AI-POWERED PREDICTIVE MODELING OF DIABETIC PATIENT READMISSIONS: AN EXPLAINABLE MACHINE LEARNING APPROACH

S2-24_DSECCZG628T
DISSERTATION

by

Harshavardhini Saravana Kumaran
2023C104040

Project Work carried out at

COGNIZANT, Coimbatore

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
Pilani (Rajasthan) India

Sept 2025

S2-24_DSECCZG628T DISSERTATION

AI-POWERED PREDICTIVE MODELING OF DIABETIC PATIENT READMISSIONS: AN EXPLAINABLE MACHINE LEARNING APPROACH

Submitted in partial fulfilment of the requirements of

M. Tech. Data Science and Engineering Program

by

Harshavardhini Saravana Kumaran
ID No. 2023C104040

Under the supervision of

Karthic G M, Testing Lead

Project Work carried out at

COGNIZANT, Coimbatore

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE
PILANI (RAJASTHAN)

September, 2025

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

CERTIFICATE

This is to certify that the Project Work entitled **AI-Powered Predictive Modeling Of Diabetic Patient Readmissions: An Explainable Machine Learning** and submitted by **Harshavardhini Saravana Kumaran** ID No. **2023C104040**. in partial fulfillment of the requirements of **S2-24_DSECCZG628T** Dissertation, embodies the work done by him/her under my supervision.

G M Karthic

Signature of the Supervisor
Name: G M Karthic
Designation: Testing Lead at Cognizant

Date:9th September 2025

BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, PILANI
S2-24_DSECCZG628T DISSERTATION
SECOND SEMESTER 2021- 2022

Project Work Title : AI-Powered Predictive Modeling of Diabetic Patient Readmissions: An Explainable Machine Learning

Name of Supervisor : Karthic G M

Name of Student : Harshavardhini Saravana Kumaran

BITS ID No. of Student :2023C104040

Abstract *(around 250 words)***:**

Hospital readmissions among diabetic patients are a significant concern due to the chronic and complex nature of diabetes management. Complications such as hypoglycemia, diabetic ketoacidosis, cardiovascular events, and infections often lead to unplanned readmissions, especially within 30 days of discharge. These events burden an already overstretched healthcare system and increase the cost of care for patients.

Despite the availability of electronic health records (EHRs), clinicians often lack effective tools to accurately predict which diabetic patients are at the highest risk of readmission. Traditional methods for identifying high-risk patients rely heavily on clinician experience and basic scoring systems (like LACE index), which lack precision and adaptability. Existing predictive models are either too generalized or lack interpretability, making them less actionable.

This study aims to compare the performance of various models such as Logistic Regression, Random Forest, XGBoost, and Artificial Neural Networks, identify the best model evaluate their interpretability using XAI techniques such as SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) suited for accurate and transparent readmission prediction. The proposed methodology also aims to make use of appropriate encryption techniques to deal with PII data and build the model while staying in line with ethical principles of data science and not compromising client sensitive data. The expected outcome is a clinically meaningful AI model that not only predicts readmissions with high accuracy but also provides insights into the key risk factors contributing to readmissions, thereby aiding physicians in making proactive decisions. The project has potential implications for reducing unnecessary hospitalizations, improving chronic disease management, and supporting data-driven healthcare.

Key Words (in Alphabetical order):

Artificial Intelligence, Deep Learning, Electronic Health Records, Explainable AI, Hospital Readmission, Interpretable Machine Learning, Medical Informatics, Predictive Analytics, SHAP, XAI

G M Karthic

Harshavardhini S

(Signature of Supervisor) (Signature of Student)

ACKNOWLEDGEMENT

The completion of this dissertation would not have been possible without the invaluable support, encouragement, and guidance of several individuals, to whom I owe my deepest gratitude.

I express my sincere thanks to my Internal Guide, **Karthic G M**, whose insightful advice, constant motivation, and patient guidance have been a driving force throughout the course of my project. Their suggestions and encouragement helped me overcome challenges and successfully complete this work within the stipulated time.

I also take this opportunity to extend my heartfelt appreciation to **Dr. Jayalakshmi N.** for providing direction and timely feedback, which greatly enriched the quality of my research.

My sincere thanks are also due to my classmates and friends for their cooperation and moral support, which made this journey more meaningful. I am equally grateful to the **Department of Computer Science and Engineering**, including both teaching and non-teaching staff, for their assistance and encouragement at every stage.

Finally, I would like to acknowledge my family for their unwavering support, understanding, and encouragement, without which this dissertation would not have been possible.

Harshavardhini Saravana Kumaran

Table of Contents

List of Images

Acronyms List:

Acronym	Full Form
AI	Artificial Intelligence
ANN	Artificial Neural Network
AUC	Area Under the Curve
DL	Deep Learning
EDA	Exploratory Data Analysis
HER	Electronic Health Records
FN	False Negative
FP	False Positive
ICU	Intensive Care Unit
LACE Index	Length of stay, Acuity of admission, Comorbidity of patient, Emergency department use
LIME	Local Interpretable Model-agnostic Explanations
LR	Logistic Regression
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
PII	Personally Identifiable Information
RF	Random Forest
ROC	Receiver Operating Characteristic
SHAP	SHapley Additive exPlanations
SMOTE	Synthetic Minority Over-sampling Technique
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
XAI	Explainable Artificial Intelligence
XGBoost	Extreme Gradient Boosting
GBDT	Gradient Boosting Decision Tree

1. INTRODUCTION

The proposed work aims to:

Develop and compare various predictive models to estimate hospital readmission risk, with respect to Isolated clinical Diabetic patient data.
Integrate Explainable AI tools to estimate prediction to clinicians.
Evaluate the predictive models built using real world or benchmark healthcare datasets.

Background of Previous Work Done in the Area:

The existing work in this area focusses on usage of traditional machine learning models like Logistic regression, Random Forest and SVM.
Some existing methodologies also make use of Deep learning techniques such as LSTM for time series analysis of the EHR data to facilitate Hospital Readmission Risk Prediction.
However, it is observed that past and current works in area have a very limited or complete lack of interpretability framework integration in clinical deployment.

Methodology (ways and means to be adopted in achieving the objective/ aim):

The proposed aim is to Integrate cutting-edge XAI methods like SHAP, LIME, and integrated gradients with Predictive models and demonstrate that this methodology is a step forward in the area of clinical risk prediction for Readmissions with respect to diabetic patients.

