Big Data Analytics for Competitive Advantage

Deliverable - 2

Diabetic Patient Readmission Prediction

ITCS-6100

Submitted by:

Ajith Kumar Gannamaneni
801305442
agannama@uncc.edu

Vivek Sai Gujja
801305448
vgujja@uncc.edu

Abhinav Botla
801290454
abotla@uncc.edu

Akhil Rayapati
801298428
arayapat@uncc.edu

Introduction:

Hospital readmission is a significant factor in overall medical costs and is a developing metric for care quality. Diabetes, similar to other chronic medical conditions, is associated with increased risk of hospital readmission. Hospitals are striving harder to get the patient details in order to equip themselves better and serve the patients effectively. This can be considered as a business problem in this case. In this model, we will predict whether the high-risk diabetic-patients is likely to get readmitted to hospital after previous encounter within thirty days or after thirty days or totally not.

Domain Knowledge:

Original data source is stored at UCI Repository. This data has been prepared to analyse factors related to readmission as well as other outcomes pertaining to patients with diabetes.

GitHub Repository - https://github.com/raoajith2210/Diabetic-Patient-Readmission-Prediction.git

Exploratory Data Analysis:

The dataset used in this project depicts whether a potential patient will be readmitted into the hospital due to their diabetic condition. Our dataset contains the below columns.

Column Name	Data type	Data type
race	STRING	Caucasian, Asian, African American or Hispanic
time_in_hospital	INT	Number of days between admission and discharge a.k.a. length of stay
number_outpatient	INT	Number of outpatient visits of the patient in a given year before the encounter
number_inpatient	INT	Number of inpatient visits of the patient in a given year before the encounter
number_emergency	INT	Number of emergency visits of the patient in a given year before the encounter
number_diagnoses	INT	Number of diagnoses entered in to the system
num_procedures	INT	Number of procedures (other than lab tests) performed during the encounter

num_medications	INT	Number of distinct generic medicines administrated during the encounter
num_lab_procedures	INT	Number of lab tests performed during the encounter
max_glu_serum	STRING	Indicates the range of result or if the test was not taken. Values: ">200", ">300", "normal" and "none" - if not measured
gender	STRING	Values: "Male", "Female" and "Unknown/Invalid"
diabetes_med	INT	Indicates if any diabetes medication was prescribed.
change	STRING	Indicates if there was a change in diabetic medications (ether dosage or generic name). Values: "change" or "no change"
age	INT	Age of patient at the time of encounter
a1c_result	STRING	Indicates range of the result of if it was not taken. Values: ">8", ">7", "normal" and "none"
readmitted	STRING	Days to inpatient readmission. Values: "<30" if patient readmitted less than 30 days, ">30" if patient readmitted after 30 days of encounter, "no" for no record of readmission

Data Preparation:

The input file features 69,570 rows along with 17 columns. It contains all the statistical information like number_of_medications and number_of_procedures.

The data contains such attributes as patient number, race, gender, age, admission type, time in hospital, medical specialty of admitting physician, number of lab test performed, HbA1c test result, diagnosis, number of medications, diabetic medications, number of outpatient, inpatient, and emergency visits in the year before the hospitalization, etc.

