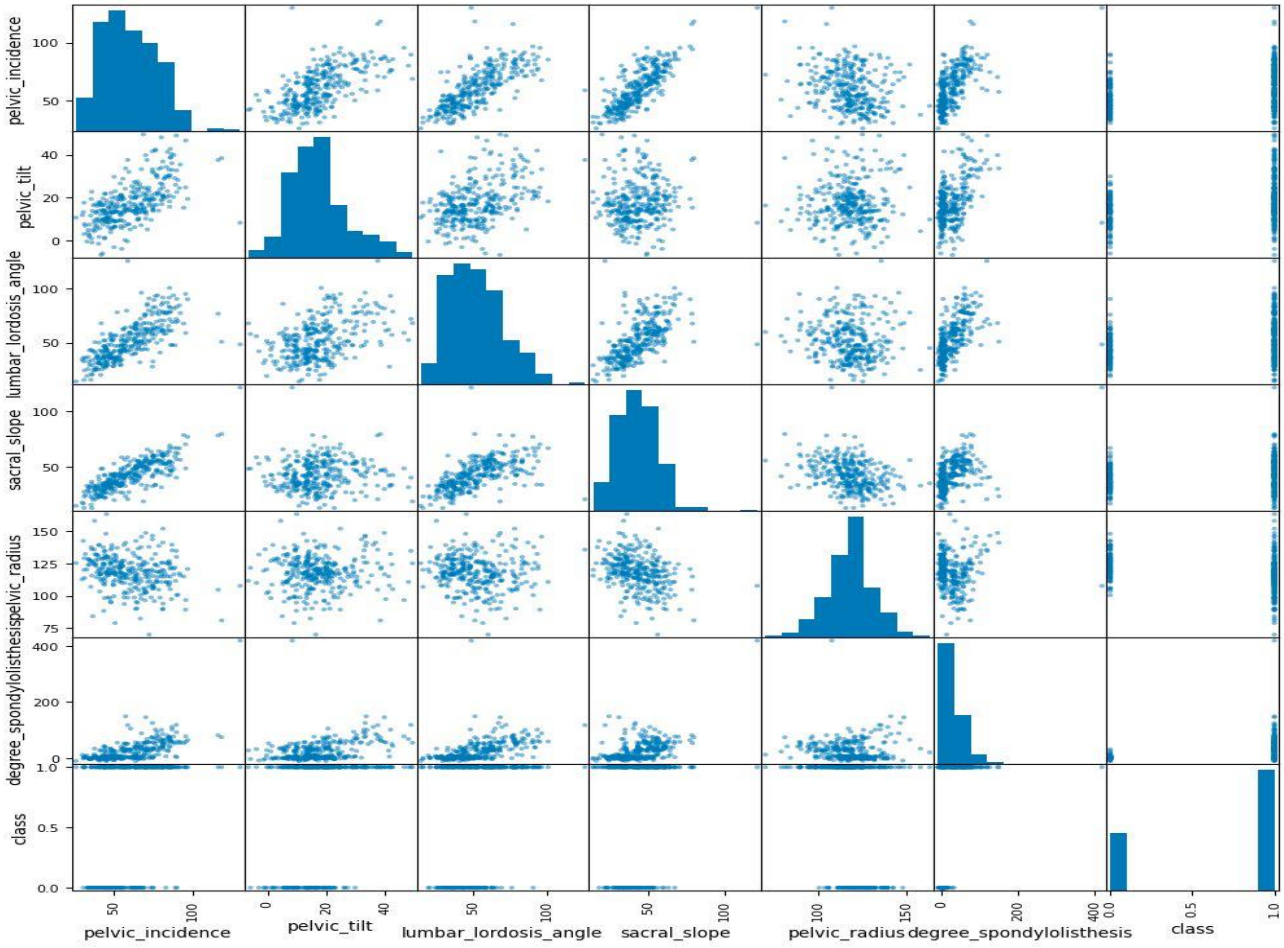
Now we have created a correlation of matrices by using the corr function for the entire dataset.