Benefits expected from the work:

Improved trust and transparency in clinical AI systems
Better patient outcomes through early interventions
Reduction in healthcare costs via prevention of unnecessary readmissions
Enhanced collaboration between data scientists and healthcare providers

2. Detailed Plan of Work (for 16 weeks)

The plan of work should have tangible weekly or fortnightly milestones and deliverables, which can be measured to assess the adherence to the plan and therefore the rate of progress in the work. The plan of work can be specified in the table given below:

Serial Number of Task	Tasks or Subtasks to be Done	Planned Duration in Weeks	Specific Deliverable in Terms of the Project
1	Conduct a literature review on hospital readmissions, diabetes, and machine learning models	1 week	Literature review chapter, list of references, gaps in existing research
2	Explore UCI Diabetes dataset and other related datasets	1 week	Dataset description, summary statistics, and dataset limitations
3	Preprocess data (handle missing values, outliers, encode categorical variables, scale features, split data)	2 weeks	Cleaned dataset, preprocessing code, feature engineering report
4	Train baseline models (Logistic Regression, Decision Tree) and evaluate with metrics (accuracy, precision, recall, AUC)	2 weeks	Baseline model results, evaluation report with metrics
5	Apply Explainable AI (XAI) methods like SHAP or LIME to interpret model predictions	2 weeks	XAI plots, SHAP value explanations, interpretation report
6	Implement advanced models (Random Forest, XGBoost, Neural Networks), perform cross-validation	1 week	Comparison of models, final model selection with evaluation charts
7	Optimize the selected model (hyperparameter tuning, model performance analysis)	1 week	Optimized model with improved performance metrics
8	Apply Explainable AI (XAI) methods like SHAP or LIME to interpret model predictions and build /deploy a dashboard using Streamlit	2 weeks	XAI plots, SHAP value explanations, interpretation report, deployment of dashboard
9	Draft dissertation chapters (Methodology, Results, Analysis)	1 week	Partial dissertation draft with methods and results sections
10	Finalize the dissertation (proofreading, formatting, creating figures, TOC, references)	1 week	Final dissertation draft, formatted and ready for submission

3. PROPOSED ENHANCEMENTS IN THIS DISSERTATION

This dissertation introduces several critical enhancements over existing methodologies for predicting hospital readmission in diabetic patients. Traditional approaches frequently suffer from limitations such as la ck of transparency in models, class imbalance, and lack of clinical interpretability. The proposed system addresses these limitations through different strategies, as described below:

A. Integration of Explainable Artificial Intelligence (XAI)

While prior models primarily rely on black-box techniques such as logistic regression or standard deep learning architectures, they often lack interpretability—a key requirement in clinical decision-making. In this study, SHapley Additive exPlanations (SHAP) are employed to enhance model transparency. SHAP is integrated with all trained models to provide both local and global feature attribution. This ensures clinicians can understand the rationale behind individual predictions, thereby increasing trust in the system's outputs.

B. Comparative Model Evaluation

Previous studies often focus on a single classification model, which restricts the scope of performance evaluation. In contrast, this work performs a comprehensive comparison of multiple models—namely, neural networks, logistic regression, decision trees, and XGBoost. Each model is evaluated using consistent metrics, including accuracy, precision, recall, F1-score, and ROC-AUC. This comparative analysis allows for the selection of the most appropriate model in terms of both predictive performance and clinical relevance.

C. Addressing Class Imbalance with SMOTE

Hospital readmission data typically exhibits a high degree of class imbalance, with distinctly fewer readmitted patients. Traditional models tend to bias toward the majority class, leading to the inaccurate of readmit patients being predicted as non-readmits. This work applies the Synthetic Minority Over-sampling Technique (SMOTE) during model training to synthetically balance the dataset. This enhancement ensures better sensitivity to the minority class (i.e., actual readmissions), which is critical in a healthcare setting.

D. Prioritization of Clinically Meaningful Metrics

Unlike conventional approaches that emphasize overall accuracy—often misleading in imbalanced settings—this work prioritizes recall and F1-score as primary evaluation metrics. Recall is particularly critical as it measures the model's ability to correctly identify patients who will be readmitted, thus reducing the risk of undetected cases. Confusion matrices and ROC curves are also employed to visualize performance across various thresholds.

F. Enhanced Data Preparation and Feature Engineering

Many existing models use raw clinical features without adequate preprocessing, which can result in unstable model behavior. This study performs extensive data preprocessing including label encoding, standardization, and feature selection, which improves model robustness and generalization.

4. DATA SOURCES

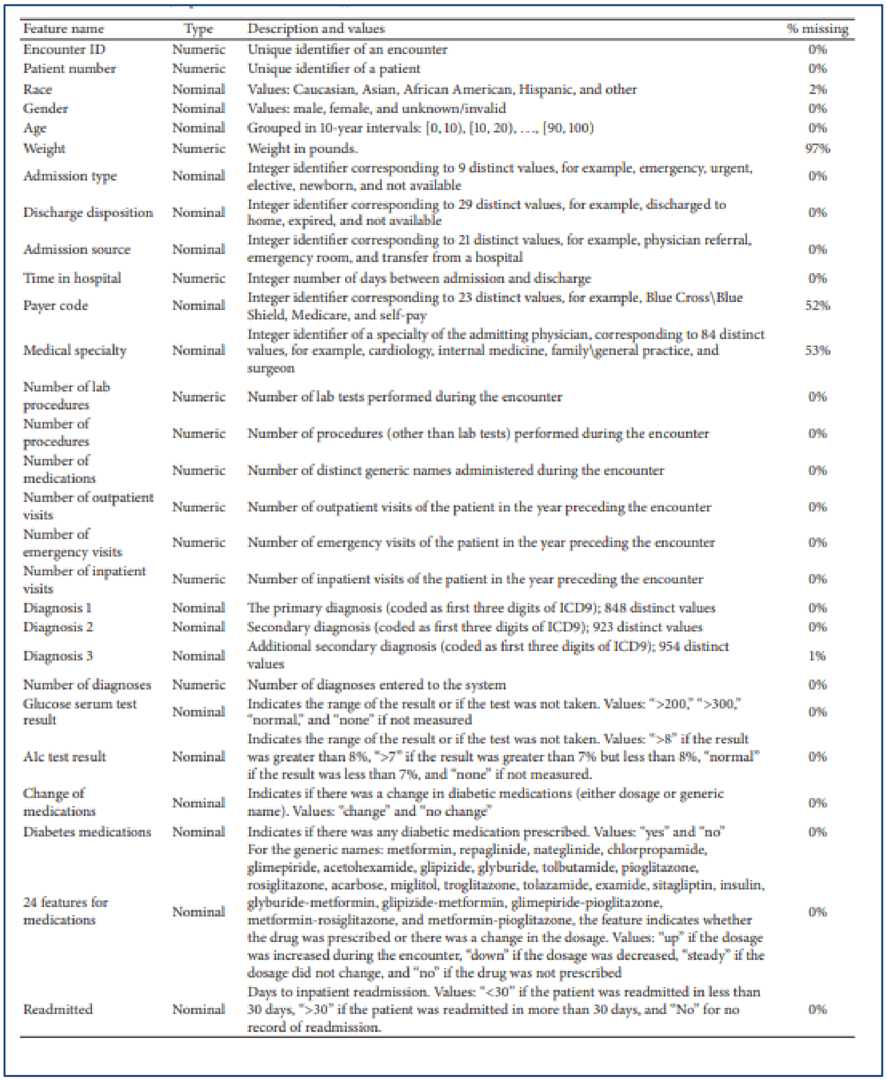
Data	Source
Diabetes 130-US Hospitals for Years 1999-2008	https://archive.ics.uci.edu/dataset/296/diabetes+130-us+hospitals+for+years+1999-2008

