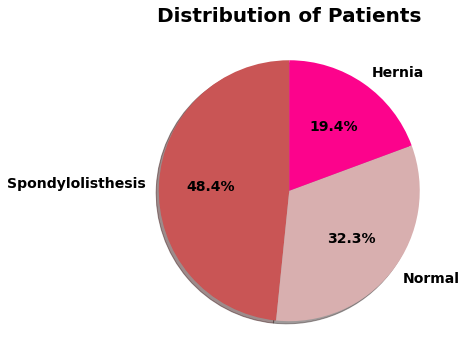
IME672 - Course Project

Group 3 | Members**:** Akanksh Mandoliya (190078), Anhad Jagdish (190133), Ayush Singh (190217), Ayushi Agarwal (190219), Devansh Kumar Sahu (190271)

**Problem Description-**

The problem consists of classifying patients belonging to one out of three categories: Normal (100 patients), Disk Hernia (60 patients) or Spondylolisthesis (150 patients) with six given attributes.

**Data Understanding**

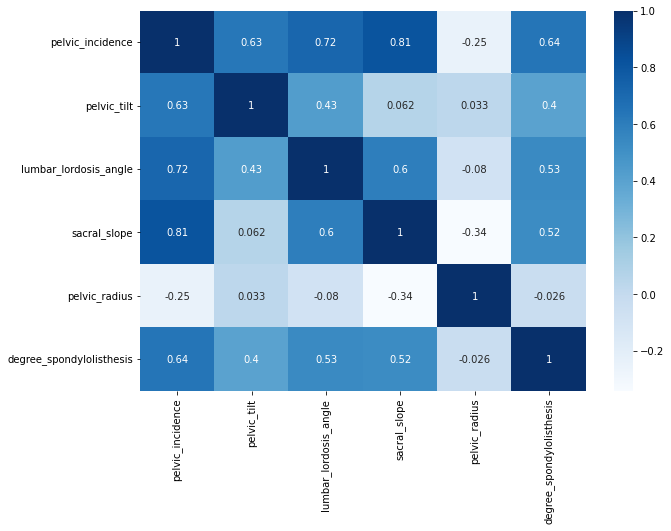
The dataset given to us is about the patients with six biomechanical features about the spinal alignment that were classified under three categories that are “Normal”, “Hernia”, and “Spondylolisthesis”. There were a total of 310 data points among which 150 were classified as “Spondylolisthesis”, 60 as “Hernia”, and 100 as “Normal”. The below pie chart shows the distribution of the same.

Now the 6 attributes that are given are as follows-

| Attribute | Type | No of outliers |
| --- | --- | --- |
| Pelvic\_incidence | Numeric(Float) | 3 |
| Pelvic\_tilt | Numeric(Float) | 13 |
| Lumbar\_lordosis\_angle | Numeric(Float) | 1 |
| Sacral\_slope | Numeric(Float) | 1 |
| Pelvic\_radius | Numeric(Float) | 11 |
| Degree\_spondylolisthesis | Numeric(Float) | 10 |

**Data Pre-processing**

The first thing we looked for in our data was null values. But we were surprised to find that there were no null values in our data set. Then we looked for some inconsistent values as if any other data type is there instead of float values as all our 6 attributes would take float values but could not find any. So our data was already clean thus we did not need to perform any data cleaning technique.

**Correlation**

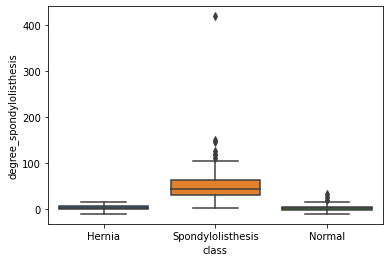
In order to find the correlation between the attributes, we plotted a correlation heat map among the six attributes. From the heat map, it was found that two attributes “pelvic\_incidence” and “sacral\_slope” are the most correlated attributes among all of them, and “pelvic\_radius” was the least correlated to any other 5 attributes.

