**A Thesis/Project/Dissertation Report**

on

**Patient Los- A Data Science Web Application**

***Submitted in partial fulfillment of the***
***requirement for the award of the degree of***

Bachelor of Technology in Computer Science and Engineering


**Under The Supervision of**

**Name of Supervisor : Dr Prashant Johri**

**Professor, Galgotias University**

Submitted By

**Sanskar Shukla - 18SCSE1070008**

**Fahad Ahmed Siddiqui - 18SCSE1130008**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**GALGOTIAS UNIVERSITY, GREATER NOIDA**

**INDIA**

**DECEMBER, 2021**

**SCHOOL OF COMPUTING SCIENCE AND ENGINEERING**
**GALGOTIAS UNIVERSITY, GREATER NOIDA**

## CANDIDATE'S DECLARATION

We hereby certify that the work which is being presented in the thesis/project/dissertation, entitled **"DETERMINING PATIENTS' LENGTH OF STAY USING PREDICTIVE ANALYSIS"** in partial fulfillment of the requirements for the award of the Bachelor of Technology in Computer Science and Engineering submitted in the School of Computing Science and Engineering of Galgotias University, Greater Noida, is an original work carried out during the period of September, 2021 to December, 2021, under the supervision of Dr Prashant Johri, Professor, Department of Computer Science and Engineering, of School of Computing Science and Engineering , Galgotias University, Greater Noida.

The matter presented in the thesis/project/dissertation has not been submitted by me/us for the award of any other degree of this or any other places.

Sanskar Shukla, 18SCSE1070008
Fahad Ahmed Siddiqui, 18SCSE1130008

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.

Dr Prashant Johri
Professor, SCSE, GU

## CERTIFICATE

The Final Thesis/Project/ Dissertation Viva-Voce examination of Sanskar Shukla:18SCSE1070008; Fahad Ahmed Siddiqui: 18SCSE1130008, has been held on _____ and his/her work is recommended for the award of Bachelor of Technology in Computer Science and Engineering:

**Signature of Examiner(s)** **Signature of Supervisor(s)**

**Signature of Project Coordinator** **Signature of Dean**

Date: December, 2021

Place: Greater Noida

## Acknowledgement

## Abstract

Data Science is potentially one of the most noteworthy advancements of the 21st century. It is a data-based or evidence-based analysis performed for future predictions. At its core, it uses scientific methods like statistics, machine learning and deep learning algorithms to analyse data which are then exhibited through interactive ways like plots and charts. Data Science has various applications in significant fields like agriculture, healthcare, education, etc. We also aim to improve a crucial sector in the nation's growth through data science, healthcare. As we live in a diverse population, it leads to various challenges to the healthcare system. As we live in a diverse population, it leads to various challenges to the healthcare system. According to the WHO, a patient's Length of Stay (LoS) in hospitals is an important performance measurement and monitoring indicator. Through a data-driven approach, we can predict the LOS at admission time and can keep our resources in check.

The arrangement of healthcare administrations is one of the biggest and most complex businesses around the world. As one of the fundamental necessities to support life, it faces the results of expanding requests in the midst of restricted monetary assets and seeking social requirements. Giving the proper clinical consideration includes dynamic as far as arranging and the executives of healthcare assets. Hospital Length of Stay (LoS) is viewed as a dependable and legitimate intermediary for estimating the utilization of emergency clinic assets. Although, normal LoS can be deceiving since the basic dissemination isn't symmetric. Thus models dependent on the normal LoS can not portray the hidden circulation of patients. Thus, it is apt to say that LoS is a valid proxy for estimating the hospital's resource consumption. Through this, we tend to mitigate some problems in hospitals. This in turn can save capital for hospitals and even the lives of patients.

We used PyCharm IDE as PyCharm is a devoted Python Integrated Development Environment (IDE) giving a wide scope of fundamental apparatuses for Python designers, firmly incorporated to establish an advantageous climate for useful Python, web, and information science improvement. Along with Scikit-Learn which is the most valuable and powerful library for AI in Python. TensorFlow is a start to finish open source stage for AI. It has a far reaching, adaptable biological system of devices, libraries and local area assets that allows specialists to push the

cutting edge in ML and designers effectively fabricate and convey ML fueled applications. Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

We tend to make a web application that is just an extension of what we work on, i.e., we want the user to have a good experience while finding the solution. We want data visualization, accuracies of various models, and also a model which par exceeds it. Thus, we want the user to not feel this as a scientific and theoretical solution but also enhance the user experience.

At last we were able to achieve our target. A web application built using Streamlit which does a good job of enhancing the user experience and also helps us to bind all the components of our project together into a single web application.

## Contents

## List of Figures

**Acronyms**

| | |
|---|---|
| ML | Machine Learning |
| SVM | Support Vector Machine |
| SVC | Bachelor of Computer Applications |
| NB | Naive Bayes |
| SGD | Stochastic Gradient Descent |
| DL | Deep Learning |
| KNN | K Nearest Neighbour |

# CHAPTER 1: INTRODUCTION

## INTRODUCTION

The expense of medical care is expanding, this normal wonder in industrialized nations could be clarified by the development of clinical information and biomedical innovation. Our contextual investigation depends on an informational index that began from the University Polyclinic Campus Bio-Medico and it is shown that standard models are not sufficient to portray the ow patients. To beat such restrictions we propose speculation that depends on a couple of boundaries and supplies a phenomenological understanding of information. Besides, to bear the cost of a functional rule for arranging and the board of medical clinic beds we propose a technique to assess the gauging capacity. Exactly we revamp a few estimates utilized in monetary danger by the executives like the Value at Risk and Expected Shortfall to show the integrity of the model. In the following segment, a short portrayal of Campus Bio-Medico clinic and some connected insights are summed up. The procedure area follows with a portrayal of stage type appropriations and the depiction of our speculation. The paper then, at that point, reports the strategies used to the information and the principal results. At last, some broad ends are drawn in regards to the reasonable ramifications of these techniques.

## FORMULATION OF PROBLEM

- As we mentioned we want to improve the healthcare system by analyzing the various models.
- We want to follow the very basic ML pipeline to do so.
- We will first explore the dataset which is the form of a csv file format.
- We will clean the dataset so that we can use it to train the model.
- Now we will engage in feature engineering which is followed by evaluating data.
- Now the visualization part kicks in where we make beautiful plots using suitable libraries.
- We will now train various models and try to get accuracies of each model.

- Then we will deploy our own model which will in turn be efficient.
- We will combine all this into a web application for better understanding and making user interaction descent.

## Tools and Technologies Used

- **PyCharm Community**
  PyCharm is a devoted Python Integrated Development Environment (IDE) giving a wide scope of fundamental apparatuses for Python designers, firmly incorporated to establish an advantageous climate for useful Python, web, and information science improvement.

- **Scikit-Learn**
  Scikit-learn (Sklearn) is the most valuable and powerful library for AI in Python. It gives a determination of productive instruments for AI and measurable displays including characterization, relapse, grouping and dimensionality decrease through a consistent interface in Python.

- **TensorFlow**
  TensorFlow is a start to finish open source stage for AI. It has a far reaching, adaptable biological system of devices, libraries and local area assets that allows specialists to push the cutting edge in ML and designers effectively fabricate and convey ML fueled applications.

- **Streamlit**
  Streamlit is an open-source Python library that makes it easy to create and share beautiful, custom web apps for machine learning and data science.

# CHAPTER 2: LITERATURE SURVEY

In the 2021 Financial Budget, medical care was declared as a first concern for the public authority, with an astounding increment of 137% over the earlier year's health sector's budget. The health sector gave off an impression of being a significant winner in the financial plan. A whopping cost of US$ 31,030 million.

Despite different focus supported projects and medical coverage conspires, the public authority wellbeing consumption has shifted from 1.2% to 1.6% of the GDP as of late. Albeit last year the govt healthcare consumption was 1.8% of GDP, this is amazingly low when contrasted with different nations on the planet, like Sri Lanka, China, Thailand, the United Kingdom and the USA. Indeed, even the objective imagined by the National Health Policy (Ministry of Health and Family Welfare, 2017) is to accomplish spending of 2.5% of its GDP on medical services by 2025!

The necessary information was extricated from patients' records utilizing an information assortment sheet.

At last, the substance legitimacy of the agenda was affirmed by specialists of wellbeing data, the executives and the emergency clinic supervisory group. Information assortment sheet included factors such as sex, age, and patient's home as segment attributes, also emergency clinic factors like day, time, and sort of confirmation, going to doctor, length of medical clinic stay, and so forth Likewise, some of the removed quantitative factors included a number of medical procedures, indicative administrations, and clinical advising. Information was gathered through investigation of patients' paper records and now and again, utilizing the Hospital Information System (HIS), by means of the presence of the analyst in the clinical records unit of the emergency clinic.

# PROJECT DESIGN

**Dataset Description**

Source: https://www.kaggle.com/nehaprabhavalkar/av-healthcare-analytics-ii

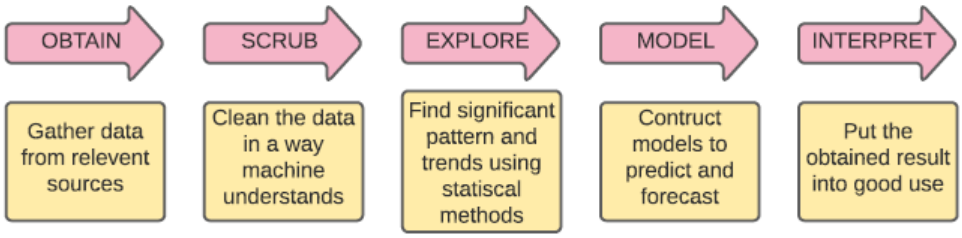
```
df_train = pd.read_csv("/kaggle/input/av-healthcare-analytics-ii/healthcare/train_data.csv",index_col=0)
df_test = pd.read_csv("/kaggle/input/av-healthcare-analytics-ii/healthcare/test_data.csv",index_col=0)
print(df_train.shape)
print(df_test.shape)
```

```
(318438, 17)
(137057, 16)
```

#null values

```
pd.isna(df_train).sum()
```

```
Hospital_code                      0
Hospital_type_code                 0
City_Code_Hospital                 0
Hospital_region_code               0
Available Extra Rooms in Hospital  0
Department                         0
Ward_Type                          0
Ward_Facility_Code                 0
Bed Grade                        113
patientid                          0
City_Code_Patient               4532
Type of Admission                  0
Severity of Illness                0
Visitors with Patient              0
Age                                0
Admission_Deposit                  0
Stay                               0
dtype: int64
```




**Workflow**

**Fig 1. Project Design**

# CHAPTER 3: THEORY

## Model Description

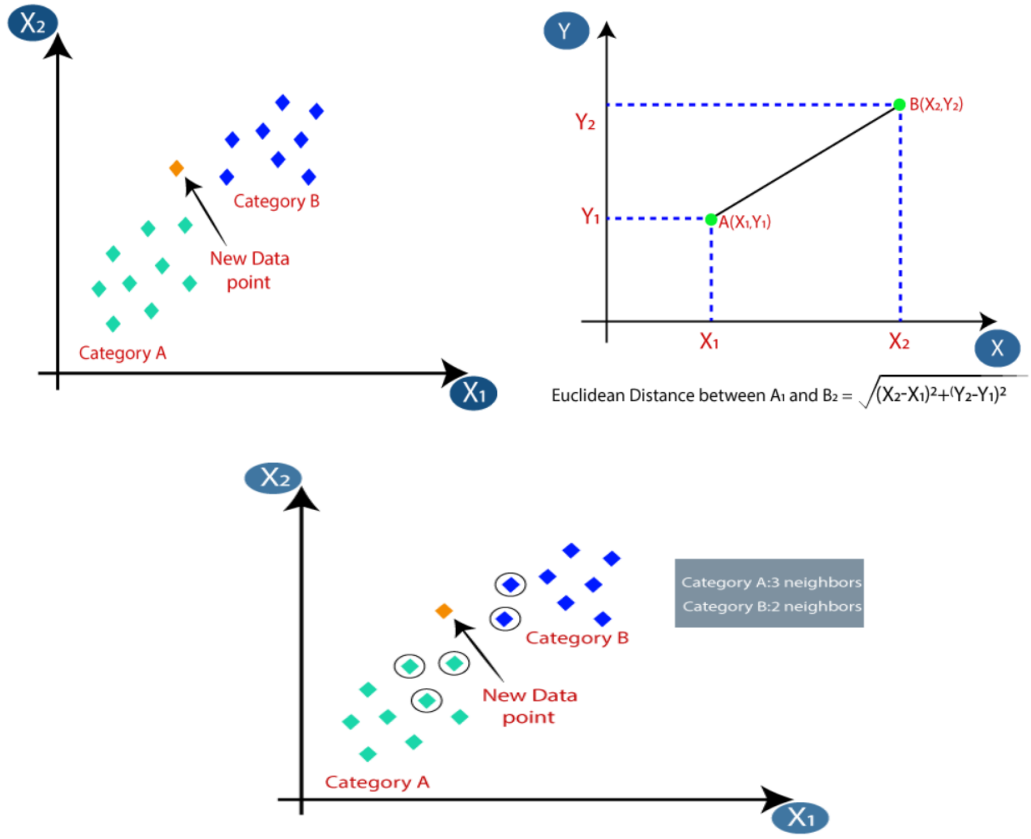
### K Neighbour Classifier

- K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.
- K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
- K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
- K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data.
- It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.
- Example: Suppose, we have an image of a creature that looks similar to cat and dog, but we want to know either it is a cat or dog. So for this identification, we can use the KNN algorithm, as it works on a similarity measure. Our KNN model will find the similar features of the new data set to the cats and dogs images and based on the most similar features it will put it in either cat or dog category.

