A

Practical Project Report

On

" MODELING ICU OCCUPANCY"

at

CELEBAL TECHNOLOGY, Jaipur

Submitted in Partial fulfilment of the requirement for the award of the degree of

Bachelor of Technology

in

Computer Engineering

(Session - 2020-2021)

Submitted To	**Submitted By**
Ms. Prachi Sharma	Aabhar Bhatt, Aarti Suthar, Arti, Aryan Jha
Associate professor Department of Computer Engineering	PCE18CS001, PCE18CS003, PCE18CS029, PCE18CS031
Poornima College of Engineering	III-A

DEPARTMENT OF COMPUTER ENGINEERING
POORNIMA COLLEGE OF ENGINEERING, JAIPUR
RAJASTHAN TECHNICAL UNIVERSITY, KOTA

July,2021

ACKNOWLEDGEMENT

I wish to express my sincere thanks and gratitude to **Dr. Sunil Kr. Yada, HOD Computer Engineering, PCE** for his affectionate and undaunted guidance as well as for his morale boosting encouragement with worthy suggestions and support.

I thank **Mr. Punit Kumwat, Assis. Professor, PIET** ,all our faculty members for their guidance in deciding the company and technology for training and helping in making this training report. Without their guidance, I would not have been able to make a proper decision.

I will be falling in my duty, if I don't offer my gratitude towards **CELEBAL TECHNOLOGY**, who always helped me time to time to understand my topic and the various related areas and guiding me throughout the training. Acknowledgement will not be over without mentioning a word of thanks to all industry persons for giving me such attention and time and helping me throughout the training.

Industrial Training would not have been possible without the efforts of **Dr Mahesh M. Bundele, Director, PCE.** I whole heartedly thank them for providing the students the opportunity to do our internship from a reputed company and move into a professional life for 60 days.

List of Figures

Table of Content

Abstract

Predicting the bed occupancy of an intensive care unit (ICU) is a daunting task. The uncertainty associated with the prognosis of critically ill patients and the random arrival of new patients can lead to capacity problems and the need for reactive measures. In this paper, we work towards a predictive model based on Random Forests which can assist physicians in estimating the bed occupancy. Our model is predicting the occupancy of icu by patients with different types of problems. (LOS)

We have taken the data from more than 6 hospitals for 210 days. We have taken the total number of ICU beds present in that hospital, available ICU beds, patient admitted and patient discharge. We have also shown the Occupancy in a single day of the hospital with the Occupancy percentage and average age of patient admitted in the hospital in a single day.

Right now the accuracy of our Project is 90% using a random forest algorithm.

1 : Introduction

In recent years, there is an increasing trend to automatically monitor, collect, process, and store clinical parameters of patients during their hospital visit. These automated systems have led to the creation of a vast array of heterogeneous and often incomplete data collections which are hypothesized to contain a wealth of hidden knowledge. However, these data compendia, often dubbed “big data goldmines” in popular media, have mainly been left untouched due to the inherent difficulty they present to (manually) extract information. In this report, we present our work to create a system which can assist physicians to predict the bed occupancy of an intensive care unit (ICU) based on automatically monitored clinical parameters.

Predicting the amount of free beds in an ICU is a difficult task as there is a substantial amount of uncertainty associated with the prognosis of critically ill patients. This prediction is further complicated by the fact that there is constant arrival of new patients that unexpectedly require intensive care. Nowadays, ICU physicians generally plan only a single day ahead based on clinical judgement whether or not a patient will leave the ICU. This can lead to situations in which capacity problems arise and planned surgeries have to be postponed. The development of an automated system which can assist physicians in these matters would clearly be beneficial to plan better and further ahead. This in turn could help reduce the financial cost associated with an intensive care unit. The latter impact should not be underestimated, as it was reported that the cost of care in 2005 for critically ill patients accounts for about 0.66% of the gross domestic product in India.

Our work is related to two streams of research. First, our research is related to the use of analytics in healthcare and, in particular, to the use of forecasting methods for planning healthcare capacity. Second, our research is related to the use of pooled forecasting and the combination of multiple methods to generate robust predictions.