5. DATA INGESTION AND EXPLORATORY ANALYSIS

DATA SOURCE AND DESCRIPTION

Sl No	Data	Source	Citation
1	Diabetes 130-US Hospitals for Years 1999-2008	Diabetes Data	J. Clore, K. Cios, J. DeShazo, and B. Strack. "Diabetes 130-US Hospitals for Years 1999-2008," UCI Machine Learning Repository, 2014. [Online]. Available: https://doi.org/10.24432/C5230J.

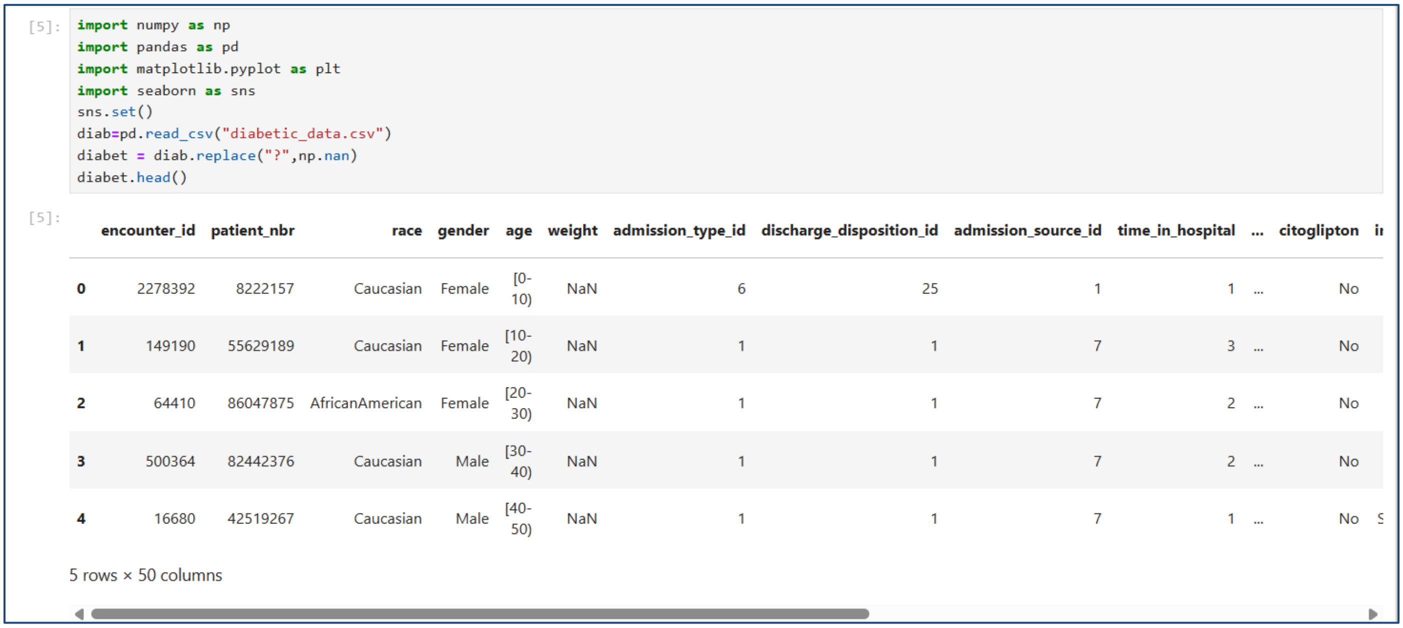
The dataset used in this study contains records of over 100,000 hospital admissions for diabetic patients collected over a 10-year period from 130 hospitals across the United States.
Each entry represents a unique inpatient encounter and includes various demographic, clinical, and administrative details associated with the patient and their treatment during the hospital stay.
The dataset features information such as patient age, gender, race, type of admission, length of stay, number of lab tests and medications, primary and secondary diagnoses, and the use of diabetes-related medications.
Additionally, the dataset indicates whether the patient was readmitted, and if so, whether it occurred within 30 days, after 30 days, or not at all. (This is the target variable)
This makes it particularly well-suited for modelling readmission risk prediction.
Several features require preprocessing due to missing values, categorical data formats, and variations in coding, especially within diagnosis fields.
The target variable for this project is a binary indicator of early readmission (within 30 days), which aligns with real-world efforts to reduce avoidable hospital readmissions among diabetic patients.
Overall, the dataset offers a rich combination of patient-level clinical and administrative features that support both predictive modelling and healthcare outcome analysis.