**ALGORITHM**

1. Select the number K of the neighbors

2. Calculate the Euclidean distance of K number of neighbors
3. Take the K nearest neighbors as per the calculated Euclidean distance.
4. Among these k neighbors, count the number of the data points in each category.
5. Assign the new data points to that category for which the number of the neighbor is maximum.




**SELECT 'K'**

1. There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
2. A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
3. Large values for K are good, but it may have some difficulties.

**ADVANTAGES**

- It is simple to implement.
- It is robust to the noisy training data
- It can be more effective if the training data is large.

**DISADVANTAGES**

- Always needs to determine the value of K which may be complex some time.
- The computation cost is high because of calculating the distance between the data points for all the training samples.

## Logistics Regression

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas Logistic regression is used for solving the classification problems.
- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc.
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using continuous and discrete datasets.

- Logistic Regression can be used to classify the observations using different types of data and can easily determine the most effective variables used for the classification. The below image is showing the logistic function.

**Assumptions for Logistic Regression:**

- The dependent variable must be categorical in nature.
- The independent variable should not have multicollinearity.

**Logistic Regression Equation:**

The Logistic regression equation can be obtained from the Linear Regression equation. The mathematical steps to get Logistic Regression equations are given below:

- We know the equation of the straight line can be written as:

$$y= b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

- In Logistic Regression y can be between 0 and 1 only, so for this let's divide the above equation by (1-y):

$$\frac{y}{1-y}\text{; 0 for y= 0, and infinity for y=1}$$

- But we need range between -[infinity] to +[infinity], then take logarithm of the equation it will become:

$$log\left[\frac{y}{1-y}\right] = b_0 + b_1x_1 + b_2x_2 + b_3x_3 + \cdots + b_nx_n$$

The above equation is the final equation for Logistic Regression.

Type of Logistic Regression:

On the basis of the categories, Logistic Regression can be classified into three types:

- Binomial: In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.

- Multinomial: In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
- Ordinal: In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

**Advantages**

- Logistic Regression is **one of the simplest machine learning algorithms** and is easy to implement yet provides great training efficiency in some cases. Also due to these reasons, training a model with this algorithm doesn't require high computation power.
- The predicted parameters (trained weights) give **inference about the importance of each feature**. The direction of association i.e. positive or negative is also given. So we can use logistic regression to find out the relationship between the features.
- This algorithm allows models to be **updated easily to reflect new data**, unlike decision trees or support vector machines. The update can be done using stochastic gradient descent.
- Logistic Regression **outputs well-calibrated probabilities** along with classification results. This is an advantage over models that only give the final classification as results. If a training example has a 95% probability for a class, and another has a 55% probability for the same class, we get an inference about which training examples are more accurate for the formulated problem.
- In a **low dimensional dataset** having a sufficient number of training examples, logistic regression is **less prone to over-fitting**.

**Disadvantages**

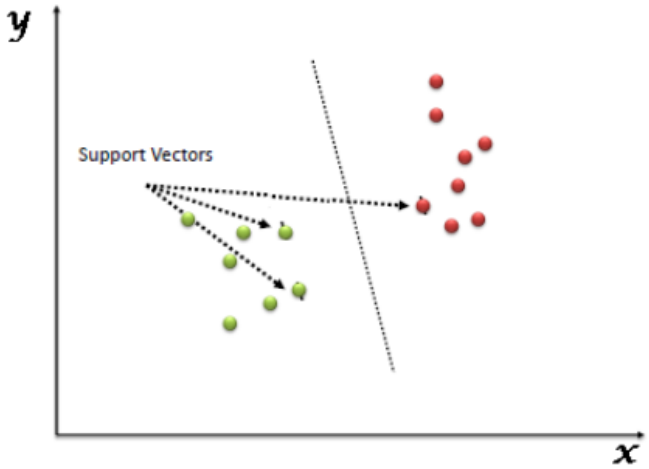
- Logistic Regression is a statistical analysis model that attempts to predict precise probabilistic outcomes based on independent features. On **high dimensional datasets**, this may lead to the model being **over-fit on the training set**, which means overstating the accuracy of predictions on the training set and thus the model **may not be able to predict accurate results on the test set**. This usually happens in the case when the

model is trained on little training data with lots of features. So on high dimensional datasets, Regularization techniques should be considered to avoid over-fitting (but this makes the model complex). Very high regularization factors may even lead to the model being under-fit on the training data.

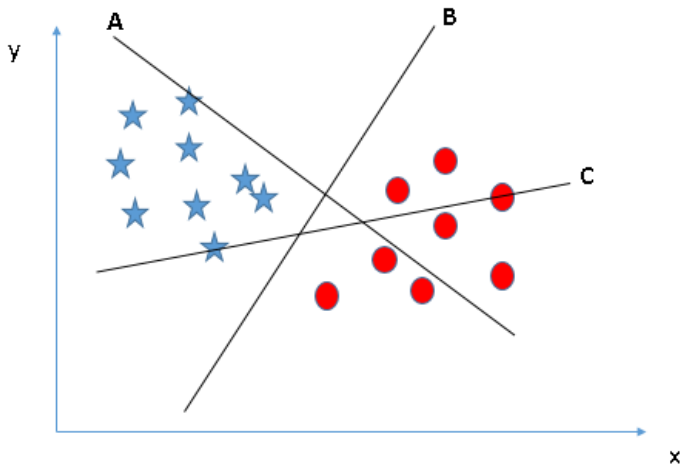
- **Non linear problems can't be solved** with logistic regression **since it has a linear decision surface**. Linearly separable data is rarely found in real world scenarios. So the transformation of non linear features is required which can be done by increasing the number of features such that the data becomes linearly separable in higher dimensions.

## Support Vector Machine(Linear Kernel)

"Support Vector Machine" (SVM) is a supervised machine learning algorithm that can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is a number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).
Linear Kernel is used when the data is Linearly separable, that is, it can be separated using a single Line. It is one of the most common kernels to be used. It is mostly used when there are a Large number of Features in a particular Data Set. One of the examples where there are a lot of features, is Text Classification, as each alphabet is a new feature. So we mostly use Linear Kernel in Text Classification.

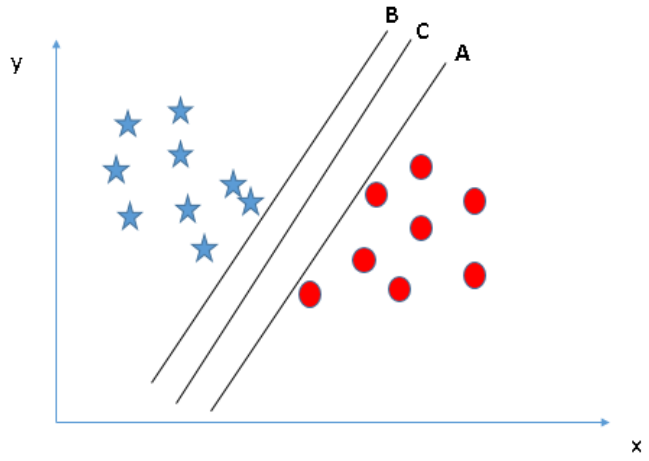



- Identify the right hyper-plane (Scenario-1): Here, we have three hyper-planes (A, B, and C). Now, identify the right hyper-plane to classify stars and circles.

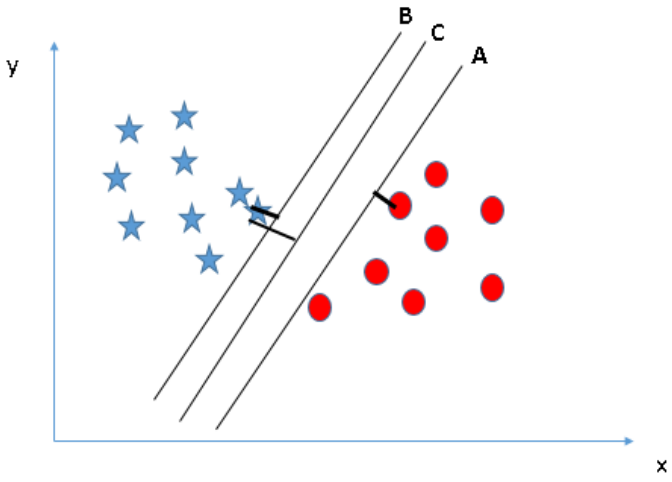



- You need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.
- Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B, and C) and all are segregating the classes well. Now, How can we identify the right

hyper-plane?




- Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called Margin. Let's look at the below snapshot:

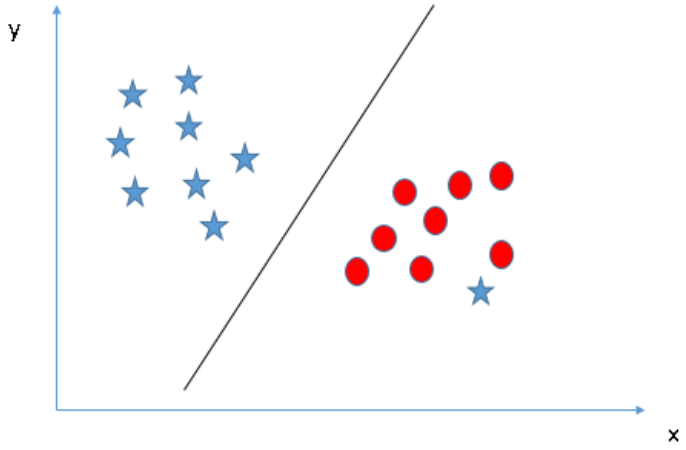



Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is a high chance of miss-classification.

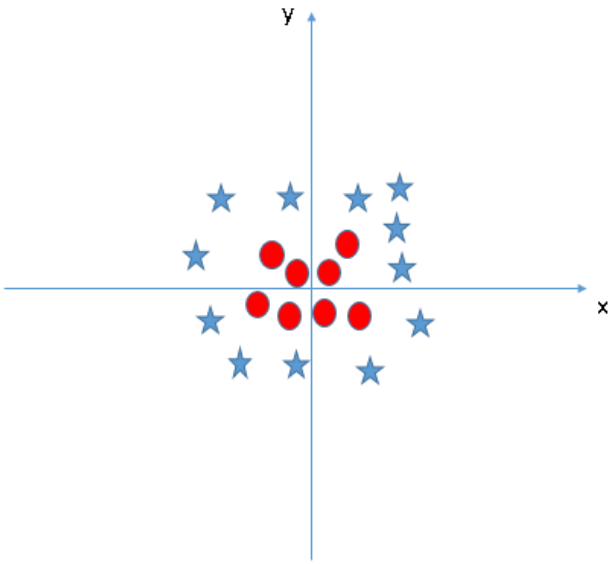
- Identify the right hyper-plane (Scenario-3):Hint: Use the rules as discussed in previous section to identify the right hyper-plane

- Some of you may have selected the hyper-plane B as it has higher margin compared to A. But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is A.
- Can we classify two classes (Scenario-4)?: Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.