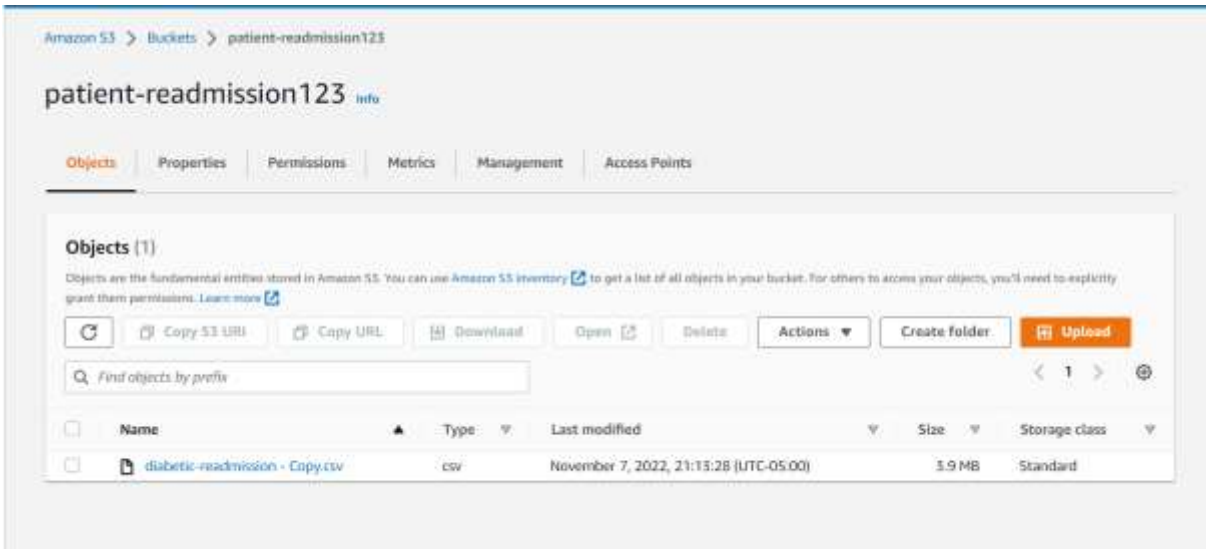
Data Limitations:

- The input dataset only contains the data from 1999–2008
- The dataset is only collected from 130 US hospitals and cannot be generalized with a global scenario.

We have loaded the dataset into Amazon Simple Storage Service (S3) primarily by creating a new bucket by name patient-readmission123.

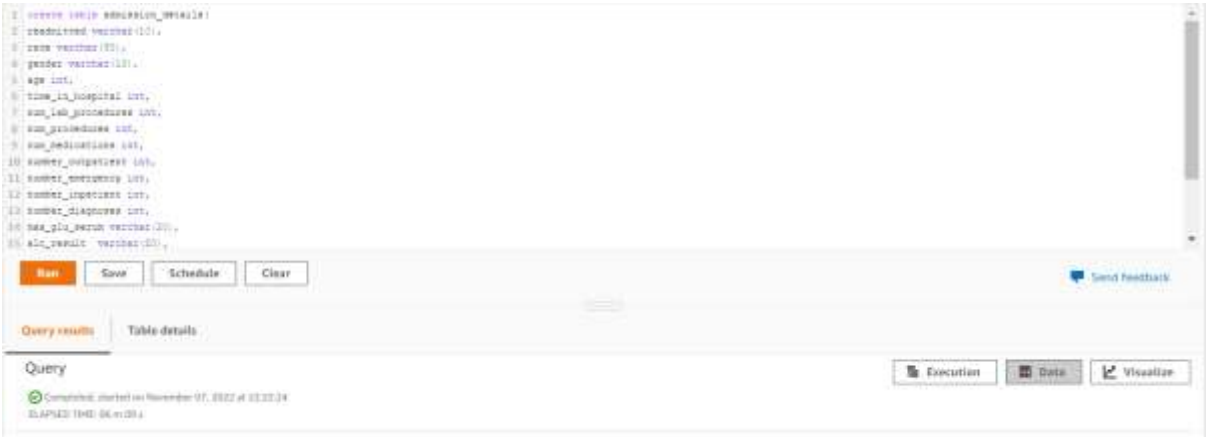
Bucket Name – **patient-readmission123**

Input Dataset – **diabetic-readmission – Copy.csv**



Once the data is loaded to S3, we will set up a new redshift cluster (dc2.large with 2 nodes for load handling). We will then set up a new redshift role in order to access S3 dataset from redshift.