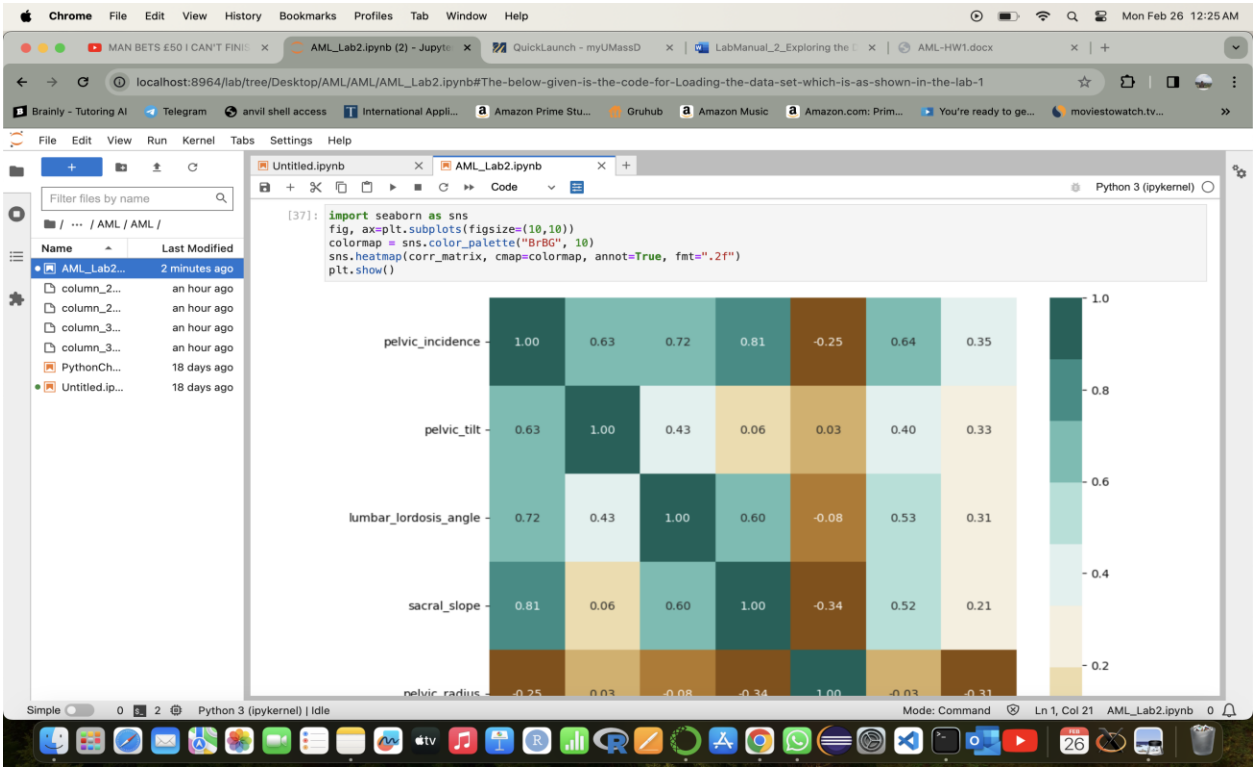


This command will generate box plots for each numeric feature within each class category, allowing you to visually inspect differences in distributions, identify outliers, and get a sense of the spread and central tendency (median) of the data across different classes.