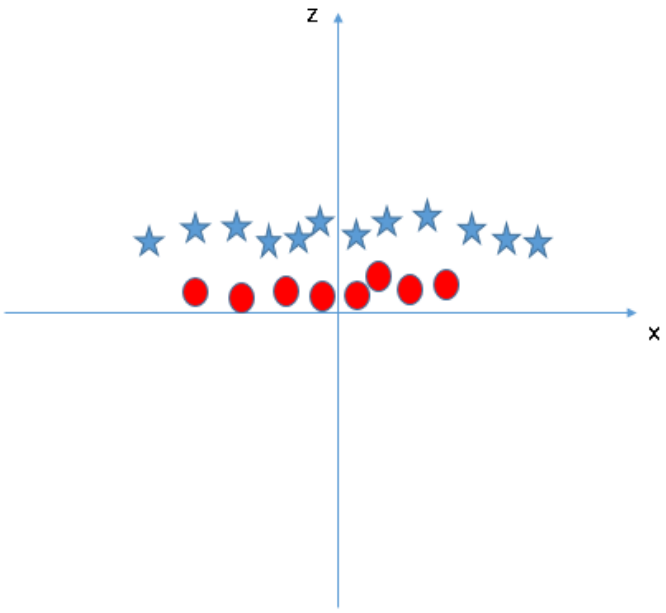

- As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.



- Find the hyper-plane to segregate to classes (Scenario-5): In the scenario below, we can't have linear hyper-plane between the two classes, so how does SVM classify these two classes? Till now, we have only looked at the linear hyper-plane.




- SVM can solve this problem. Easily! It solves this problem by introducing additional feature. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:




In above plot, points to consider are:

- All values for z would be positive always because z is the squared sum of both x and y
- In the original plot, red circles appear close to the origin of x and y axes, leading to lower value of z and star relatively away from the origin result to higher value of z.
- In the SVM classifier, it is easy to have a linear hyper-plane between these two classes. But, another burning question which arises is, should we need to add this feature manually to have a hyper-plane. No, the SVM algorithm has a technique called the kernel trick. The SVM kernel is a function that takes low dimensional input space and transforms it to a higher dimensional space i.e. it converts not separable problem to separable problem. It is mostly useful in non-linear separation problem. Simply put, it does some extremely complex data transformations, then finds out the process to separate the data based on the labels or outputs you've defined.

## Support Vector Machine(RBF Kernel)

In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification.

The RBF kernel on two samples x and x', represented as feature vectors in some input space, is defined as

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

$\|\mathbf{x} - \mathbf{x}'\|^2$ may be recognized as the squared Euclidean distance between the two feature vectors.

$\sigma$ is a free parameter. An equivalent definition involves a parameter $\gamma = \frac{1}{2\sigma^2}$:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma\|\mathbf{x} - \mathbf{x}'\|^2)$$

Since the value of the RBF kernel decreases with distance and ranges between zero (in the limit) and one (when x = x'), it has a ready interpretation as a similarity measure. The feature space of the kernel has an infinite number of dimensions; for $\sigma = 1$, its expansion is:

$$\begin{aligned}
\exp\left(-\frac{1}{2}\|\mathbf{x} - \mathbf{x}'\|^2\right) &= \exp(\frac{2}{2}\mathbf{x}^\top\mathbf{x}' - \frac{1}{2}\|\mathbf{x}\|^2 - \frac{1}{2}\|\mathbf{x}'\|^2) \\
&= \exp(\mathbf{x}^\top\mathbf{x}')\exp(-\frac{1}{2}\|\mathbf{x}\|^2)\exp(-\frac{1}{2}\|\mathbf{x}'\|^2) \\
&= \sum_{j=0}^{\infty}\frac{(\mathbf{x}^\top\mathbf{x}')^j}{j!}\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)\exp\left(-\frac{1}{2}\|\mathbf{x}'\|^2\right) \\
&= \sum_{j=0}^{\infty}\sum_{\sum n_i = j}\exp\left(-\frac{1}{2}\|\mathbf{x}\|^2\right)\frac{x_1^{n_1}\cdots x_k^{n_k}}{\sqrt{n_1!\cdots n_k!}}\exp\left(-\frac{1}{2}\|\mathbf{x}'\|^2\right)\frac{{x'}_1^{n_1}\cdots {x'}_k^{n_k}}{\sqrt{n_1!\cdots n_k!}}
\end{aligned}$$

**Approximations**

Because support vector machines and other models employing the kernel trick do not scale well to large numbers of training samples or large numbers of features in the input space, several approximations to the RBF kernel (and similar kernels) have been introduced. Typically, these take the form of a function z that maps a single vector to a vector of higher dimensionality, approximating the kernel:

$$\langle z(\mathbf{x}), z(\mathbf{x}')\rangle \approx \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}')\rangle = K(\mathbf{x}, \mathbf{x}')$$

where $\varphi$ is the implicit mapping embedded in the RBF kernel.
One way to construct such a z is to randomly sample from the Fourier transformation of the kernel.[1] Another approach uses the Nyström method to approximate the eigendecomposition of the Gram matrix K, using only a random sample of the training set.

## Gaussian NB

When we work with statistics and specifically probabilities, Gaussian Naive Bayes law is one of the most popular and fascinating theorems to dive into. The Bayesian theorem when working with statistics will allow you to calculate the probability that an event will occur provided that you have prior knowledge and information related to the specific event. Converted from text into a mathematical representation, the Gaussian Naive Bayes theorem appears as follows:

$$P(X|Y) = \frac{P(Y|X)\,P(X)}{P(Y)}$$

Where:

- X and Y are two independent events, and P(Y) is not equal to zero- because we are dealing with a fraction formula, division by zero leads to no real answer.
- P(X|Y) is the probability that event X will occur if event Y is true.
- P(Y|X) is the probability that event Y will occur if event X is true.
- P(X) and P(Y) is the probability that event X and event Y will occur independently of each other.

A Gaussian Naive Bayes algorithm is a special type of NB algorithm. It's specifically used when the features have continuous values. It's also assumed that all the features are following a gaussian distribution i.e, normal distribution.
Gaussian Naive Bayes

Naive Bayes can be extended to real-valued attributes, most commonly by assuming a Gaussian distribution.

This extension of naive Bayes is called Gaussian Naive Bayes. Other functions can be used to estimate the distribution of the data, but the Gaussian (or Normal distribution) is the easiest to

work with because you only need to estimate the mean and the standard deviation from your training data.

**Representation for Gaussian Naive Bayes**

Above, we calculated the probabilities for input values for each class using a frequency. With real-valued inputs, we can calculate the mean and standard deviation of input values (x) for each class to summarize the distribution.

This means that in addition to the probabilities for each class, we must also store the mean and standard deviations for each input variable for each class.

Learn a Gaussian Naive Bayes Model From Data

This is as simple as calculating the mean and standard deviation values of each input variable (x) for each class value.

$$mean(x) = 1/n * sum(x)$$

Where n is the number of instances and x are the values for an input variable in your training data.

We can calculate the standard deviation using the following equation:

$$standard\ deviation(x) = sqrt(1/n * sum(xi-mean(x)^2 ))$$

This is the square root of the average squared difference of each value of x from the mean value of x, where n is the number of instances, sqrt() is the square root function, sum() is the sum function, xi is a specific value of the x variable for the i'th instance and mean(x) is described above, and ^2 is the square.

**Make Predictions With a Gaussian Naive Bayes Model**

Probabilities of new x values are calculated using the Gaussian Probability Density Function (PDF).

When making predictions these parameters can be plugged into the Gaussian PDF with a new input for the variable, and in return the Gaussian PDF will provide an estimate of the probability of that new input value for that class.

$$pdf(x, mean, sd) = (1 / (sqrt(2 * PI) * sd)) * exp(-((x-mean^2)/(2*sd^2)))$$

Where pdf(x) is the Gaussian PDF, sqrt() is the square root, mean and sd are the mean and standard deviation calculated above, PI is the numerical constant, exp() is the numerical constant e or Euler's number raised to power and x is the input value for the input variable.

We can then plug in the probabilities into the equation above to make predictions with real-valued inputs.

For example, adapting one of the above calculations with numerical values for weather and car:

$$go\text{-}out = P(pdf(weather)|class=go\text{-}out) * P(pdf(car)|class=go\text{-}out) * P(class=go\text{-}out)$$

**Best Prepare Your Data For Naive Bayes**

- **Categorical Inputs**: Naive Bayes assumes label attributes such as binary, categorical or nominal.
- **Gaussian Inputs**: If the input variables are real-valued, a Gaussian distribution is assumed. In which case the algorithm will perform better if the univariate distributions of your data are Gaussian or near-Gaussian. This may require removing outliers (e.g. values that are more than 3 or 4 standard deviations from the mean).
- **Classification Problems**: Naive Bayes is a classification algorithm suitable for binary and multiclass classification.
- **Log Probabilities**: The calculation of the likelihood of different class values involves multiplying a lot of small numbers together. This can lead to an underflow of numerical precision. As such it is good practice to use a log transform of the probabilities to avoid this underflow.

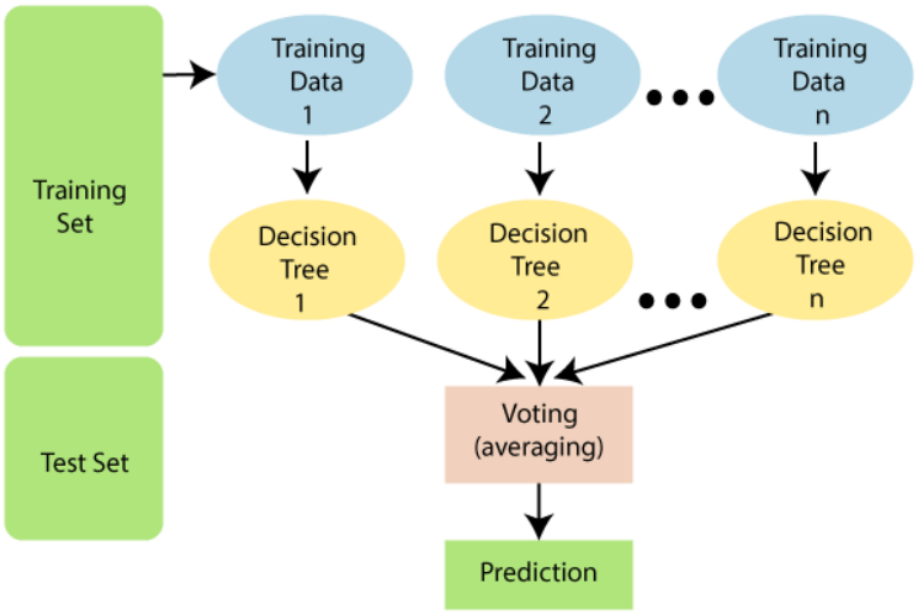
- **Kernel Functions**: Rather than assuming a Gaussian distribution for numerical input values, more complex distributions can be used such as a variety of kernel density functions.
- **Update Probabilities**: When new data becomes available, you can simply update the probabilities of your model. This can be helpful if the data changes frequently.

## Random Forest

Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique. It can be used for both Classification and Regression problems in ML. It is based on the concept of ensemble learning, which is a process of *combining multiple classifiers to solve a complex problem and to improve the performance of the model.*

As the name suggests, *"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."* Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.

The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.