2. Materials and Methods

2.1. Data Extraction

The data we have used is from 6 hospitals around the city for 210 days. This data is highly overrated for our project consisting of everyday input and output records of patient admission admitted and patient discharge. About final output is free and labelled and Occupancy in a hospital at a given date or a time.

Analytics have been shown to be relevant for supporting decisions in different components of healthcare systems . In recent years, we have seen an explosive growth of analytics applications in diverse facets of health care, including medical diagnosis, human resources, supply chain management, and the design of health care insurance (to name a few). Although the use of mathematical modeling in this area has brought a number of challenges, there are ample opportunities to generate essential and timely knowledge to support decision-making. In the context of the control of infectious diseases, the combination of big data and tractable analytical techniques has provided new tools to fight against pandemics. The global impact of COVID-19 has motivated numerous modeling efforts to provide guidelines for the control and management of the outbreaks. Certainly, there is a close relationship between the spread of the infection and the demand for medical resources. Therefore, mathematical models that describe the evolution of the pandemic can provide a first-order approximation of the demand for ICU beds. For this reason, we were especially concerned about the modeling effort needed to forecast the spread of the outbreak for the purpose of estimating the requirements of hospital resources..

3. Technology

- ☐ We used Different machine learning algorithm to perform our Task

3.1About Machine Learning:

Machine Learning is undeniably one of the most influential and powerful technologies in today's world. More importantly, we are far from seeing its full potential. There's no doubt it will continue to make headlines for the foreseeable future. This article is designed as an introduction to the Machine Learning concepts, covering all the fundamental ideas without being too high level.

Machine learning is a tool for turning information into knowledge. In the past 50 years, there has been an explosion of data. This mass of data is useless unless we analyse it and find the

patterns hidden within. Machine learning techniques are used to automatically find the valuable underlying patterns within complex data that we would otherwise struggle to discover. The hidden patterns and knowledge about a problem can be used to predict future events and perform all kinds of complex decision making.

Most of us are unaware that we already interact with Machine Learning every single day. Every time we Google something, listen to a song or even take a photo, Machine Learning is becoming part of the engine behind it, constantly learning and improving from every interaction. It's also behind world-changing advances like detecting cancer, creating new drugs and self-driving cars.

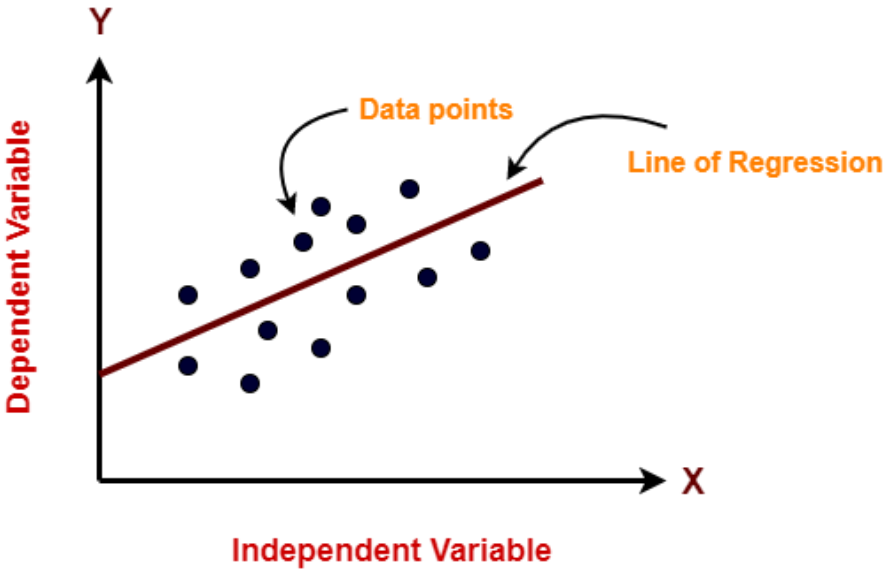
3.2 Algorithms Used

We used three algorithms to train our model. The first algorithm we used was Logistic regression and 2nd algorithm we used was k-nearest neighbour and 3rd algorithm to train our model was random forest.

