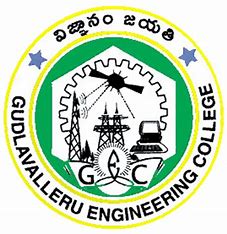
****

**SESHADRI RAO**

**GUDLAVALLERU ENGINEERING COLLEGE**

**Department of Mechanical Engineering**

**Internship Project**

**On**

**Hospital Readmission prediction using IBM Watson**

**K. Venkata Ramana (20485A0307)**

**D. Jithendhar (19481A0326)**

**K. Naveen (20485A0313)**

**HOSPITAL READMISSION PREDICTION USING IBM WATSON**

**INTRODUCTION**

* 1. **OVERVIEW**

So many hospitals there in this world.. let consider one hospital. Patients with different problems and illness are join in the hospital. if a hospital has multiple Readmissions, it means that the hospital needs to work on the quality of services it is providing with respect to the health and wellness of its patients. Being able to predict whether a person will be able to predict whether a person will be readmitted to the hospital within 30 days or not, will be of great help to the hospital in developing an idea of the incoming number of repeated patients which in turn helps to provide better services for patients with increased risk of disease.

One patient population that is at increased risk of hospitalization and readmission is diabetes. Diabetes is a medical condition that affects approximately 1 in 10 patients In the united states. So in this project, we will be focussing on hospital readmission prediction for patients who are having diabetes.

* 1. **PURPOSE**

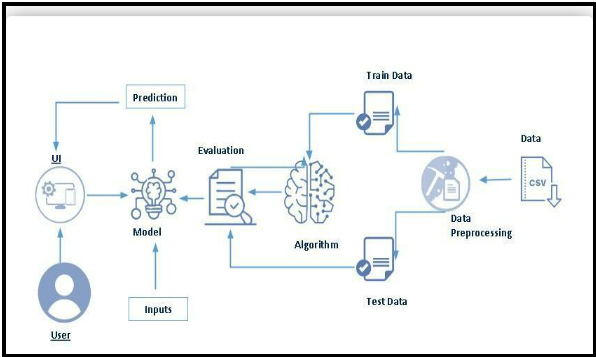
This study helps us to predict whether a person who is suffering from diabetes and consulting a specific hospital will be readmitted or not, based on multiple factors.

1. **LITERATURE SURVEY**
   1. **Existing Approach and method to solve the problem**

The main purpose of this project is to predict whether a person who is suffering from diabetes and consulting a specific hospital will be readmitted or not, based on multiple factors.

We will be using classification algorithms such as Logistic Regression, KNN, Decision tree, Random Forest, AdaBoost, and GradientBoost. We will train and test the data with these algorithms. From this, the best model is selected and saved in pkl format. We will also be deploying our model locally using Flask.

1. **Theoritical Analysis**
   1. **Block Diagram**



* 1. **Software Requirements of the project**

To complete this project, you will require the following software, concepts, and packages

    Anaconda navigator:

o    Refer to the link below to download the anaconda navigator

**Python packages:**

o “numpy”

o “pandas”

o   “scikit-learn”

o    ”matplotlib”

o   ”scipy”

o    ”pickle-mixin”

o    ”seaborn”

* “ Flask”

1. **Experimental Investigation**

The Aim of this study is identify possible predictions of avoidable hospital admissions of patients with diabetes that are related to improper management of the disease. Variable under consideration were patient demographics (age, gender, ethnicity), lifestyle components (marriage status, religious affiliation or not and smoking status), biomarkers (glucose level, blood pressure) and disease management aspects (physician specially or not and participation in a diabetes education program) to predicting hospitalization readmissions for patients with diabetes.

1. **Flowchart**

Data Collection

Data Pre-Processing

Data Analysis And Visualization

Model Building

Application Building

Train The Model On The IBM

**Data Collection**

There are many popular open sources for collecting the data. Eg: kaggle.com, UCI repository, etc.

In this project, we have used diabetic\_data.csv. This data is downloaded from the Kaggle.com

Data Pre-Processing

Handling Null Values and Removing Unnecessary Columns

Though the dataset seems to be completely free of null values, it is not so.  We observe from the head and tail of data that a number of fields are filled with ‘?’.  These are nothing but null values.  So in order to get the null values count of each column, replace all ‘?’ in data with np.nan and then find the sum of null values.