Chrome File Edit View History Bookmarks Profiles Tab Window Help
Mon Feb 26 12:23 AM
AML_Lab2.ipynb (2) - Jupyte
QuickLaunch - myUMassD
LabManual_2_Exploring the D
AML-HW1.docx
localhost:8964/lab/tree/Desktop/AML/AML/AML_Lab2.ipynb#The-below-given-is-the-code-for-Loading-the-data-set-which-is-as-shown-in-the-lab-1
Brainly - Tutoring AI
Telegram
anvil shell access
International Appli...
Amazon Prime Stu...
Gruhub
Amazon Music
Amazon.com: Prim...
You're ready to ge...
moviestowatch.tv...
File Edit View Run Kernel Tabs Settings Help
Untitled.ipynb
AML_Lab2.ipynb
Code
Python 3 (ipykernel)
Filter files by name
/ AML / AML /
Name
Last Modified
AML_Lab2... a minute ago
column_2... an hour ago
column_2... an hour ago
column_3... an hour ago
column_3... an hour ago
PythonCh... 18 days ago
Untitled.ip... 18 days ago
[69]:
pd.plotting.scatter_matrix(df,figsize=(12,12))
plt.show()
pelvic_incidence
pelvic_tilt
lumbar_lordosis_angle
sacral_slope
Simple
Python 3 (ipykernel) | Idle
Mode: Command
Ln 1, Col 21
AML_Lab2.ipynb

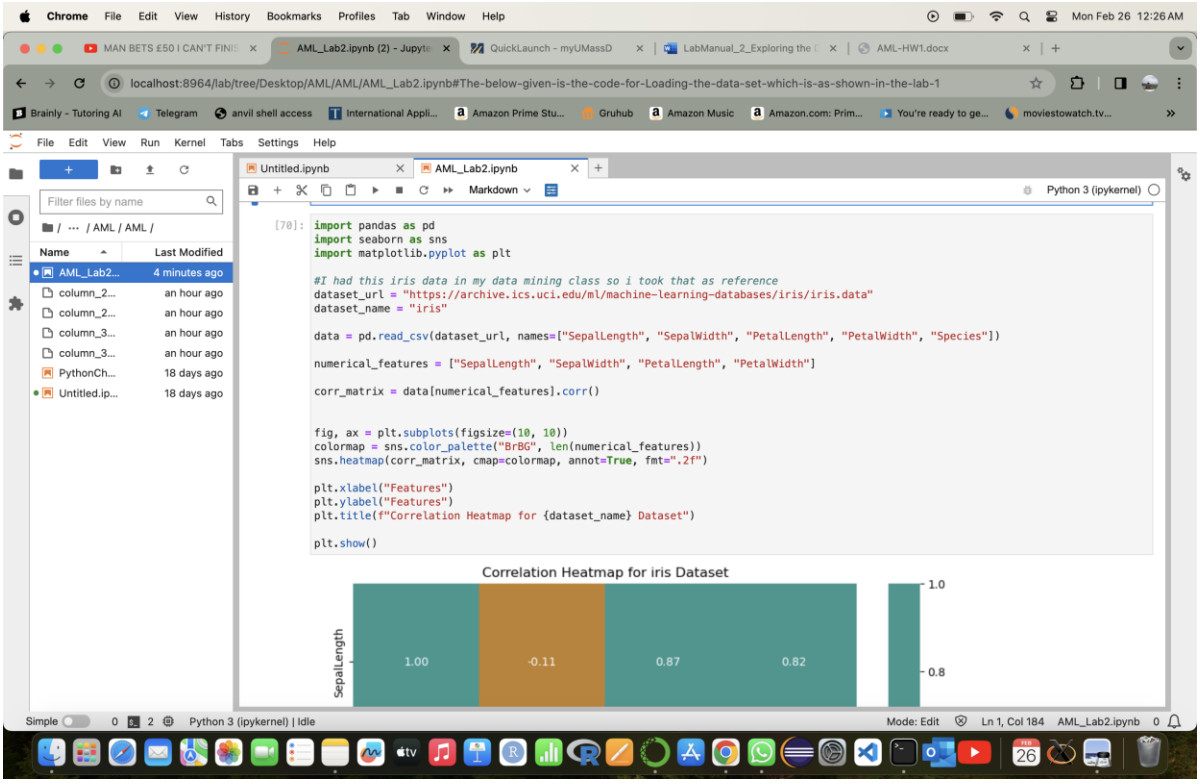

pelvic_incidence
pelvic_tilt
lumbar_lordosis_angle
sacral_slope
pelvic_radius
degree_spondylolisthesis
class