3.2.1. Linear regression

Linear Regression is an ML algorithm used for supervised learning. Linear regression performs the task to predict a dependent variable(target) based on the given independent variable(s). So, this regression technique finds out a linear relationship between a dependent variable and the other given independent variables. Hence, the name of this algorithm is Linear Regression.

FIGURE 1: Linear Regression



In the figure above, on X-axis is the independent variable and on Y-axis is the output. The regression line is the best fit line for a model. And our main objective in this algorithm is to find this best fit line.

Pros:

- Linear Regression is simple to implement.
- Less complexity compared to other algorithms.
- Linear Regression may lead to over-fitting but it can be avoided using some dimensionality reduction techniques, regularization techniques, and cross-validation.

Cons:

- Outliers affect this algorithm badly.
- It over-simplifies real-world problems by assuming a linear relationship among the variables, hence not recommended for practical use-cases.

Implementation

3.2.2 Logistic 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:

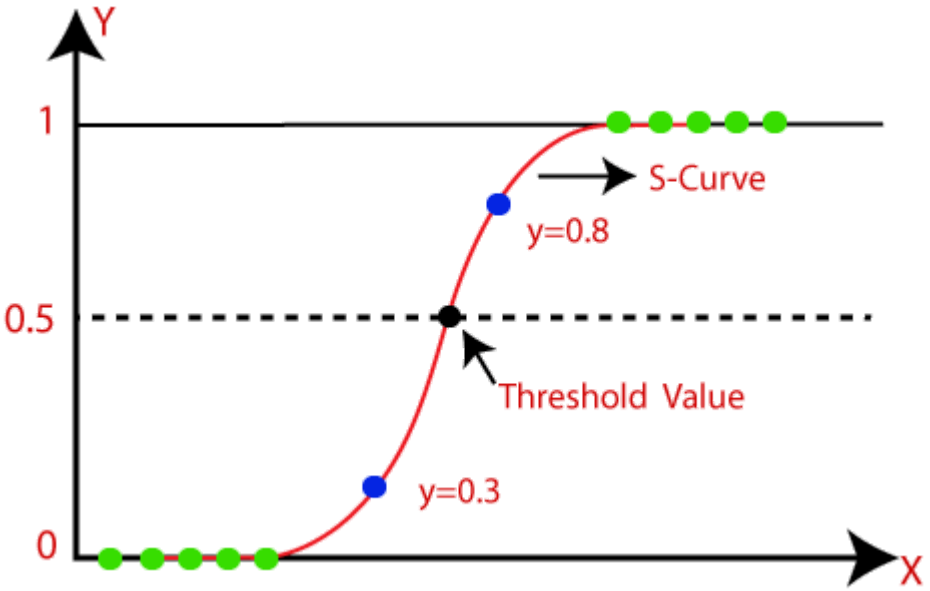


FIG 2: LOGISTIC REGRESSION

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):

$\frac{y}{1-y}$; 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.

Implementation:

```
from sklearn.linear_model import LogisticRegression

from sklearn.metrics import f1_score, auc

from sklearn.metrics import roc_auc_score

logreg = LogisticRegression()

logreg.fit(train_x, train_y)

pred_date = logreg.predict_proba(valid_x)

pred_occupancy = logreg.predict_proba(valid_y)

logreg.score(valid_x,valid_y)
```

3.2.2 KNN Algorithm

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry. The following two properties would define KNN well −

- Lazy learning algorithm − KNN is a lazy learning algorithm because it does not have a specialized training phase and uses all the data for training while classification.
- Non-parametric learning algorithm − KNN is also a non-parametric learning algorithm because it doesn't assume anything about the underlying data.