**Data Analysis And Visualization**

**Uni Variate Analysis**

In simple words, univariate analysis is understanding the data with a single feature.

Let us first plot the values of our target column – readmitted

### Bi Variate Analysis

We use bivariate analysis to find the relation between two features.  Here we are visualising the relationship of various features with respect to readmitted, which is our target variable.

**Descriptive Analysis**

Descriptive analysis is to study the basic statistical features of data. We can achieve it by using the .describe() function. With this describe function we can understand the unique, top, and frequent values of categorical features. Also, we can find mean, std, min, max and percentile values of numerical features.

### Model Building

### Handling Categorical Values

As we can see our dataset has categorical data.  Before training our model, we must convert the categorical data into a numeric form.

There are multiple encoding techniques to convert the categorical columns into numerical columns. For this project we will be encoding some features manually and some others using OrdinalEncoder()

Firstly, let us modify the values in the columns admission\_type\_id, discharge\_disposition\_id and admission\_source\_id with the help of IDs\_mapping.csv file.

**Splitting Data Into Train And Test Sets**

For splitting the data into train and test sets, we are using the train\_test\_split() function from sklearn. As parameters, we are passing X, y,stratify, test\_size, random\_state.

**Sampling Data**

We will be using SMOTE algorithm for oversampling our minority class.  To know in detail about this algorithm.

### Comparing Performance Of Various Models

We will be considering multiple models to train our data and choose the one that performs the best.  So, we need to import the necessary libraries and create a dictionary of our models.

### Feature Selection

We have trained our model with 29 features.  But all these features may not be important for prediction.  Hence we will select the features that contribute significantly to the model performance.

### Evaluating Final Model Performance

We will compare the confusion matrix, ROC curve and classification report for both models.

In order to obtain these, we will be using the confusion\_matrix(),roc\_curve() and classification\_report() functions from sklearn.metric

**Application Building**

In this section, we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has to enter the values for predictions. The entered values are given to the saved model and prediction is showcased on the UI.  
   
This section has the following tasks

* Building HTML Pages
* Building server-side script

### Building Html Pages

For this project create three HTML files namely

* home.html
* index.html
* output.html

and save them in the templates folder.

Let’s see how our home.html page looks like:

The objective tab describes the main objective of executing this project

**Run The Application**

### ?         Open anaconda prompt from the start menu

?         Navigate to the folder where your python script is.

?         Now type “python app.py” command

?         Navigate to the localhost where you can view your web page.

?         Click on the proceed button, enter the inputs, click on the predict button, and see the result/prediction on the web.

### Train The Model On IBM

* In this milestone, you will learn how to build a Deep Learning Model and deploy it on the IBM Cloud.