Feature name	Type	Description and values	% missing
Encounter ID	Numeric	Unique identifier of an encounter	0%
Patient number	Numeric	Unique identifier of a patient	0%
Race	Nominal	Values: Caucasian, Asian, African American, Hispanic, and other	2%
Gender	Nominal	Values: male, female, and unknown/invalid	0%
Age	Nominal	Grouped in 10-year intervals: [0, 10), [10, 20), . . ., [90, 100)	0%
Weight	Numeric	Weight in pounds.	97%
Admission type	Nominal	Integer identifier corresponding to 9 distinct values, for example, emergency, urgent, elective, newborn, and not available	0%
Discharge disposition	Nominal	Integer identifier corresponding to 29 distinct values, for example, discharged to home, expired, and not available	0%
Admission source	Nominal	Integer identifier corresponding to 21 distinct values, for example, physician referral, emergency room, and transfer from a hospital	0%
Time in hospital	Numeric	Integer number of days between admission and discharge	0%
Payer code	Nominal	Integer identifier corresponding to 23 distinct values, for example, Blue Cross\Blue Shield, Medicare, and self-pay	52%
Medical specialty	Nominal	Integer identifier of a specialty of the admitting physician, corresponding to 84 distinct values, for example, cardiology, internal medicine, family\general practice, and surgeon	53%
Number of lab procedures	Numeric	Number of lab tests performed during the encounter	0%
Number of procedures	Numeric	Number of procedures (other than lab tests) performed during the encounter	0%
Number of medications	Numeric	Number of distinct generic names administered during the encounter	0%
Number of outpatient visits	Numeric	Number of outpatient visits of the patient in the year preceding the encounter	0%
Number of emergency visits	Numeric	Number of emergency visits of the patient in the year preceding the encounter	0%
Number of inpatient visits	Numeric	Number of inpatient visits of the patient in the year preceding the encounter	0%
Diagnosis 1	Nominal	The primary diagnosis (coded as first three digits of ICD9); 848 distinct values	0%
Diagnosis 2	Nominal	Secondary diagnosis (coded as first three digits of ICD9); 923 distinct values	0%
Diagnosis 3	Nominal	Additional secondary diagnosis (coded as first three digits of ICD9); 954 distinct values	1%
Number of diagnoses	Numeric	Number of diagnoses entered to the system	0%
Glucose serum test result	Nominal	Indicates the range of the result or if the test was not taken. Values: “>200,” “>300,” “normal,” and “none” if not measured	0%
A1c test result	Nominal	Indicates the range of the result or if the test was not taken. Values: “>8” if the result was greater than 8%, “>7” if the result was greater than 7% but less than 8%, “normal” if the result was less than 7%, and “none” if not measured.	0%
Change of medications	Nominal	Indicates if there was a change in diabetic medications (either dosage or generic name). Values: “change” and “no change”	0%
Diabetes medications	Nominal	Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”	0%
24 features for medications	Nominal	For the generic names: metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, sitagliptin, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, and metformin-pioglitazone, the feature indicates whether the drug was prescribed or there was a change in the dosage. Values: “up” if the dosage was increased during the encounter, “down” if the dosage was decreased, “steady” if the dosage did not change, and “no” if the drug was not prescribed	0%
Readmitted	Nominal	Days to inpatient readmission. Values: “<30” if the patient was readmitted in less than 30 days, “>30” if the patient was readmitted in more than 30 days, and “No” for no record of readmission.	0%

Figure 1: The above image summarises the attributes available as a part of the Data set used.

EDA – EXPLORATORY DATA ANALYSIS, DATA EXPLORATION AND VISUALIZATION



```
[5]: import numpy as np
     import pandas as pd
     import matplotlib.pyplot as plt
     import seaborn as sns
     sns.set()
     diab=pd.read_csv("diabetic_data.csv")
     diabet = diab.replace("?",np.nan)
     diabet.head()
```

[5]:

	encounter_id	patient_nbr	race	gender	age	weight	admission_type_id	discharge_disposition_id	admission_source_id	time_in_hospital	...	citoglipton	ir
0	2278392	8222157	Caucasian	Female	[0-10)	NaN	6	25	1	1	...	No	
1	149190	55629189	Caucasian	Female	[10-20)	NaN	1	1	7	3	...	No	
2	64410	86047875	AfricanAmerican	Female	[20-30)	NaN	1	1	7	2	...	No	
3	500364	82442376	Caucasian	Male	[30-40)	NaN	1	1	7	2	...	No	
4	16680	42519267	Caucasian	Male	[40-50)	NaN	1	1	7	1	...	No	S

5 rows × 50 columns

Figure 2: Importing libraries and reading CSV

Code explanation:

We import pandas, numpy and the visulaisation libraries. Also, we import the libraries and replace the missing values with np.nan and see the first 50 rows of the data set.

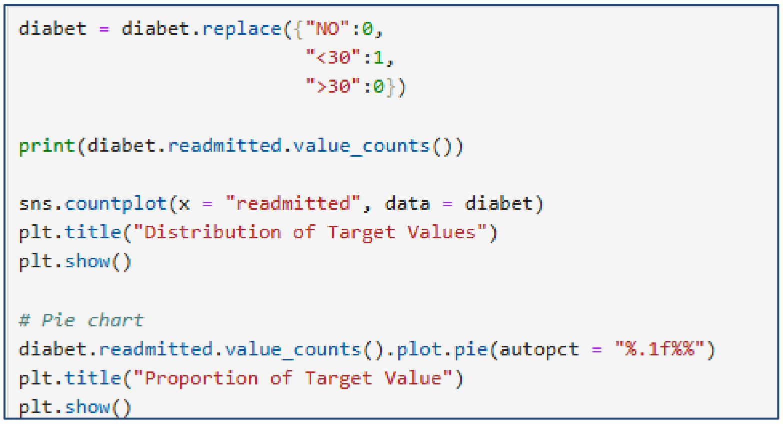
Readmitted

This column our target feature. It is about "days to inpatient readmission"

- If the patient was readmitted in less than 30 days "<30"
- if the patient was readmitted in more than 30 days ">30"
- If there is no record "NO"

We decided to reduce these values to two and map them according to the following rule:

- NO and >30-> 0
- <30 -> 1



```
diabet = diabet.replace({"NO":0,
                         "<30":1,
                         ">30":0})

print(diabet.readmitted.value_counts())

sns.countplot(x = "readmitted", data = diabet)
plt.title("Distribution of Target Values")
plt.show()

# Pie chart
diabet.readmitted.value_counts().plot.pie(autopct = "%.1f%%")
plt.title("Proportion of Target Value")
plt.show()
```