Working of KNN Algorithm

K-nearest neighbors (KNN) algorithm uses 'feature similarity' to predict the values of new datapoints which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of following steps −

Step 1 − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

Step 2 − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

Step 3 − For each point in the test data do the following −

- 3.1 − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.
- 3.2 − Now, based on the distance value, sort them in ascending order.
- 3.3 − Next, it will choose the top K rows from the sorted array.

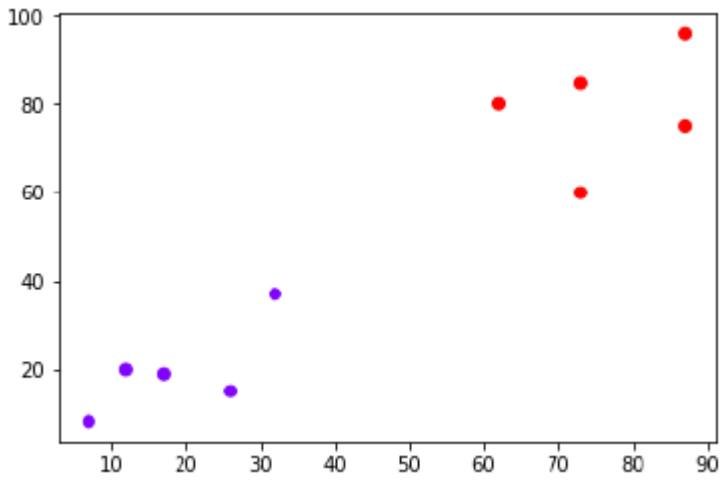
- 3.4 − Now, it will assign a class to the test point based on most frequent class of these rows.

Step 4 − End

Example

The following is an example to understand the concept of K and working of KNN algorithm –

Suppose we have a dataset which can be plotted as follows −



Now, we need to classify new data point with black dot (at point 60,60) into blue or red class. We are assuming K = 3 i.e. it would find three nearest data points. It is shown in the next diagram −

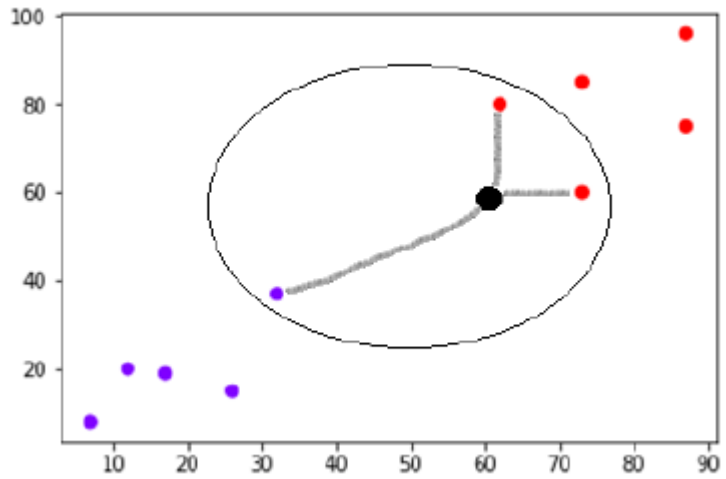
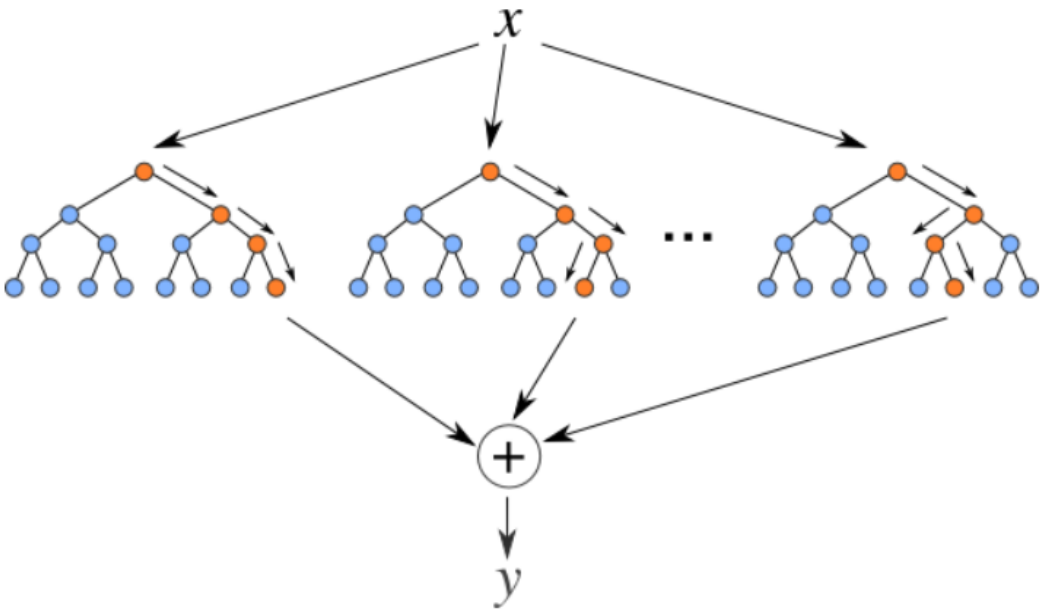