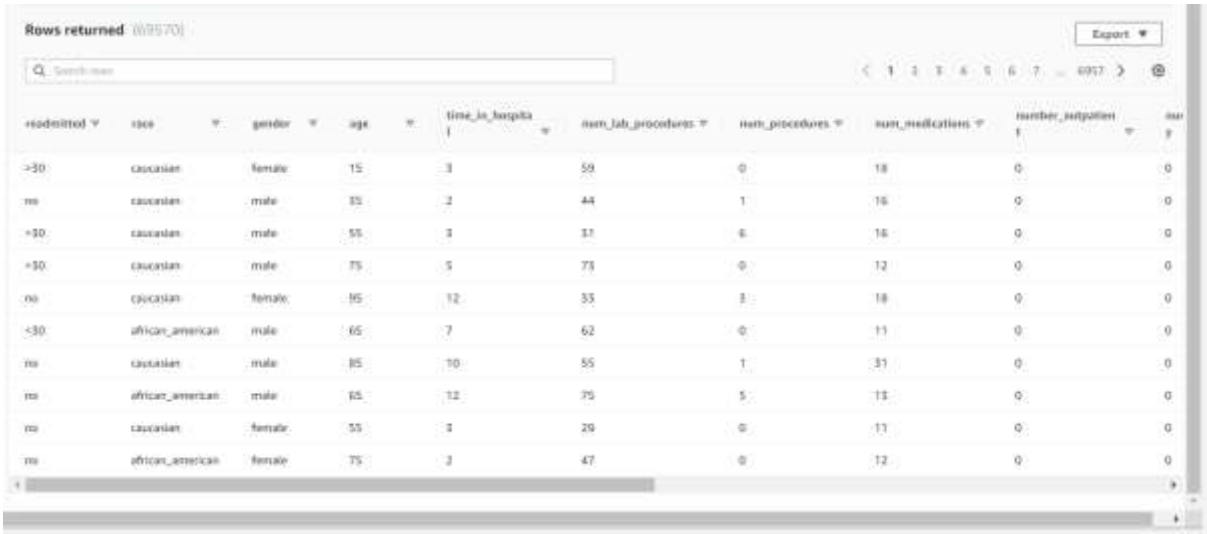
Created a new table by name **admission_details** in redshift as shown in the below snip.



Datatypes used in the table are integer and varchar to store numeric and character data.

Once the table is created in redshift, run a copy command in order to ingest data from S3 to redshift.

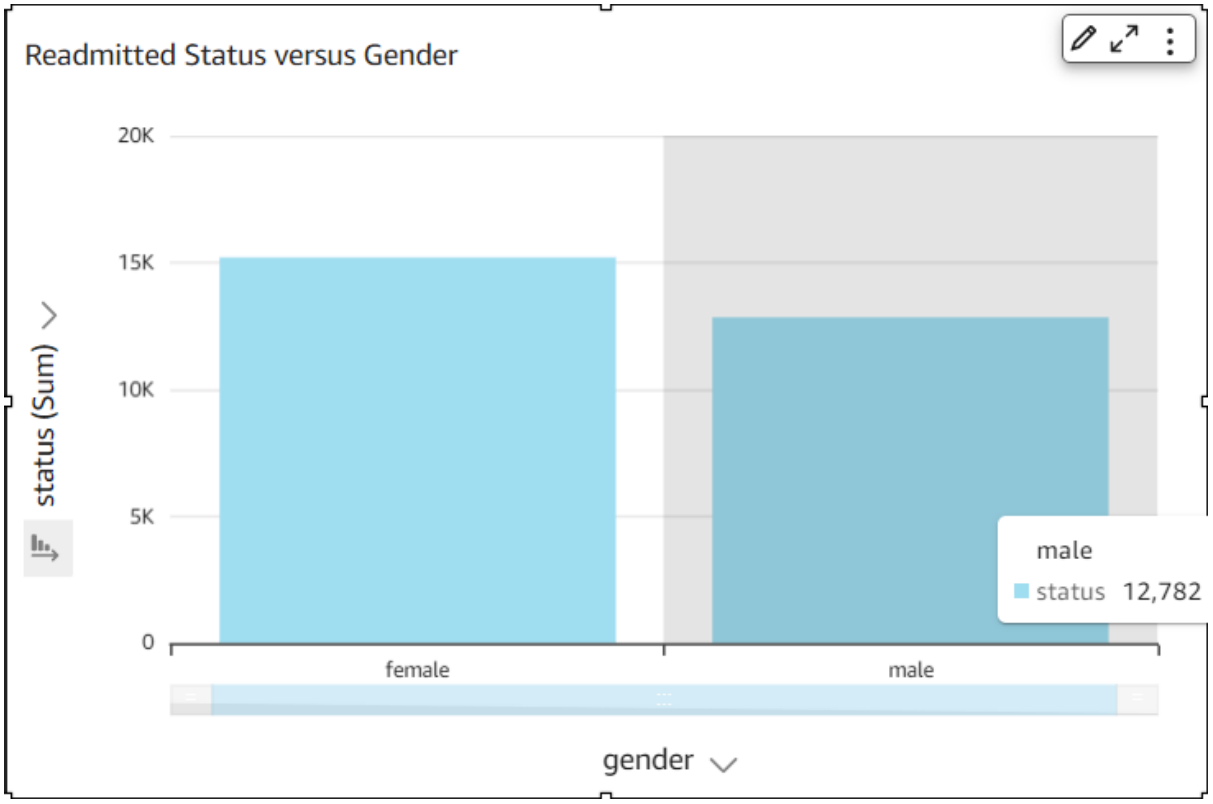
Confirm whether all the rows are loaded from S3 to redshift.