Figure 3: Target Variable Visualisation Code

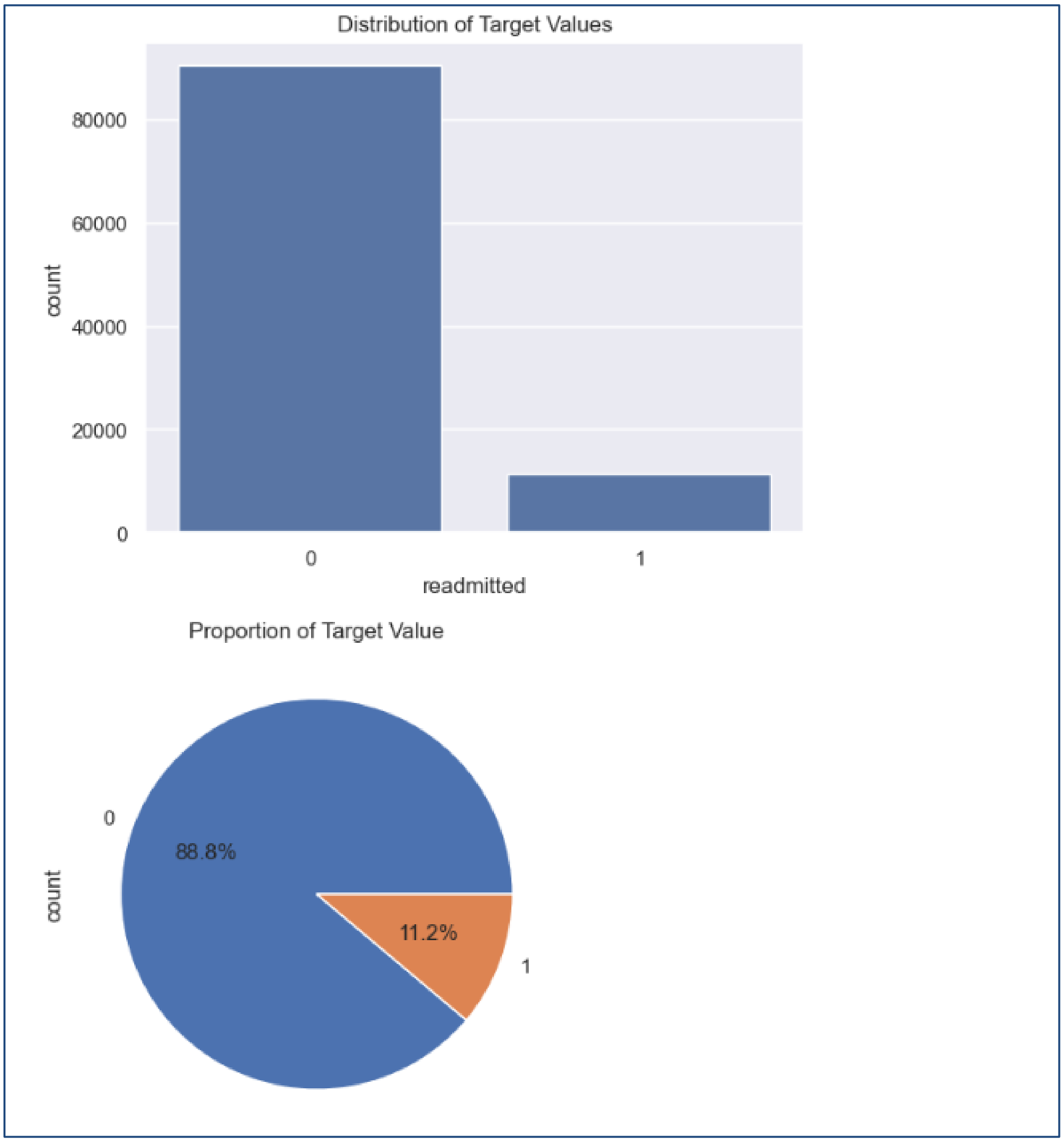


Figure 4: Target Value Visualisation - Bar Chart and Pie Chart

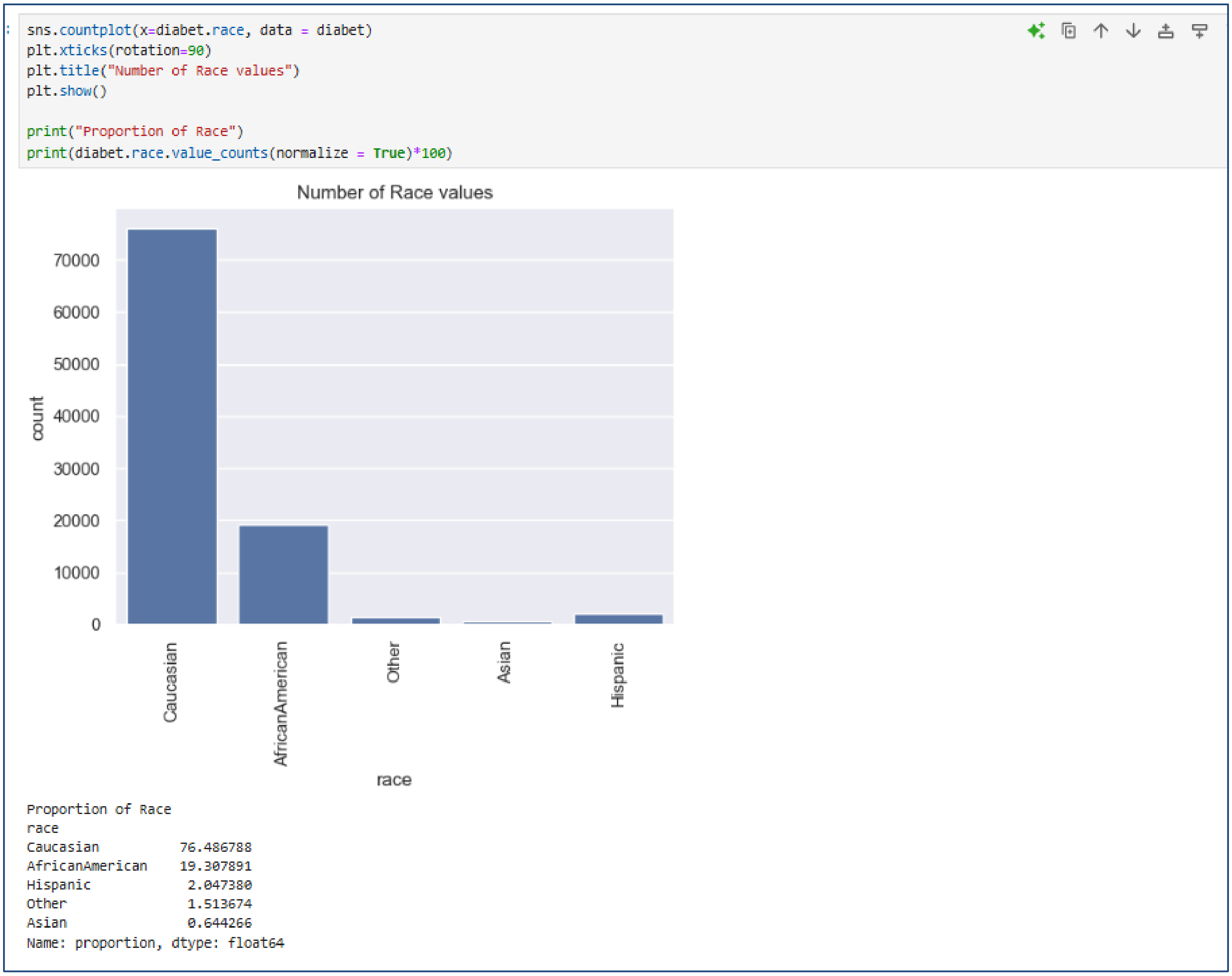
Inference:

We can see from this that the dataset is imbalanced. Therefore, we need to use strategies such as smote to balance the dataset to even out the class distribution.