**PP Score**

Since all our six attributes are float values and our class is an object, we plotted a PP score heat map to find out which of the 6 attributes has the highest predictive power to predict the class.



It turns out that the attribute “degree\_spodylolisthesis” has the highest PP score to predict the class of the patient. So we plotted a box plot between “degree\_spodylolisthesis” and class labels, as shown below.

After studying that box plot we observed that there is one data point that lies far away from the whole data set and hence is a noisy data point. We had to remove this one noisy data point to clean our data.

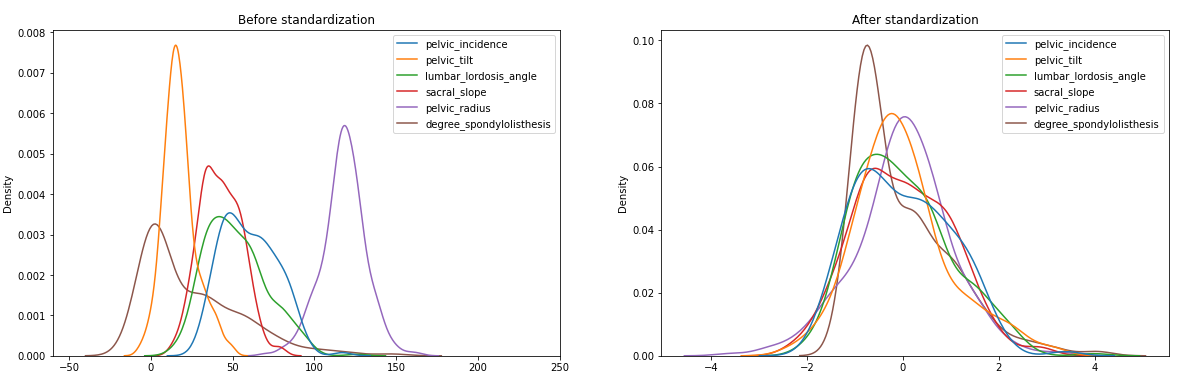
Now in order to draw some insights about our data to successfully build our model we plotted box plots, scatterplots, and bar graphs for all the six attributes which are as follows-

|  |  |
| --- | --- |
|  | |

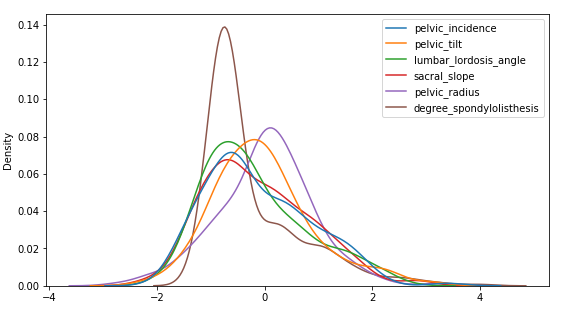
By looking at these plots, we found out that the average value of “degree\_spondylolisthesis” is distinctly very high for the spondylolisthesis class as compared to the other two classes. Also, the average value of “pelvic\_tilt” for the normal class is quite lower than the abnormal classes. Along with these, “sacral\_slope” is relatively low in the case of Hernia and the mean values for “pelvic\_radius” for all the three classes are almost similar.

**Normalizing and Balancing classes**

As we see in the data that the values of all the different classes have different ranges, we normalized the data using z-score normalization. This brought all the values in the range between -3 to 3. The following kernel density estimation plot shows the same:-



Also in our dataset, we see that the number of rows for spondylolisthesis patients is higher than that of patients having hernia (149 vs 60). So we used SMOTE to generate synthetic data points so that our classes are balanced. After resampling the data, we again generated kernel density estimation plot to check whether our new data is equivalent to the previous one:



As we see our estimates are not changed, we can proceed with training and testing the models.