Count of rows should be 69,570 (number of rows in initial dataset)

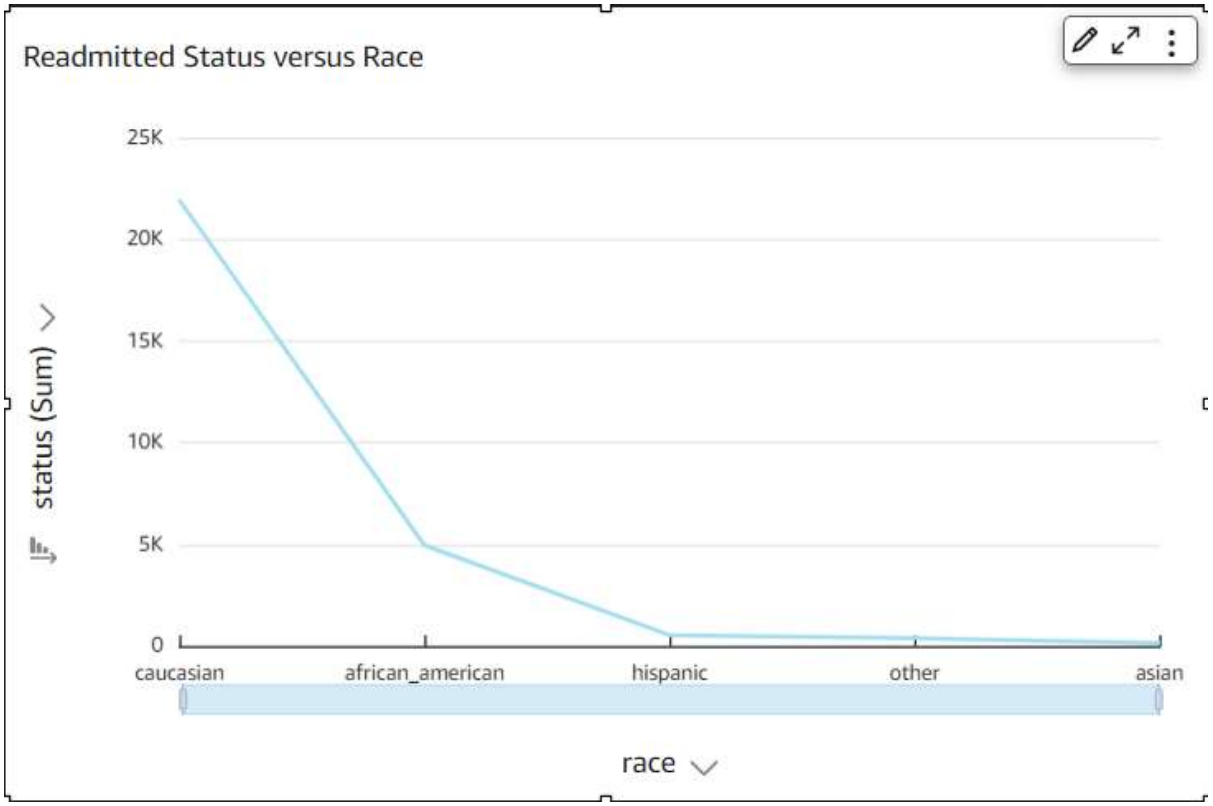
Integrate Amazon QuickSight with S3 in order to share the input dataset.