Race

We have 5 different races value, these are:

- Caucasian
- AfricanAmerican
- Hispanic
- Asian
- Other



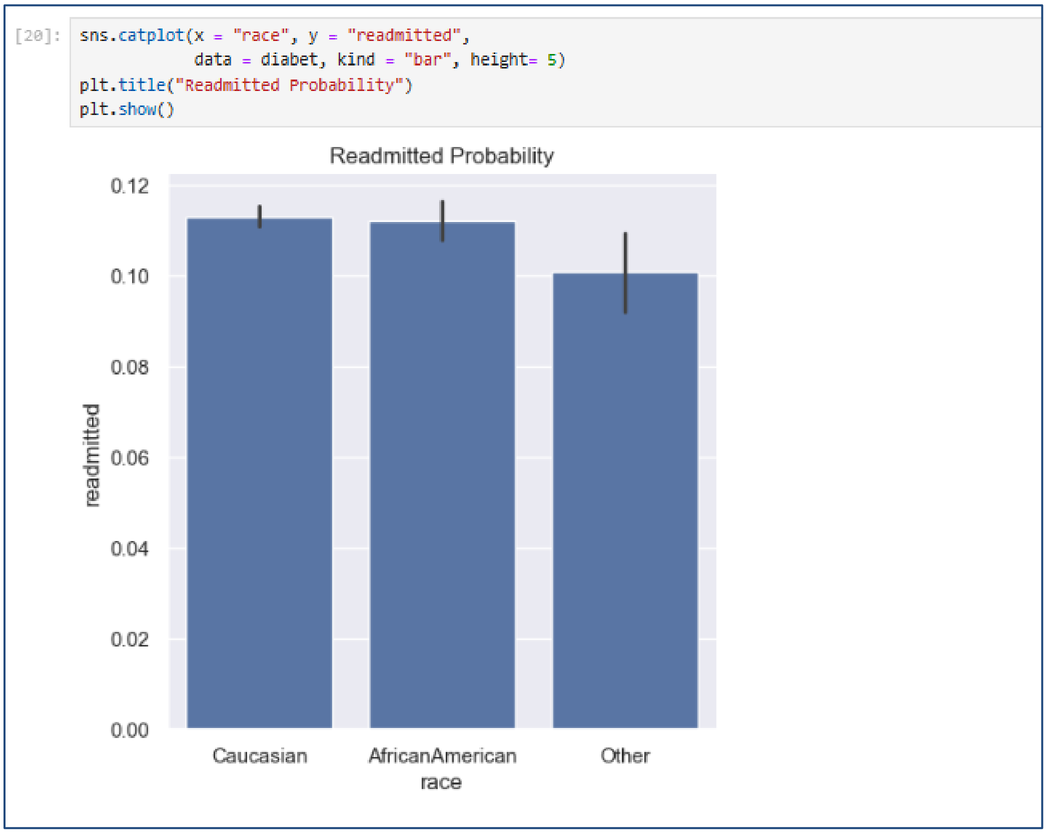
```
sns.countplot(x=diabet.race, data = diabet)
plt.xticks(rotation=90)
plt.title("Number of Race values")
plt.show()

print("Proportion of Race")
print(diabet.race.value_counts(normalize = True)*100)
```

```
Proportion of Race
race
Caucasian          76.486788
AfricanAmerican    19.307891
Hispanic            2.047380
Other               1.513674
Asian               0.644266
Name: proportion, dtype: float64
```

Figure 5: Race Categorical Variable visualization and Code

Inference:

We see that majority of the individuals belong to Caucasian followed by AfricanAmerican origin.



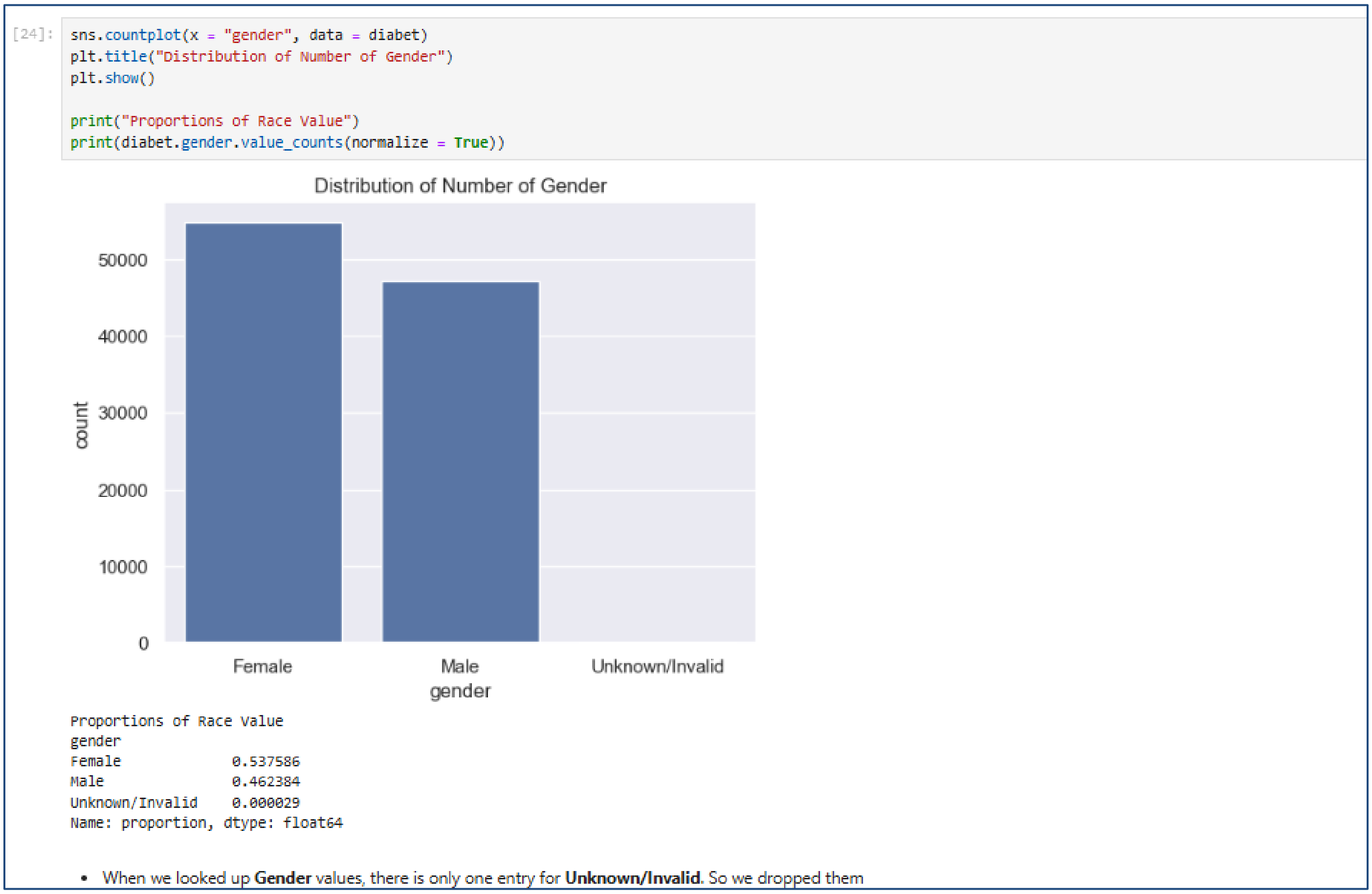
```
[20]: sns.catplot(x = "race", y = "readmitted",
                  data = diabet, kind = "bar", height= 5)
      plt.title("Readmitted Probability")
      plt.show()
```