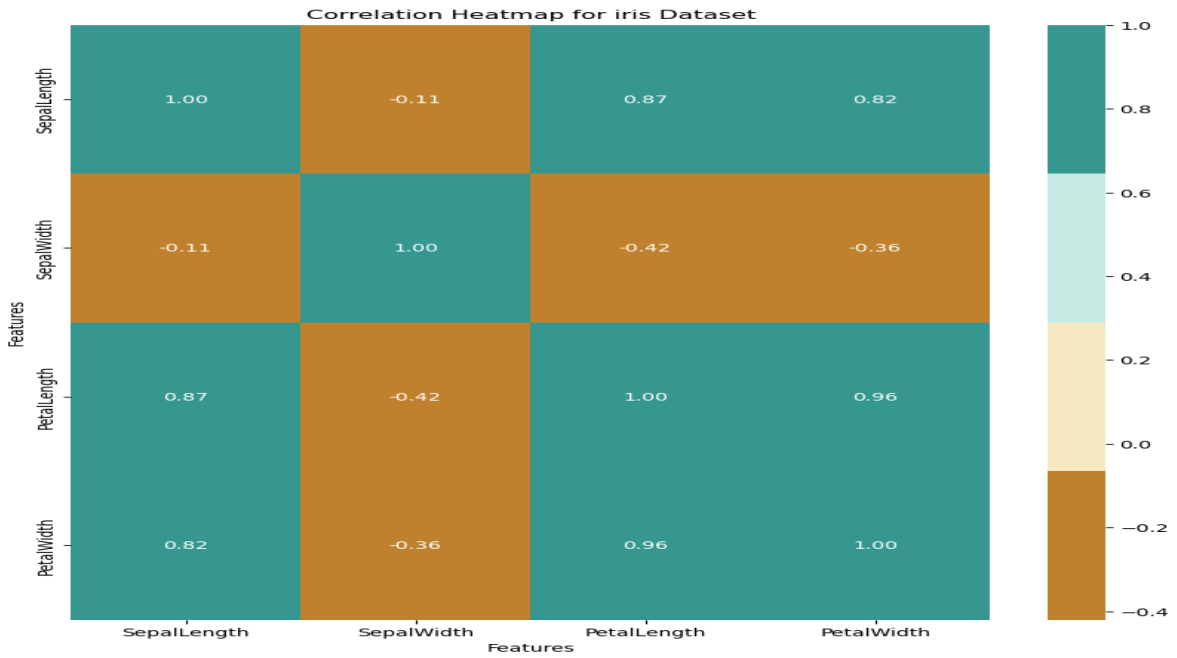
It uses pandas plotting capabilities to create a scatter matrix (also known as a pair plot) for the DataFrame df. This scatter matrix will include all the numerical columns in df.



The Seaborn library visualizes the correlation matrix of a dataset as a heatmap. This visualization is an effective way to quickly understand the relationships between variables in your dataset.

As given in the challenging task to "Find other data from the UCI Machine Learning Repository. Using the previous code for reference", The below is the code for that ML Repository





It calculates the correlation matrix for the numerical features of the dataset and visualizes this matrix as a heatmap using Seaborn. This visualization helps in understanding the relationships between different numerical features of the Iris dataset.

Conclusion:

We have learned more about our dataset after doing this data exploration process:

- → We'll learn about the dataset's size, structure, and features count
- → Target variable analysis aids in understanding its distribution
- → Class names may need numerical conversion for modeling.
- → Data visualization aids in pattern comprehension.
- → Scatter plots and heatmaps reveal numerical and categorical relationships.
- → Outlier exploration helps assess their impact on analysis.
- → Density plots and outlier detection unveil data patterns and possible anomalies.
- → Data distribution insights come from numerical features' statistics using df.describe().
- → Knowing the data types of columns guides our preprocessing steps.

By thoroughly investigating the data, we can make informed choices regarding data preprocessing, selecting the most pertinent characteristics, and developing modeling techniques. This comprehensive approach helps in creating more successful predictive models and extracting significant insights from the data.