Let us try to co-relate the gender field with the project predictive(status). In this case, let us do a quick comparison between gender and status field.



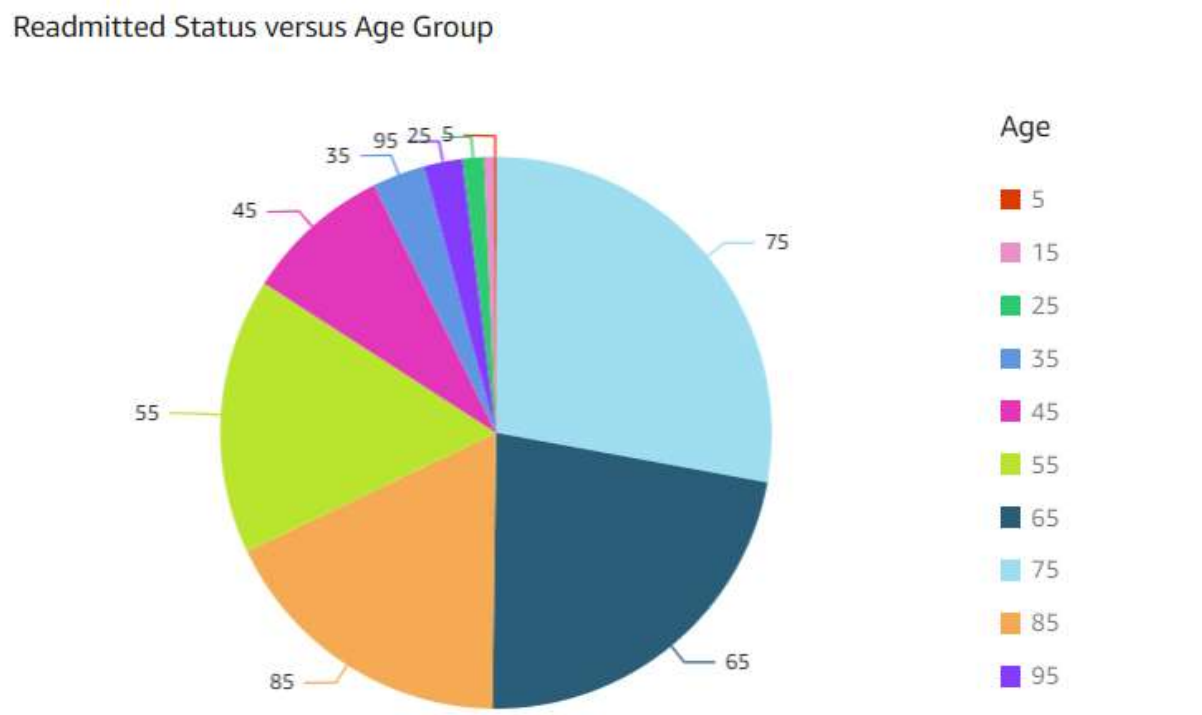
In the above bar graph, we can infer that the number of female patients getting re-admitted is slightly higher than male patients.

There are 5 different races available in the dataset. Let us compare the race attribute with the readmitted status.