Figure 6: Readmission probability based on Race

Inference:

Most of the readmitted patients were Caucasians followed by AfricanAmericans.

Gender:

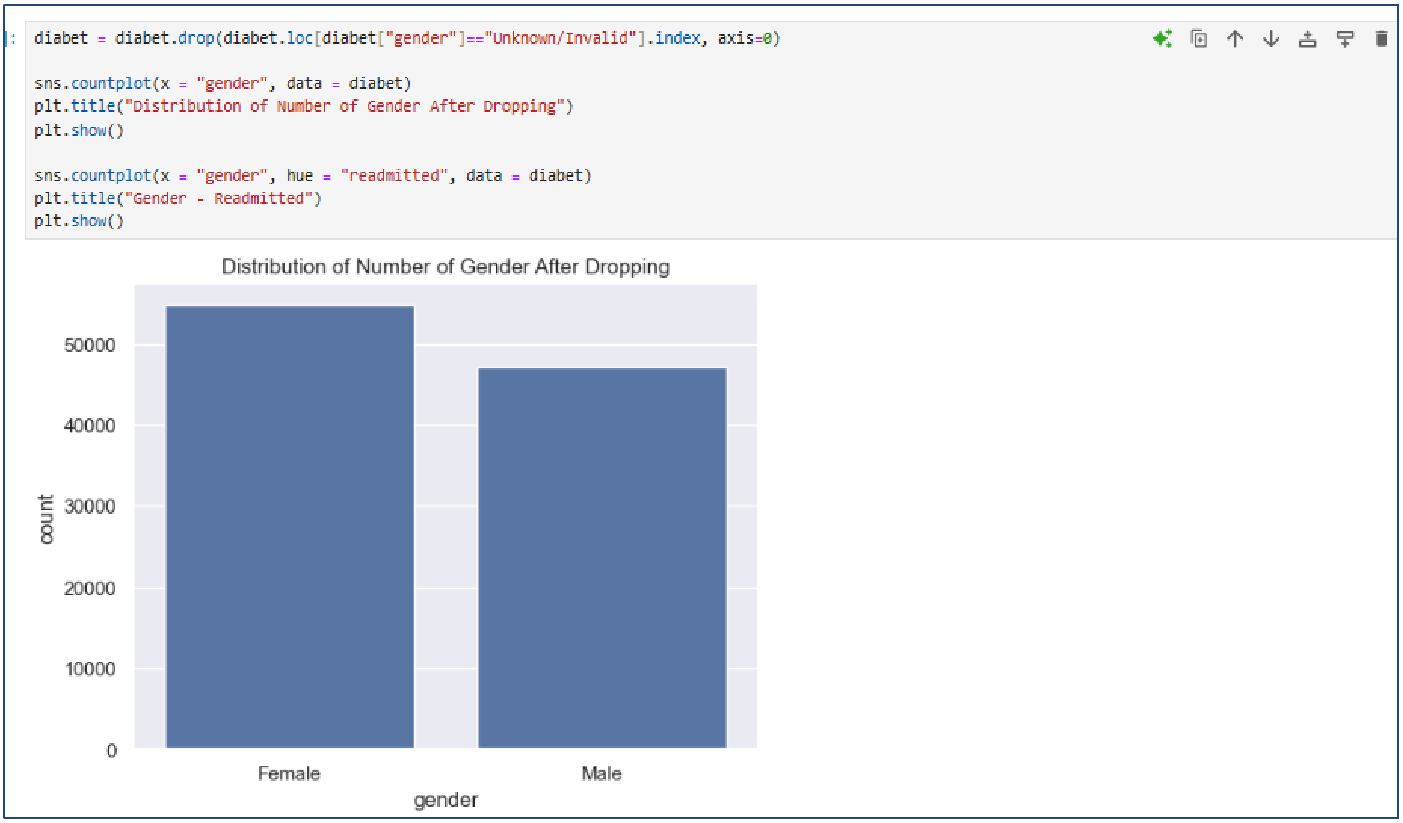


```
[24]: sns.countplot(x = "gender", data = diabet)
plt.title("Distribution of Number of Gender")
plt.show()

print("Proportions of Race Value")
print(diabet.gender.value_counts(normalize = True))
```

```
Proportions of Race Value
gender
Female            0.537586
Male              0.462384
Unknown/Invalid   0.000029
Name: proportion, dtype: float64
```

- When we looked up **Gender** values, there is only one entry for **Unknown/Invalid**. So we dropped them