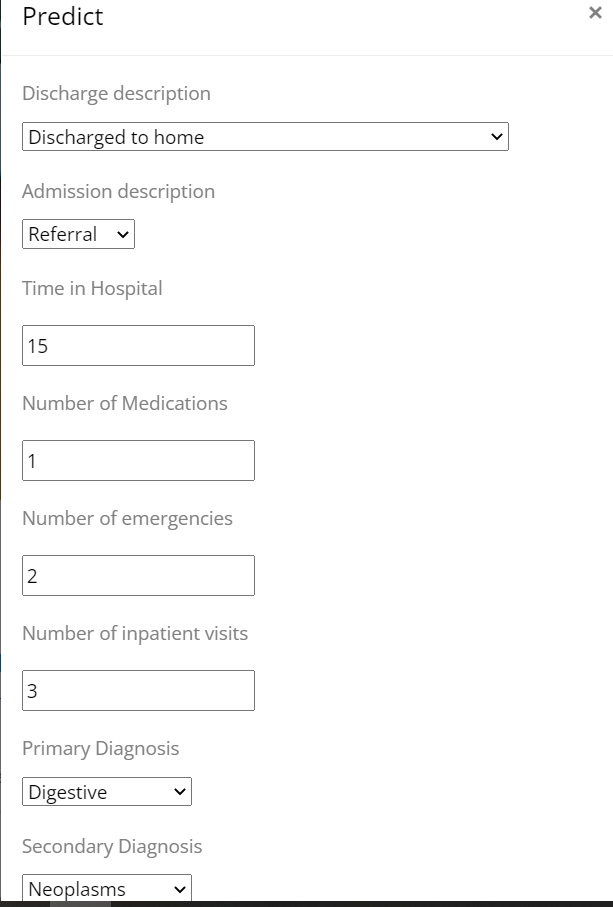
Registration For IBM Cloud

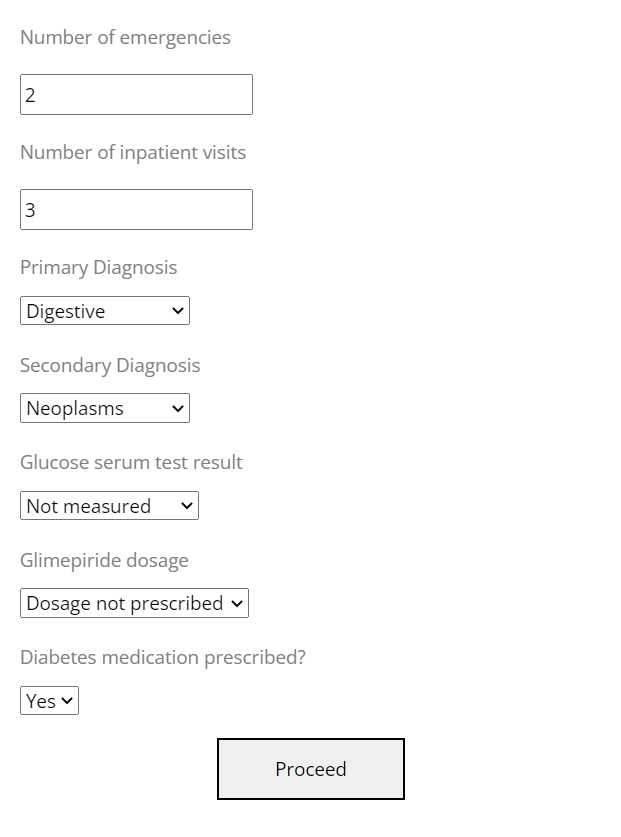
Register for the IBM -Cloud to further Operations.

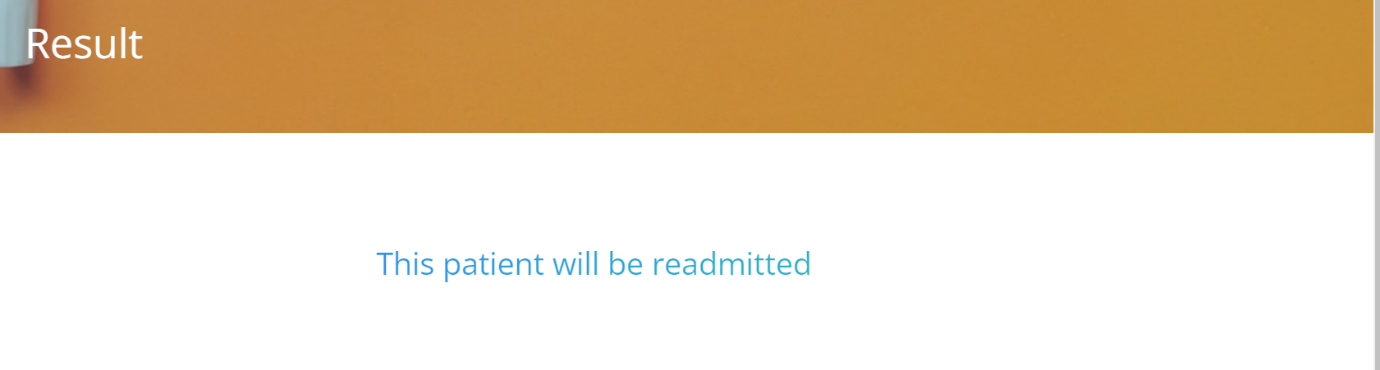
### Train The ML Model On IBM

### Integrate Flask With Scoring End Point

**Results are Obtained Following**







THANK YOU!