**Model Building and Evaluation:**

**Models:** We used 6 models for classification of our data, Decision Tree Classifier, K-Nearest Neighbors, Support Vector Machine, Naive Bayes, Random Forest and Neural Network.

**- Decision Tree Classifier:** Here, we used the decision tree as a predictive model to map features to conclusions about the target value. In our case since we had only 6 attributes, our tree was quite small. There was overfitting initially as for training data, our model accuracy was 100%, so we did post pruning by using the cost complexity pruning method. It reduced the training data accuracy a bit but increased our test data accuracy.

Train accuracy: 99.03846% | Test accuracy: 80.64516%

**- Random Forest:** Here, we used Random forests as an ensemble learning method for classify by constructing a multitude of decision trees at training time and outputted the class that is the mode of the classes as the classification result. We also used GridSearchCV to do an exhaustive search over specified parameter values for an estimator, which was used in our model. There was no overfitting though the training data accuracy was 100%.

Train accuracy: 100.0% | Test accuracy: 81.72043%

**- K Nearest Neighbors:** Here in the k-Nearest Neighbors algorithm, a non-parametric method is used for classification. A sample is classified by a majority vote of its neighbors, with the sample being assigned to the class most common among its k nearest neighbors. In this also, we used GridSearchCV to do an exhaustive search over specified parameter values for an estimator, which was used in our model.

Train accuracy: 96.47436% | Test accuracy: 72.04301%

**- Support Vector Machine:** In SVM, we give a set of training samples to the model, each marked as belonging to one or the other of two categories, then the SVM training algorithm builds a model that assigns new test samples to one category or the other.

Train accuracy: 91.02564% | Test accuracy: 79.56989%

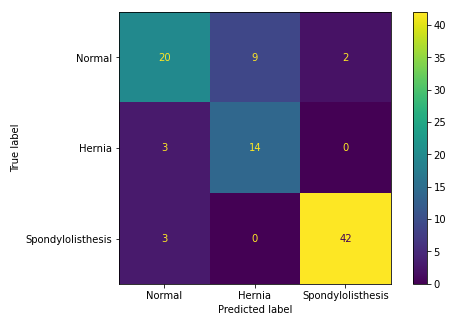
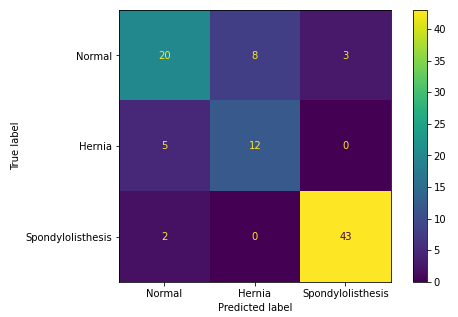
**Naive Bayes:** In Naive Bayes classifiers, we used a family of simple probabilistic classifiers based on applying Bayes' theorem with strong (naive) independence assumptions between the features.

Train accuracy: 82.69231% | Test accuracy: 79.56989%

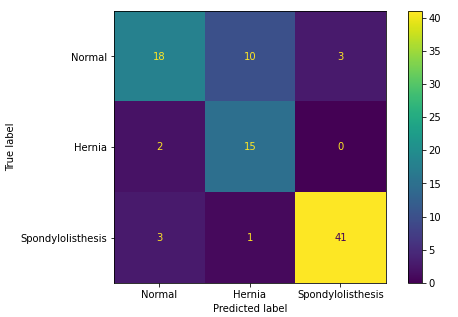
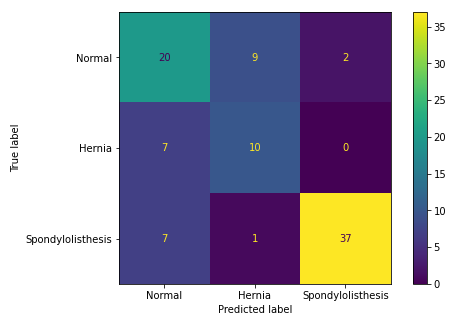
**Neural Networks:** In Neural Networks, we have complex and more powerful algorithms than standard machine learning, it belongs to deep learning models. We used Keras (Keras is a high level API for tensorflow) to build a neural network.