**Assumptions for Random Forest**

Since the random forest combines multiple trees to predict the class of the dataset, it is possible that some decision trees may predict the correct output, while others may not. But together, all the trees predict the correct output. Therefore, below are two assumptions for a better Random forest classifier:

- There should be some actual values in the feature variable of the dataset so that the classifier can predict accurate results rather than a guessed result.
- The predictions from each tree must have very low correlations.

**Why use Random Forest?**

Below are some points that explain why we should use the Random Forest algorithm:

- It takes less training time as compared to other algorithms.
- It predicts output with high accuracy, even for the large dataset it runs efficiently.
- It can also maintain accuracy when a large proportion of data is missing.

**How does the Random Forest algorithm work?**

Random Forest works in two-phase first is to create the random forest by combining N decision trees, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

- Select random K data points from the training set.
- Build the decision trees associated with the selected data points (Subsets).
- Choose the number N for decision trees that you want to build.
- Repeat Step 1 & 2.
- For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

**Advantages of Random Forest**

- Random Forest is capable of performing both Classification and Regression tasks.
- It is capable of handling large datasets with high dimensionality.
- It enhances the accuracy of the model and prevents the overfitting issue.

**Disadvantages of Random Forest**

- Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.

## Perceptron

A perceptron model, in Machine Learning, is a supervised learning algorithm of binary classifiers. A single neuron, the perceptron model detects whether any function is an input or not and classifies them in either of the classes.
Representing a biological neuron in the human brain, the perceptron model or simply a perceptron acts as an artificial neuron that performs human-like brain functions. A linear ML

algorithm, the perceptron conducts binary classification or two-class categorization and enables neurons to learn and register information procured from the inputs.

This model uses a hyperplane line that classifies two inputs and classifies them on the basis of the 2 classes that a machine learns, thus implying that the perceptron model is a linear classification model. Invented by Frank Rosenblatt in 1957, the perceptron model is a vital element of Machine Learning as ML is recognized for its classification purposes and mechanism.

There are 4 constituents of a perceptron model. They are as follows-

1. Input values
2. Weights and bias
3. Net sum
4. Activation function

The perceptron model enables machines to automatically learn coefficients of weight which helps in classifying the inputs. Also recognized as the Linear Binary Classifier, the perceptron model is extremely efficient and helpful in arranging the input data and classifying the same in different classes.

Understanding the Perceptron

As we have already learned about the perceptron model in Machine Learning, we will now understand the model in a bit more detail. Let's get started!

The simplest variant of artificial neuron networks, the perceptron model resembles a biological neuron that simply helps in the linear binary classification with the help of a hyperplane line.

There are 2 types of perceptron models-

1. Single Layer Perceptron- The Single Layer perceptron is defined by its ability to linearly classify inputs. This means that this kind of model only utilizes a single hyperplane line and classifies the inputs as per the learned weights beforehand.
2. Multi-Layer Perceptron- The Multi-Layer Perceptron is defined by its ability to use layers while classifying inputs. This type is a high processing algorithm that allows machines to classify inputs using various more than one layer at the same time.

The working of the model is based on the Perceptron Learning Rule that implies that the algorithm is enabled to automatically learn respective coefficients of weights that designate several inputs.

The perceptron model registers inputs with the machine and allots them with certain weights as per the coefficients that lead a particular input into a specific class. This is decided on the basis of the final value derived by calculating the net sum and activation function at the end stages.

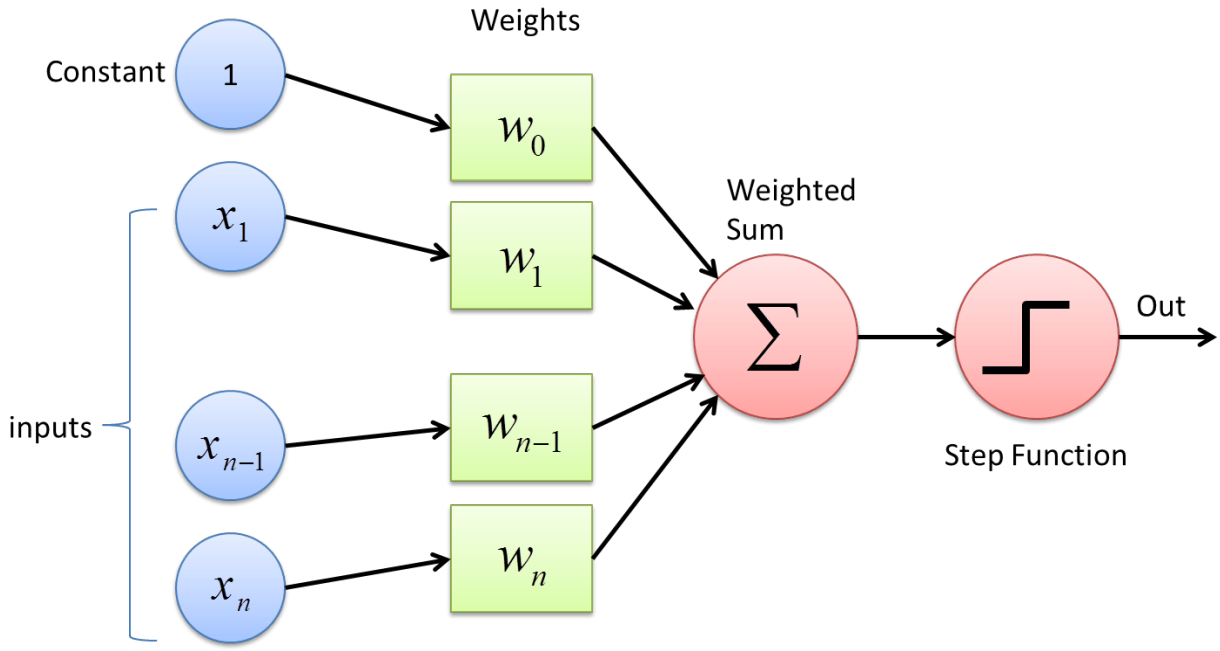
Let us now go through a step-by-step procedure in order to understand the way the perceptron model operates.

1. Enter bits of information that are supposed to serve as inputs in the first layer (Input Value).
2. All weights (pre-learned coefficients) and input values will be multiplied. The multiplied values of all input values will be added.
3. The bias value will shift to the final stage (activation function/output result).
4. The weighted input will proceed to the stage of the activation function. The bias value will be now added.
5. The value procured will be the output value that will determine if the output will be released or not.

The perceptron algorithm, using the Heaviside activation function is summarised as follows-

$f(z) = \{1 \text{ if } xTw+b > 0$

$= \{0 \text{ otherwise}$




The Input value of the model consists of various artificial neurons in artificial intelligence that facilitate the entry of data into the system or machine.

When the inputs are registered in the machine, the perceptron algorithm primarily applies the already learned value of weight (dimension or strength of the connection between data units). These weights are then multiplied with the input values and headed to the net sum (total value). Ultimately, the input value proceeds to the activation function where output is released or scrapped out. The activation function (weighted sum total added with bias) in the final stage is important for determining if an input's value is greater than 0.

The process that enables the perceptron model to conduct mathematical operations for converting input into output is called training. As the process of training is implemented in the working of the perceptron model wherein machines are made fully capable of calculating output values even without being fed with input values.

The process of training involves feeding machines with historic data in order to prepare them for the future and instill predictive patterns. Based on artificial neural networks that tend to imitate the human brain, the perceptron model works along the lines of machine learning as it continuously interprets data and produces qualitative patterns.

## SGD Classifier

Stochastic Gradient Descent (SGD) is a simple yet efficient optimization algorithm used to find the values of parameters/coefficients of functions that minimize a cost function. In other words, it is used for discriminative learning of linear classifiers under convex loss functions such as SVM and Logistic regression. It has been successfully applied to large-scale datasets because the update to the coefficients is performed for each training instance, rather than at the end of instances.

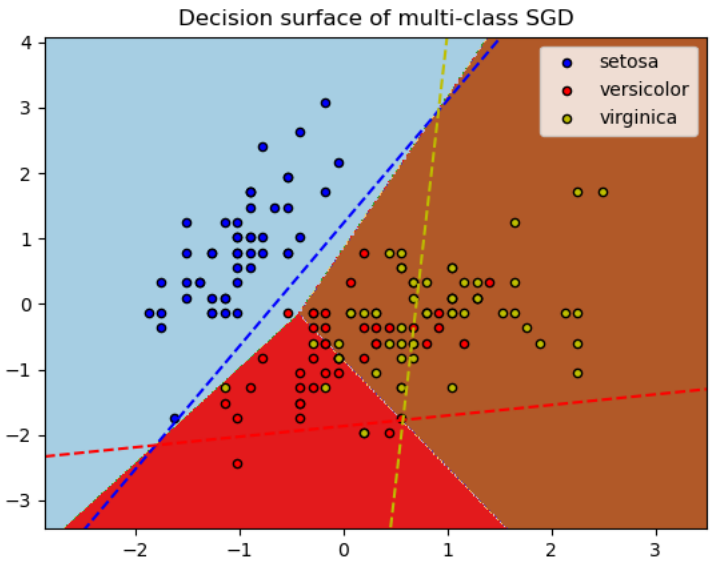
Stochastic Gradient Descent (SGD) classifier basically implements a plain SGD learning routine supporting various loss functions and penalties for classification. Scikit-learn provides SGDClassifier module to implement SGD classification.

Why do we use SGD classifiers, when we already have linear classifiers such as LogReg or SVM?

As we can read from the previous text, SGD allows minibatch (online/out-of-core) learning. Therefore, it makes sense to use SGD for large scale problems where it's very efficient.

The minimum of the cost function of Logistic Regression cannot be calculated directly, so we try to minimize it via Stochastic Gradient Descent, also known as Online Gradient Descent. In this process we descend along the cost function towards its minimum (please have a look at the diagram above) for each training observation we encounter.

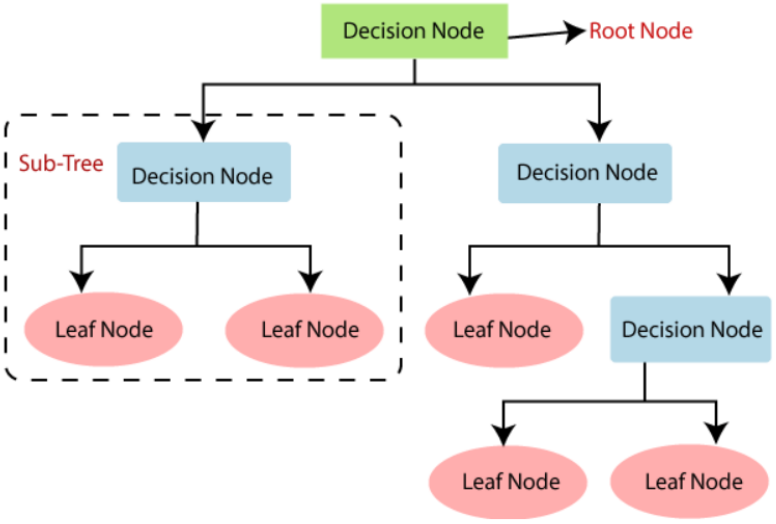
Another reason to use SGD Classifier is that SVM or logistic regression will not work if you cannot keep the record in RAM. However, SGD Classifier continues to work.




## Decision Tree Classifier

- Decision Tree is a Supervised learning technique that can be used for both classification and Regression problems, but mostly it is preferred for solving Classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome.
- In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches.
- The decisions or the test are performed on the basis of features of the given dataset.
- It is a graphical representation for getting all the possible solutions to a problem/decision based on given conditions.
- It is called a decision tree because, similar to a tree, it starts with the root node, which expands on further branches and constructs a tree-like structure.
- In order to build a tree, we use the CART algorithm, which stands for Classification and Regression Tree algorithm.

- A decision tree simply asks a question, and based on the answer (Yes/No), it further split the tree into subtrees.




Why use Decision Trees?

There are various algorithms in Machine learning, so choosing the best algorithm for the given dataset and problem is the main point to remember while creating a machine learning model. Below are the two reasons for using the Decision tree:

- Decision Trees usually mimic human thinking ability while making a decision, so it is easy to understand.
- The logic behind the decision tree can be easily understood because it shows a tree-like structure.