FIG 3: KNN

We can see in the above diagram the three nearest neighbors of the data point with black dot. Among those three, two of them lies in Red class hence the black dot will also be assigned in red class. Implementation in Python

3.2.3 Random Forest Algorithm

Random Forests are an ensemble(combination) of decision trees. It is a Supervised Learning algorithm used for classification and regression. The input data is passed through multiple decision trees. It executes by constructing a different number of decision trees at training time and outputting the class that is the mode of the classes (for classification) or mean prediction (for regression) of the individual trees.



Source: https://levelup.gitconnected.com

Pros:

- Good at learning complex and non-linear relationships
- Very easy to interpret and understand

Cons:

- They are prone to overfitting
- Using larger random forest ensembles to achieve higher performance slows down their speed and then they also need more memory.

Implementation

```
from sklearn.ensemble import RandomForestClassifier

rfc=RandomForestClassifier()
```

```
rfc.fit(x_train,y_train)

pred_2=rfc.predict(x_test)

score_2=accuracy_score(y_test,pred_2)

score_2
```

4 Model Snapshot and visuals

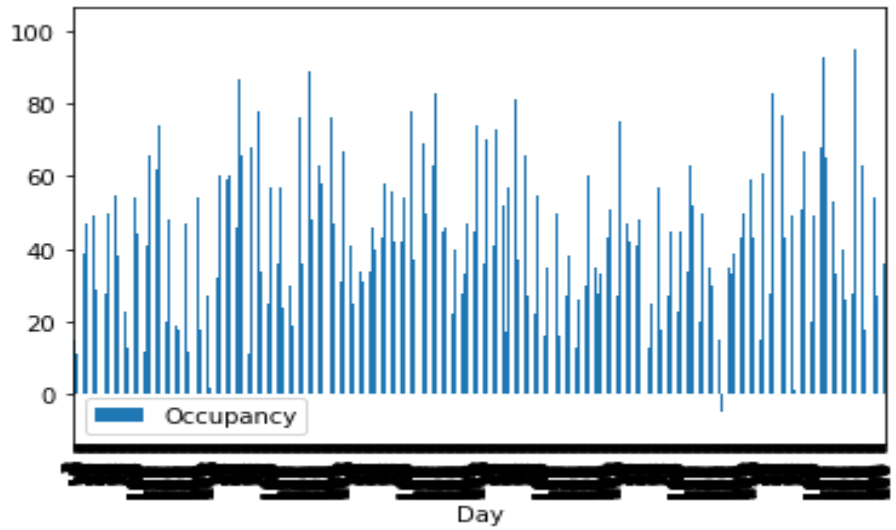
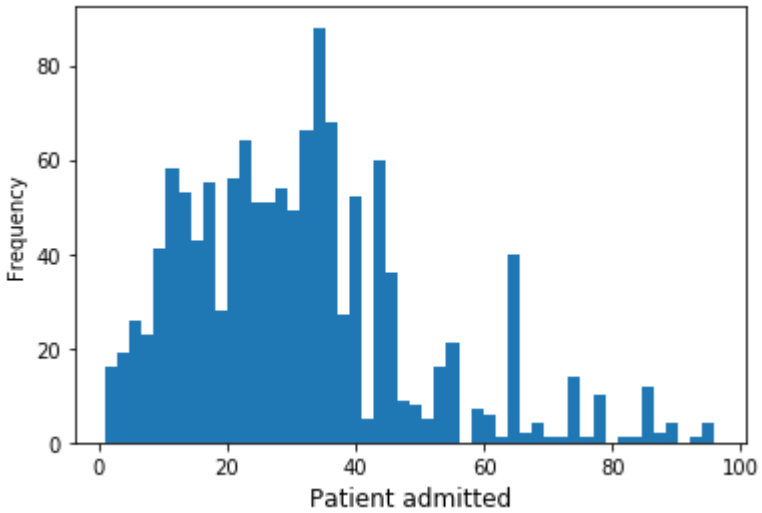
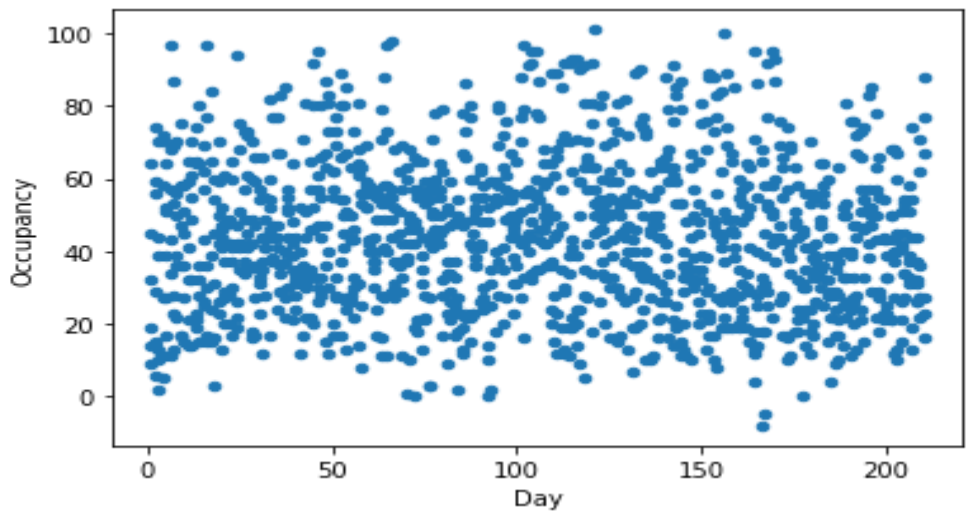


Fig: Subplot of Accuracy





jupyter ICU BED PREDICTION (Final) (1) Last Checkpoint: a minute ago (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help