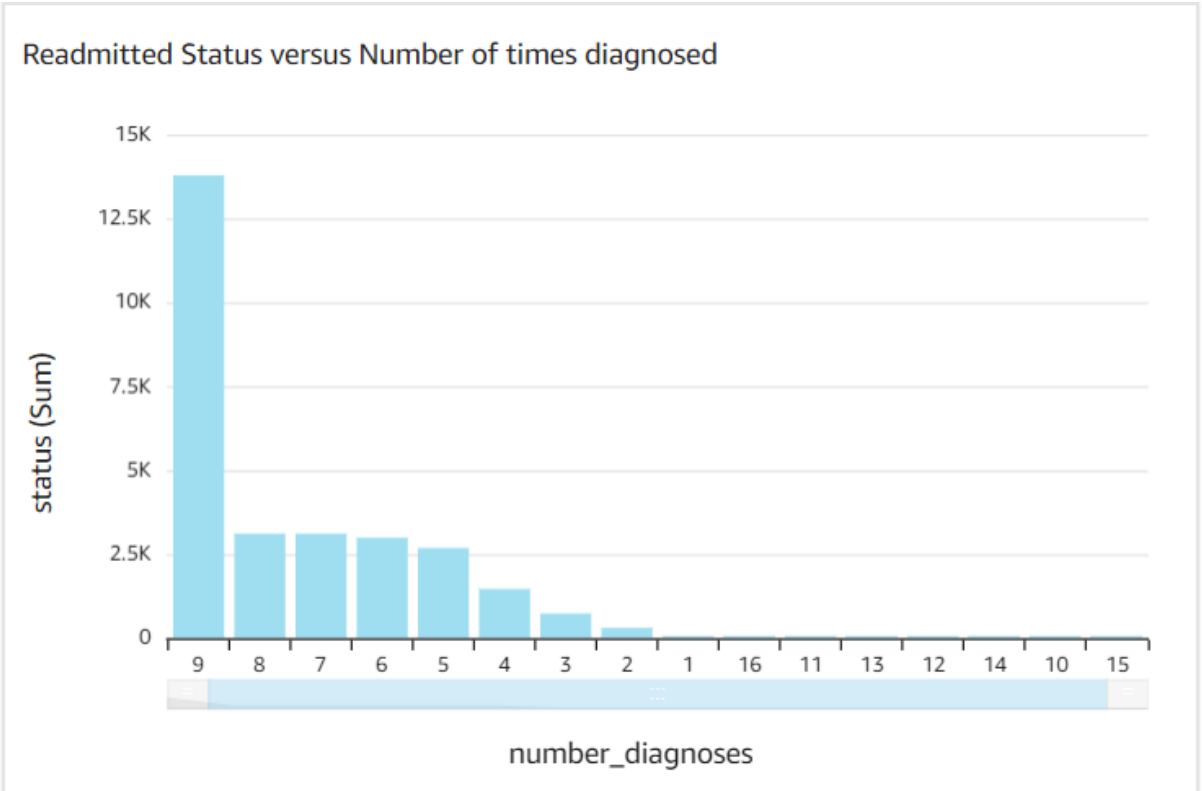


In the above line chart, most of the Caucasians are getting readmitted due to some factors. Asians on the other side have a minimal impact on readmission and are predicted to never visit a hospital after admission.

We can also compare age attribute with the readmission status field. With this analysis, we can infer the age group that are most likely to be impacted even after hospital visit.



With the above pie chart, we can infer that the higher age groups are most likely to be impacted i.e., people with age>55 are mostly readmitted compared to age<35.



Combining all the above attributes, we can quantify that the age and race plays a crucial role in predicting the project outcome. A **Caucasian female with age>55** is the most likely person in the entire dataset to get re-admitted and an **Asian male with age<35** is the least likely person to get re-admitted. We can also consider all the other attributes to arrive at a solid foundation on the factors effecting health.

In this study, we have selected and compared three ML models. The random forest (RF) algorithm is a basic classification algorithm built by a decision tree (DT). Every DT is considered as a weak classifier, and the collection of responses produces a strong classifier. Each DT is relatively independent and the category of input data is judged by learning a series of binary problems, which is advanced in its easy-to-understand design, high accuracy, and good robustness.