Decision Tree Terminologies

Root Node: Root node is from where the decision tree starts. It represents the entire dataset, which further gets divided into two or more homogeneous sets.

Leaf Node: Leaf nodes are the final output node, and the tree cannot be segregated further after getting a leaf node.

Splitting: Splitting is the process of dividing the decision node/root node into sub-nodes according to the given conditions.

Branch/Sub Tree: A tree formed by splitting the tree.

Pruning: Pruning is the process of removing the unwanted branches from the tree.

Parent/Child node: The root node of the tree is called the parent node, and other nodes are called the child nodes.

How does the Decision Tree algorithm Work?

In a decision tree, for predicting the class of the given dataset, the algorithm starts from the root node of the tree. This algorithm compares the values of root attribute with the record (real dataset) attribute and, based on the comparison, follows the branch and jumps to the next node. For the next node, the algorithm again compares the attribute value with the other sub-nodes and move further. It continues the process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm:

History of Java

- Begin the tree with the root node, says S, which contains the complete dataset.
- Find the best attribute in the dataset using Attribute Selection Measure (ASM).
- Divide the S into subsets that contain possible values for the best attributes.
- Generate the decision tree node, which contains the best attribute.
- Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and call the final node as a leaf node.

Attribute Selection Measures

While implementing a Decision tree, the main issue arises that how to select the best attribute for the root node and for sub-nodes. So, to solve such problems there is a technique which is called as Attribute selection measure or ASM. By this measurement, we can easily select the best attribute for the nodes of the tree. There are two popular techniques for ASM, which are:

- Information Gain
- Gini Index

1. Information Gain:

- Information gain is the measurement of changes in entropy after the segmentation of a dataset based on an attribute.
- It calculates how much information a feature provides us about a class.
- According to the value of information gain, we split the node and build the decision tree.

- A decision tree algorithm always tries to maximize the value of information gain, and a node/attribute having the highest information gain is split first. It can be calculated using the below formula:

1. Information Gain= Entropy(S)- [(Weighted Avg) *Entropy(each feature)

Entropy: Entropy is a metric to measure the impurity in a given attribute. It specifies randomness in data. Entropy can be calculated as:

Entropy(s)= -P(yes)log2 P(yes)- P(no) log2 P(no)

Where,

- S= Total number of samples
- P(yes)= probability of yes
- P(no)= probability of no

2. Gini Index:

- Gini index is a measure of impurity or purity used while creating a decision tree in the CART(Classification and Regression Tree) algorithm.
- An attribute with the low Gini index should be preferred as compared to the high Gini index.
- It only creates binary splits, and the CART algorithm uses the Gini index to create binary splits.
- Gini index can be calculated using the below formula:

Gini Index= $1- \sum_j P_j^2$

Pruning: Getting an Optimal Decision tree

Pruning is a process of deleting the unnecessary nodes from a tree in order to get the optimal decision tree.

A too-large tree increases the risk of overfitting, and a small tree may not capture all the important features of the dataset. Therefore, a technique that decreases the size of the learning tree without reducing accuracy is known as Pruning. There are mainly two types of tree pruning technology used:

- Cost Complexity Pruning

- Reduced Error Pruning.

Advantages of the Decision Tree

- It is simple to understand as it follows the same process which a human follow while making any decision in real-life.
- It can be very useful for solving decision-related problems.
- It helps to think about all the possible outcomes for a problem.
- There is less requirement of data cleaning compared to other algorithms.

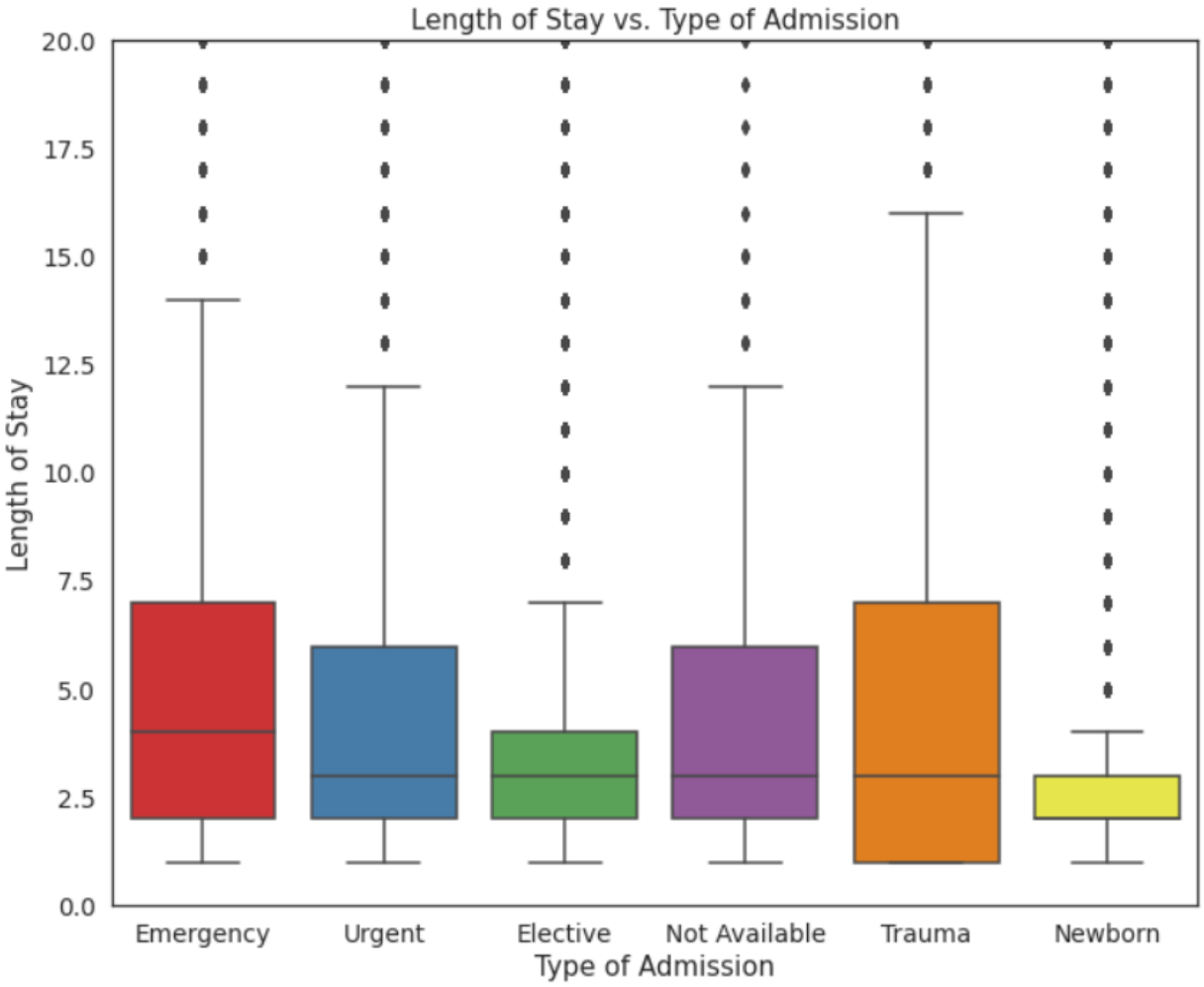
Disadvantages of the Decision Tree

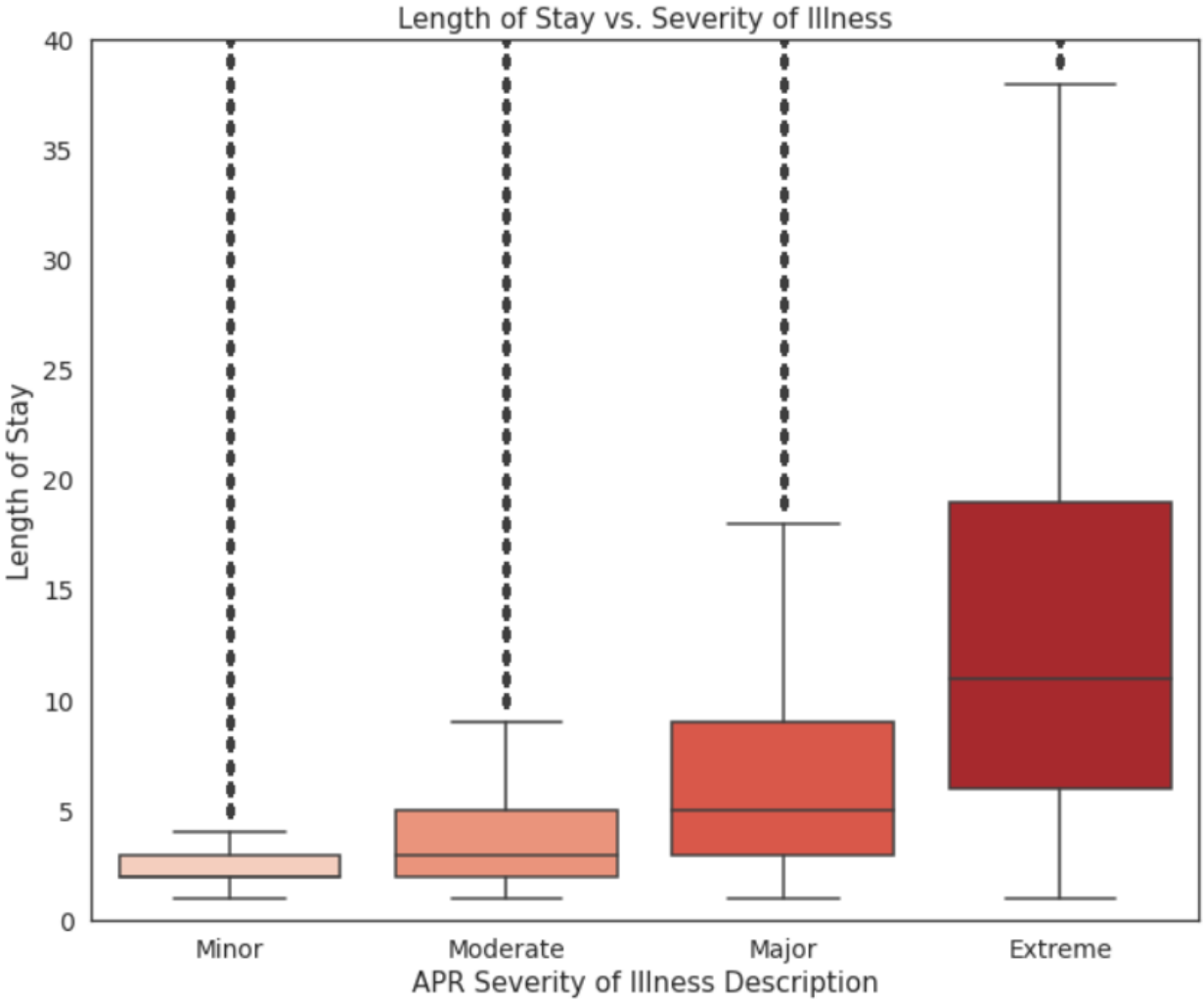
- The decision tree contains lots of layers, which makes it complex.
- It may have an overfitting issue, which can be resolved using the Random Forest algorithm.

## Webscraping an Additional Feature

In the brief discussion of health insurance programs above, I also mentioned patient income as potentially playing a role, and ultimately being a feature that might be correlated with length of stay. One example of how this might be the case is that for some low-income patients, the hospital provides better living conditions than their situation outside of the hospital. Thus, these patients are motivated to do everything in their ability to lengthen their stay. Unfortunately, our dataset does not include any information about patient income. However, it does include the 3 digit zipcode of the patient. As we prepare to begin the modeling section of this project, is important to consider the fact that zipcode is not a good feature for most ML models. Although its information is encoded by numbers, it is not a numerical feature, as its numbers do not mathematically mean anything. In turn, this makes zipcodes hard to understand and create patterns from for prediction by ML models. As a result, the next step in my analysis will be replacing this raw 3 digit zipcode data with a rough measure of patient income, while will prove a much more useful feature.