Not Trusted | Python 3

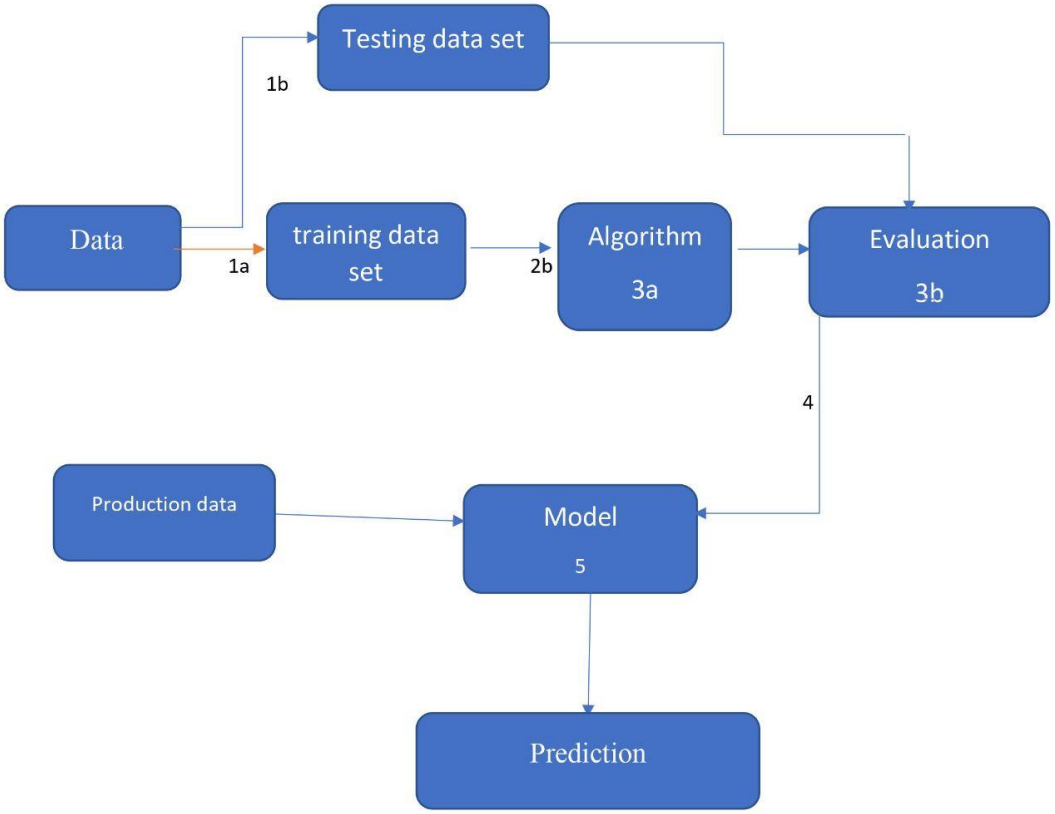
Run Code

```
from sklearn.ensemble import RandomForestRegressor
# create regressor object
regressor = RandomForestRegressor(n_estimators = 100, random_state = 0)

# fit the regressor with x and y data
regressor.fit(x, y)
    #predicting test values
y_pred1=regressor.predict(x_test)
    #predicting hospital values
y_pred2=regressor.predict([a[i][2:]])

# Output
from sklearn.metrics import r2_score
print("For Hospital", p)
Total_beds = hos['Total ICU Bed'].max()
print("Total Beds:", Total_beds)
print("accuracy: ",int((r2_score(y_test,y_pred1))*100), "%")
print("Occupancy:",int(y_pred2))
print("% Occupancy:", int((int(y_pred2)/Total_beds)*100), "%")
print("Free Beds Available:", Total_beds-int(y_pred2))
print()
```

```
For Hospital 3
Total Beds: 90
accuracy:  97 %
Occupancy: 20
% Occupancy: 22 %
Free Beds Available: 70
```

In [ ]:

Data Flow Diagram for Modelling ICU Occupancy



5. Conclusion:

The database offered surprisingly good depth and detail related to medical admissions which enabled me to create a hospital length-of-stay prediction model that considered a lot of interesting input features. The most surprising aspect of this work was how the patient's diagnosis played a more important role than age when predicting the length-of-stay. By far, the most challenging aspect of this project was the feature engineering of the data diagnosis into a more practical and interpretable form of supercategories. However, therein also lies the most obvious area for future improvement. Given that the diagnoses have such strong feature importance, it would be worth evaluating whether additional subdividing of the primary ICD-9 categories would yield a better prediction model. My theory is that the prediction model would become more accurate (lower RMSE) with this optimization, so long as there were enough admission records in the dataset to support reasonable diagnoses model training.

GITHUB LINK- **https://github.com/artiyadav039/Modelling-ICU-Occupancy**

6: Reference:

https://www.essentialguru.org/coursera-applied-data-science-python-review/

https://www.coursera.org/

https://towardsdatascience.com

https://www.kaggle.com/

https://github.com/