Train accuracy: 87.50% | Test accuracy: 84.95%

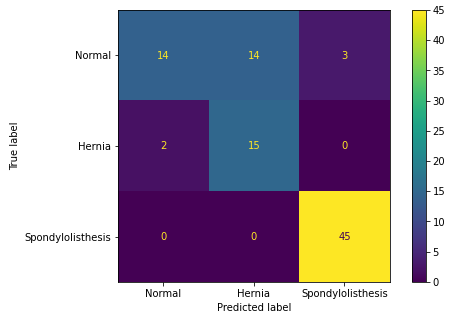
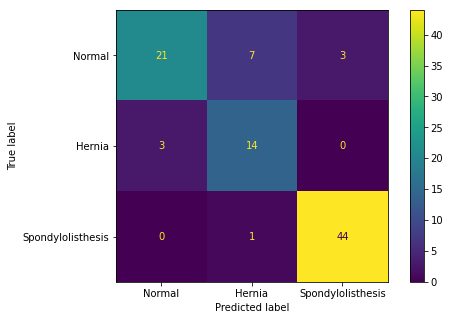
**Results and Interpretations:**

****

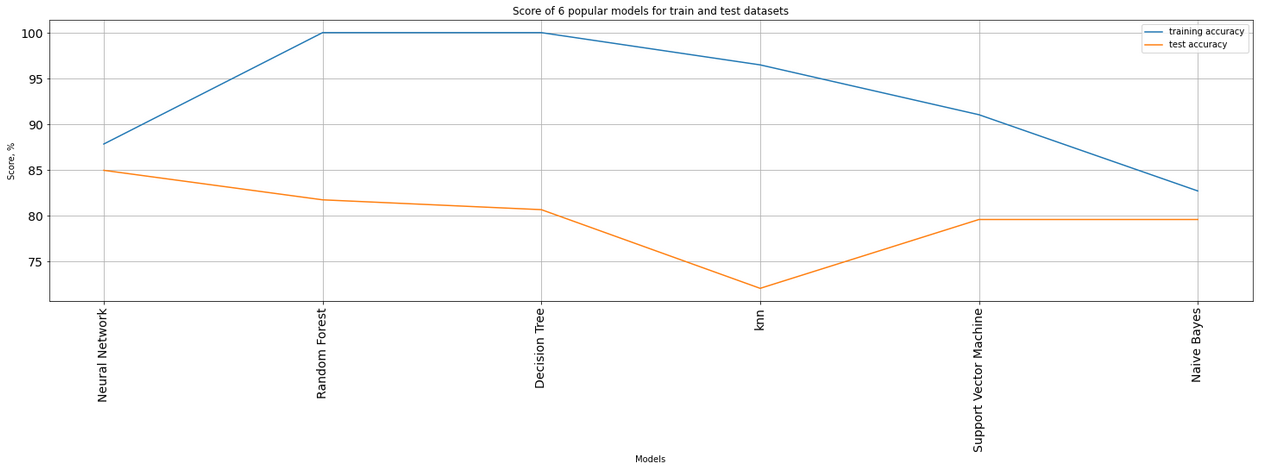
Decision Trees Random Forest



KNN SVM

Naive Bayes Neural Network



Model accuracy comparison

|  | **ROC Curve for Normal vs Abnormal Classification**  The ROC curve shows the trade-off between sensitivity (or TPR) and specificity (1 – FPR). Classifiers that give curves closer to the top-left corner indicate a better performance.  Among all algorithms, Neural Network performed the best giving an accuracy of 84.5% on the test set.  Overall we observe that models are facing issues while predicting data from Hernia class as many are getting classified as Normal. |
| --- | --- |