To achieve this, I webscraped average income data by zipcode from this website, which contains data from 2006–2010. Below is a plot of the median income for each 3 digit zipcode present in the dataset, achieved by inner joining (through SQL in python) the webscraped data onto our dataframe.


Length of Stay vs. Type of Admission
Length of Stay
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
Emergency
Urgent
Elective
Not Available
Trauma
Newborn
Type of Admission


Length of Stay vs. Severity of Illness
Length of Stay
0
5
10
15
20
25
30
35
40
Minor
Moderate
Major
Extreme
APR Severity of Illness Description

# CHAPTER 4: WORKING OF A PROJECT

## Data Exploration

st.write(train.head())

| | case_id | Hospital_code | Hospital_type_code | City_Code_Hospital | Available Extra Ro |
|---|---|---|---|---|---|
| 0 | 1 | 8 | c | 3 | |
| 1 | 2 | 2 | c | 5 | |
| 2 | 3 | 10 | e | 1 | |
| 3 | 4 | 26 | b | 2 | |
| 4 | 5 | 26 | b | 2 | |

st.table(load_train().isna().sum())

Check for Null values

| | 0 |
|---|---|
| case_id | 0 |
| Hospital_code | 0 |
| Hospital_type_code | 0 |
| City_Code_Hospital | 0 |
| Hospital_region_code | 0 |
| Available Extra Rooms in Hospital | 0 |
| Department | 0 |
| Ward_Type | 0 |
| Ward_Facility_Code | 0 |
| Bed Grade | 113 |
| patientid | 0 |
| City_Code_Patient | 4532 |
| Type of Admission | 0 |
| Severity of Illness | 0 |
| Visitors with Patient | 0 |
| Age | 0 |
| Admission_Deposit | 0 |
| Stay | 0 |

# Data Visualization

## Bivariate Analysis

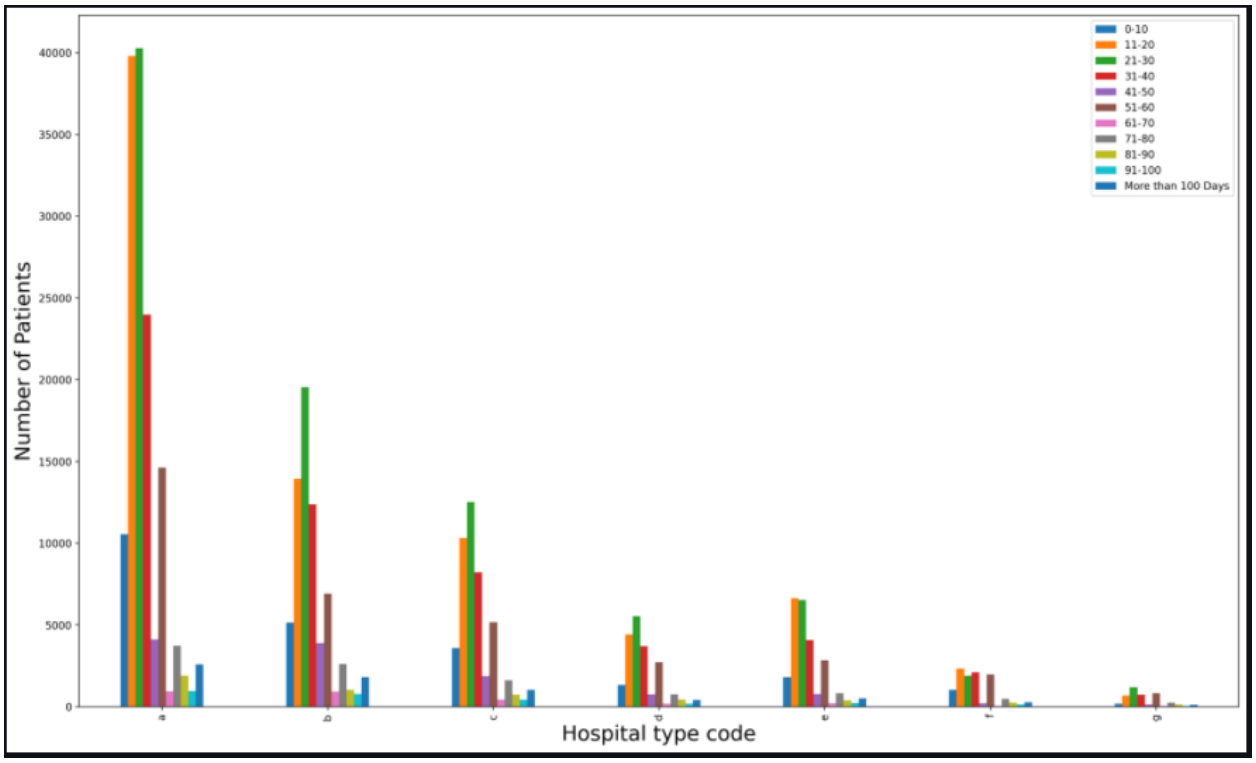



**Fig. 2 Bivariate Analysis**

## Data Wrangling

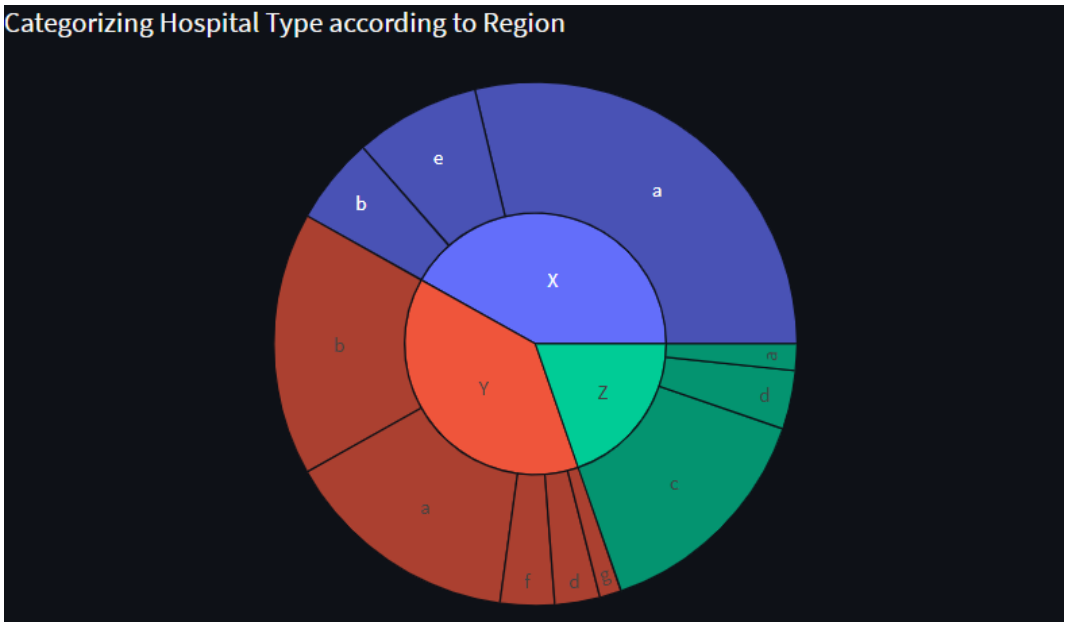



**Fig. 3 Hospital Type in accordance with Region**

## Available Extra Rooms

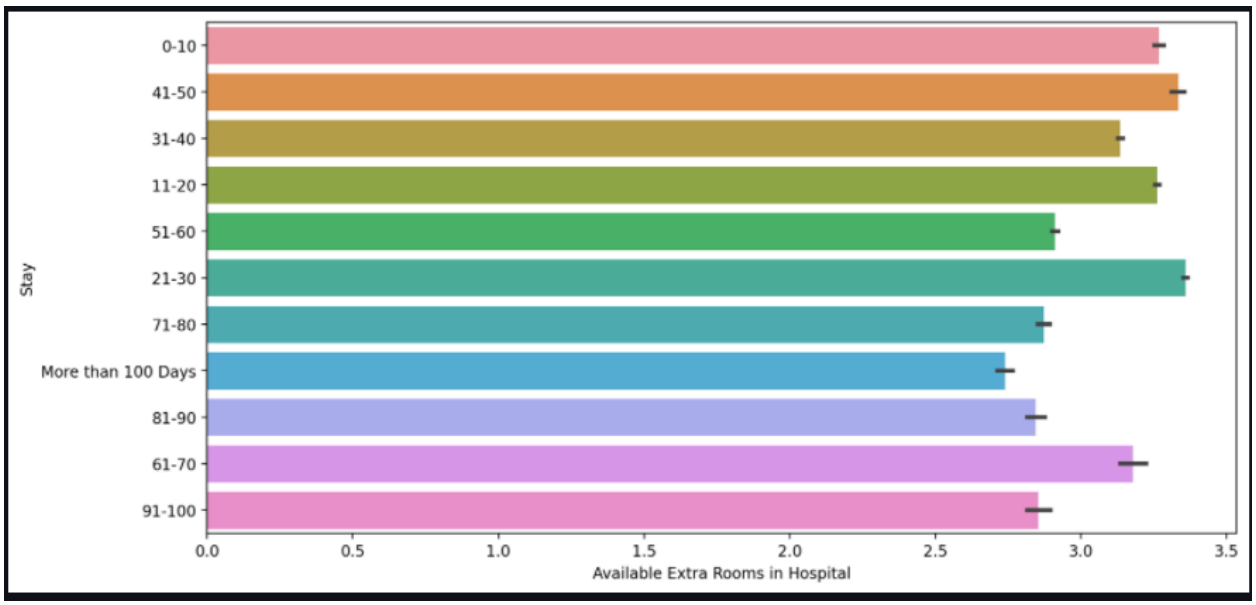



**Fig. 4 No of extra rooms vs Stay**

Patient Density per Department

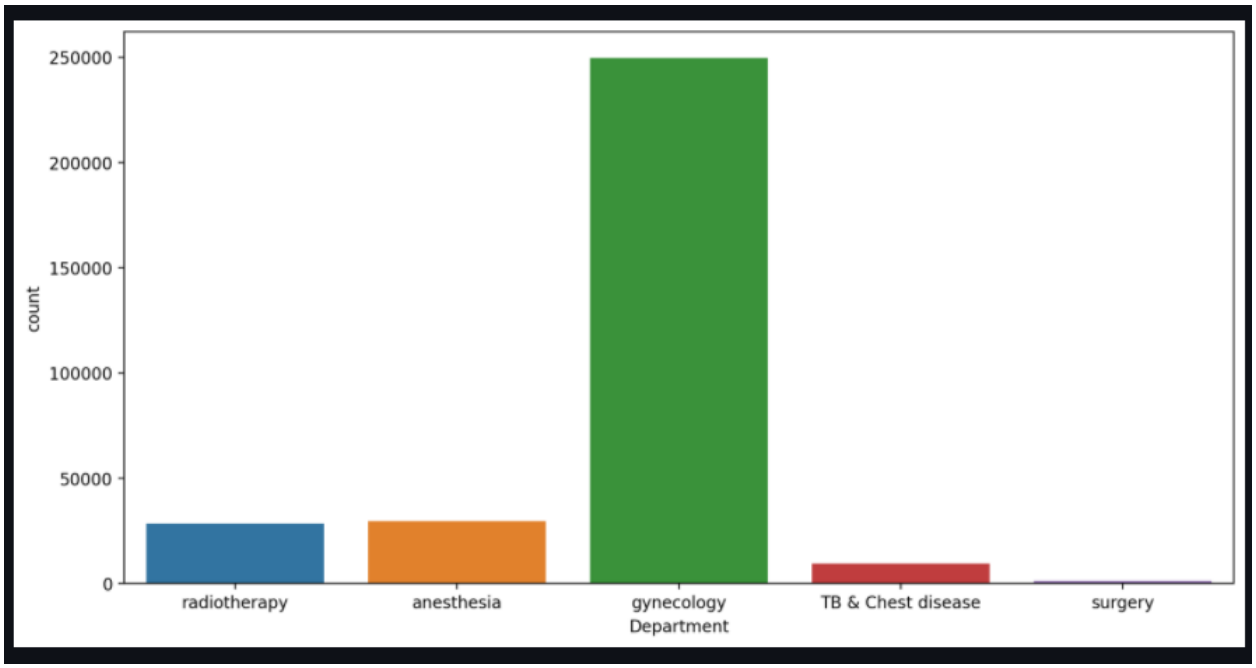
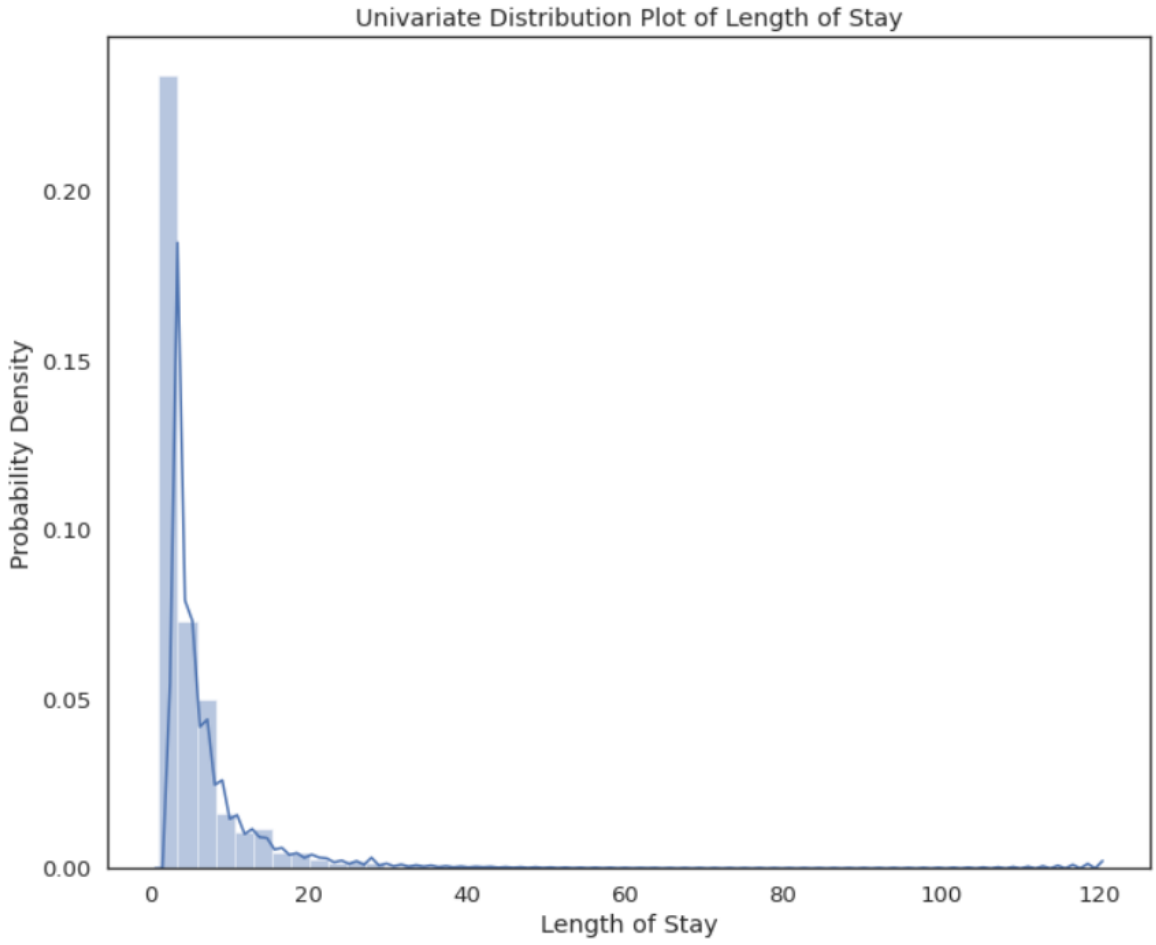



**Fig. 5 Patient Density per Department**




## Data Processing and Feature Engineering

We performed Data Processing through a function getting_combined(dataset) where we are filling missing data with LabelEncoder()

Also we are changing Age feature from the range(eg., 0-10 to 1, 11-20 to 2, and so on) for better model learning.

```
def getting_combined(combined):
  for dataset in combined:
    label = LabelEncoder()
    dataset['Department'] = label.fit_transform(dataset['Department'])
  for dataset in combined:
```

```
    label = LabelEncoder()
    dataset['Hospital_type_code'] = label.fit_transform(dataset['Hospital_type_code'])
    dataset['Ward_Facility_Code'] = label.fit_transform(dataset['Ward_Facility_Code'])
    dataset['Ward_Type'] = label.fit_transform(dataset['Ward_Type'])
    dataset['Type of Admission'] = label.fit_transform(dataset['Type of Admission'])
    dataset['Severity of Illness'] = label.fit_transform(dataset['Severity of Illness'])
age_dict = {'0-10': 0, '11-20': 1, '21-30': 2, '31-40': 3, '41-50': 4, '51-60': 5, '61-70': 6, '71-80': 7,
        '81-90': 8, '91-100': 9}
for dataset in combined:
    dataset['Age'] = dataset['Age'].replace(age_dict.keys(), age_dict.values())
return combined
```

## Predictive Analysis

We are using pre-trained models available from keras and performing model training and evaluating accuracy in each model. This way one can identify which among them are better suited for web apps like ours.

DECISION TREE

- In a Decision tree, there are two hubs, which are the Decision Node and Leaf Node. Choice hubs are utilized to settle on any choice and have different branches, while Leaf hubs are the result of those choices and don't contain any further branches.
- The choices or the test are performed based on elements of the given dataset.
- It is a graphical portrayal for getting every one of the potential answers for an issue/choice dependent on given conditions.
- It is known as a choice tree in light of the fact that, like a tree, it begins with the root hub, which develops further branches and builds a tree-like construction.
- A choice tree essentially poses an inquiry, and in view of the appropriate response.

DECISION TREE

- It depends on the idea of gathering realism, which is a course of joining different

classifiers to tackle an intricate issue and to work on the exhibition of the model.

- As the name recommends, Random Forest is a classifier that contains various choice trees on different subsets of the given dataset and takes the normal to work on the prescient exactness of that dataset.
- Rather than depending on one choice tree, the arbitrary backwoods takes the forecast from each tree and in view of the larger part votes of expectations, and it predicts the last result.
- The more prominent number of trees in the backwoods prompts higher exactness and forestalls the issue of overfitting.

SVM

- Support Vector Machine or SVM is perhaps the most famous Supervised Learning algorithm, which is utilized for Classification just as Regression issues.
- Nonetheless, principally, it is utilized for Classification issues in Machine Learning.
- The objective of the SVM calculation is to make the best line or choice limit that can isolate n-dimensional space into classes so we can undoubtedly put the new element in the right classification later on. This best choice limit is known as a hyperplane.
- SVM picks the outrageous focuses/vectors that assist in making the hyperplane.
- These outrageous cases are called help vectors, and consequently calculation is named a Support Vector Machine.

GAUSSIAN NAIVE BAYES

- A Gaussian Naive Bayes calculation is an exceptional kind of NB calculation. It's particularly utilized when the elements have ceaseless qualities.
- It's additionally expected that every one of the elements are following a gaussian conveyance i.e, typical appropriation.
- The guileless Bayes classifier expects every one of the elements to be autonomous to one another. Regardless of whether the highlights rely upon one another or upon the presence of different elements.
- Guileless Bayes classifier thinks about these properties to freely add to the likelihood that

the client purchases the MacBook.

K Nearest Neighbours

- K-Nearest Neighbor is one of the least complex Machine Learning calculations dependent on Supervised Learning procedure.
- K-NN calculation expects the similitude between the new case/information and accessible cases and puts the new case into the classification that is generally like the accessible classifications.
- K-NN calculation stores every one of the accessible information and characterizes another information point dependent on the similitude. This implies when new information shows up then it tends to be effectively ordered into a well suite classification by utilizing K-NN calculation.
- K-NN calculation can be utilized for Regression just as for Classification however generally it is utilized for the Classification issues.
- K-NN is a non-parametric calculation, which implies it doesn't make any presumption on fundamental information.
- It is likewise called a sluggish student calculation since it doesn't gain from the preparation set promptly rather it stores the dataset and at the hour of characterization, it plays out an activity on the dataset.

LOGISTIC REGRESSION

- It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.
- Logistic Regression is much similar to the Linear Regression except that how they are used. Linear Regression is used for solving Regression problems, whereas **Logistic**

**regression is used for solving the classification problems**.

- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1).
- Logistic Regression is a significant machine learning algorithm because it has the ability to provide probabilities and classify new data using and discrete datasets.

## Contribute to Dataset

- A simple layout to add extra values in the dataset.
- This includes a number of input values and different values.
- We have to select a layout which offers the best response to the user in terms of UI and experience.
- The data would be added to the csv file.
- We will create multiple variables where we insert all the values in these variables.
- Then, we will create a dataframe and add these values.

# CHAPTER 5: RESULTS AND DISCUSSIONS

We were able to develop a Web Application with Python and Streamlit. The app has a sidebar for navigation. It has four options namely, data exploration, data visualization. Prediction and contribution to dataset.

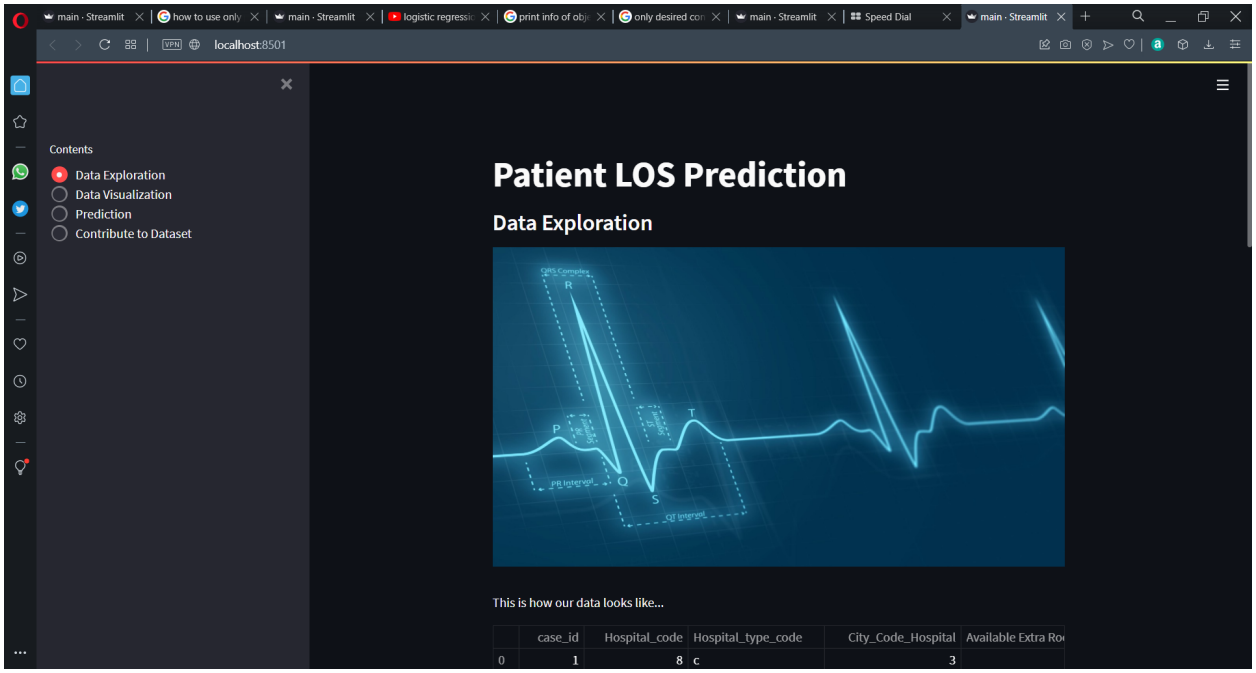



**Fig. 6 Data Exploration**

Now there are subsections to Data Visualization and Prediction which are represented by a slider menu.

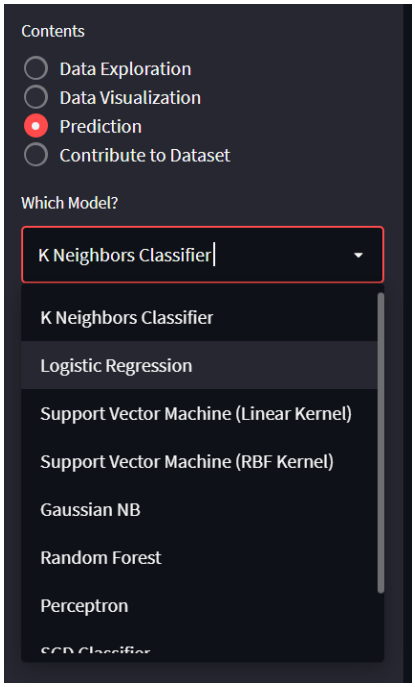



**Fig. 7 Navigation menu with scroll**

Figures are made beautifully with python libraries like matplotlib, seaborn and plotly.

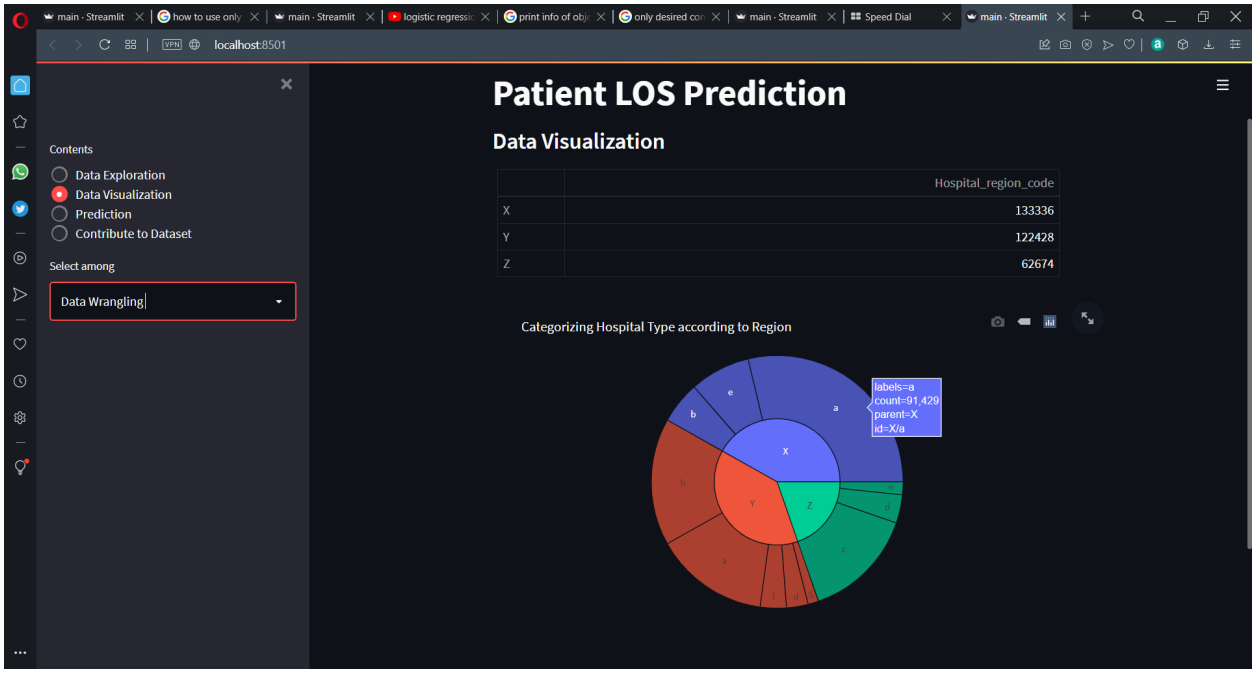



**Fig. 8 Data Visualization sample**

Then we obtained accuracies of many models like K Neighbors Classifier, Logistic Regression, Support Vector Machine (Linear Kernel), Support Vector Machine (RBF Kernel), Gaussian NB, Random Forest, Perceptron, SGD Classifier, Decision Tree Classifier.

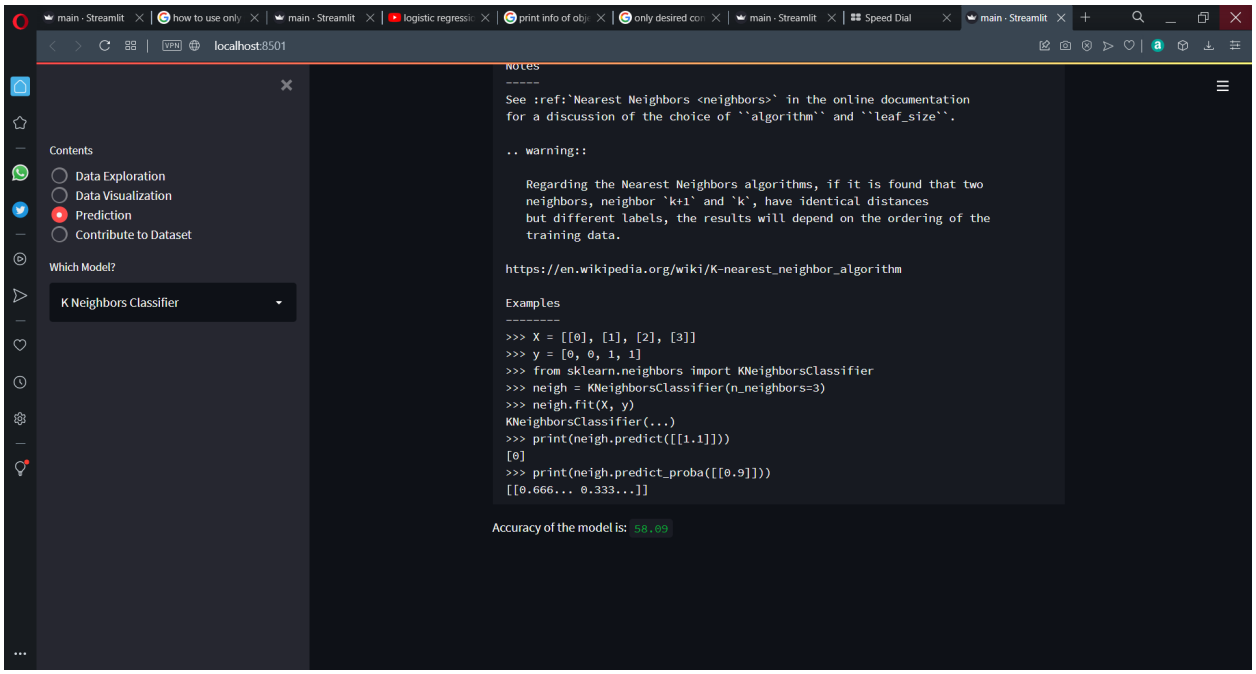
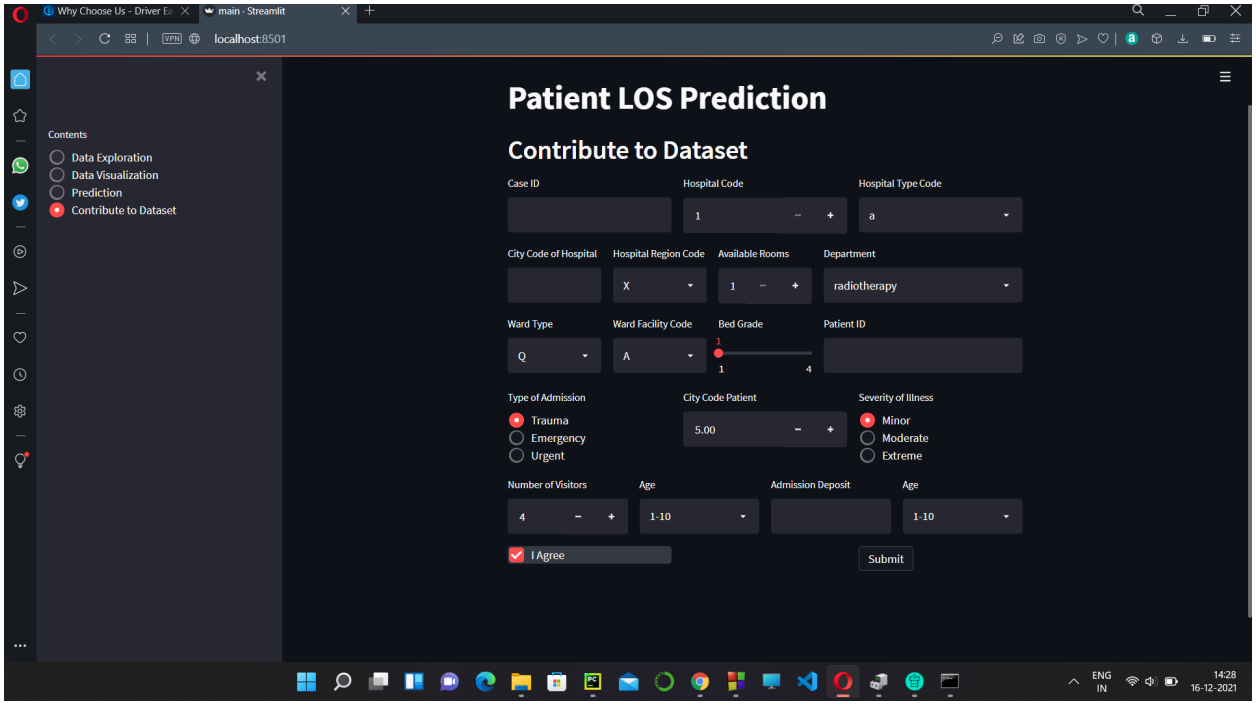



**Fig. 9 Getting Accuracy**

Then we will contribute to the dataset


Why Choose Us - Driver Ea
main - Streamlit
localhost:8501
Contents
Data Exploration
Data Visualization
Prediction
Contribute to Dataset
Patient LOS Prediction
Contribute to Dataset
Case ID
Hospital Code
1
Hospital Type Code
a
City Code of Hospital
Hospital Region Code
X
Available Rooms
1
Department
radiotherapy
Ward Type
Q
Ward Facility Code
A
Bed Grade
1
1
4
Patient ID
Type of Admission
Trauma
Emergency
Urgent
City Code Patient
5.00
Severity of Illness
Minor
Moderate
Extreme
Number of Visitors
4
Age
1-10
Admission Deposit
Age
1-10
I Agree
Submit
ENG
IN
14:28
16-12-2021

# CHAPTER 6: CONCLUSIONS AND FUTURE SCOPE

We made an honest approach towards solving a problem. We made an interactive web solution for any user to compare various models to solve the problem. The application shows various options in the sidebar, like exploring data, contributing to dataset, prediction, etc. Users can easily navigate through the application without a fuss. The application looks promising and holds a decent scope of development though.

1. Deploying the web app on online hostings like Heroku, Digital Ocean, etc.
2. Introducing and customizing the UI for better user experience.
3. Getting the patent for our project.

## FUTURE SCOPE

- Predicting real time data.
- Showing plots of various algorithms

## REFERENCES


- https://towardsdatascience.com/predicting-hospital-length-of-stay-at-time-of-admission-55dfdfe69598
- Predicting length of stay in hospitals intensive care unit using general admission features by Merhan A., Abd-Elrazeka, Ahmed A.Eltahawib, Mohamed H., Abd Elazizc, Mohamed N., and Abd-Elwhab
- Awad A, Bader-El-Den M, McNicholas J. Patient length of stay and mortality prediction: A survey. Health Serv Manage Res. 2017 May;30(2):105-120. doi: 10.1177/0951484817696212. Epub 2017 Mar 22. PMID: 28539083.
- Google Images