Figure 7: Gender Column Categorical Visualisation



```
diabet = diabet.drop(diabet.loc[diabet["gender"]=="Unknown/Invalid"].index, axis=0)

sns.countplot(x = "gender", data = diabet)
plt.title("Distribution of Number of Gender After Dropping")
plt.show()

sns.countplot(x = "gender", hue = "readmitted", data = diabet)
plt.title("Gender - Readmitted")
plt.show()
```

Figure 8: Gender Column Categorical Visualisation after dropping NA

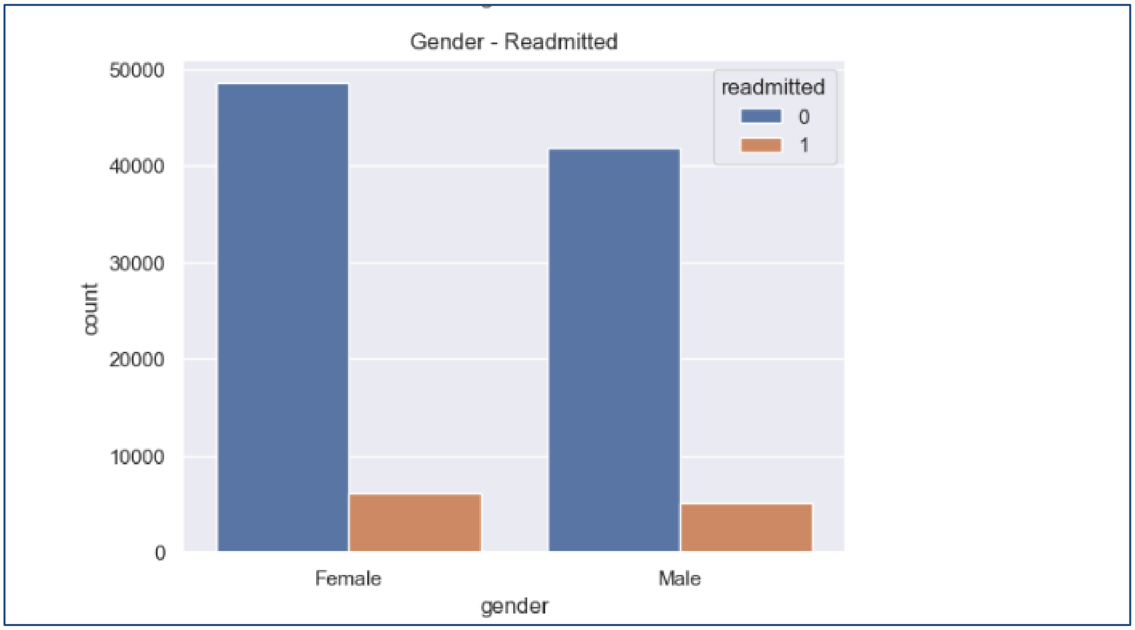
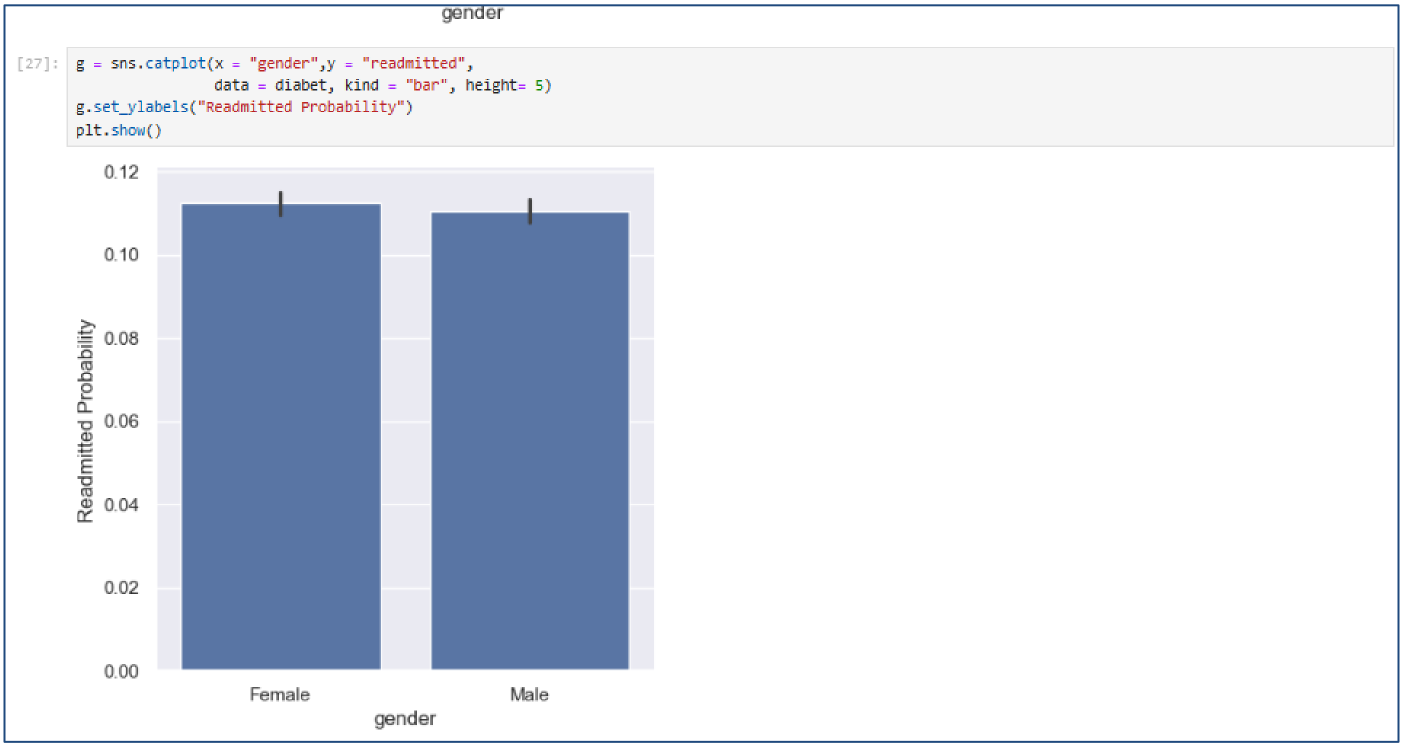


Figure 9: Distribution of Readitted Vs Readmitted patients between Females and Males



```
g = sns.catplot(x = "gender",y = "readmitted",
                data = diabet, kind = "bar", height= 5)
g.set_ylabels("Readmitted Probability")
plt.show()
```

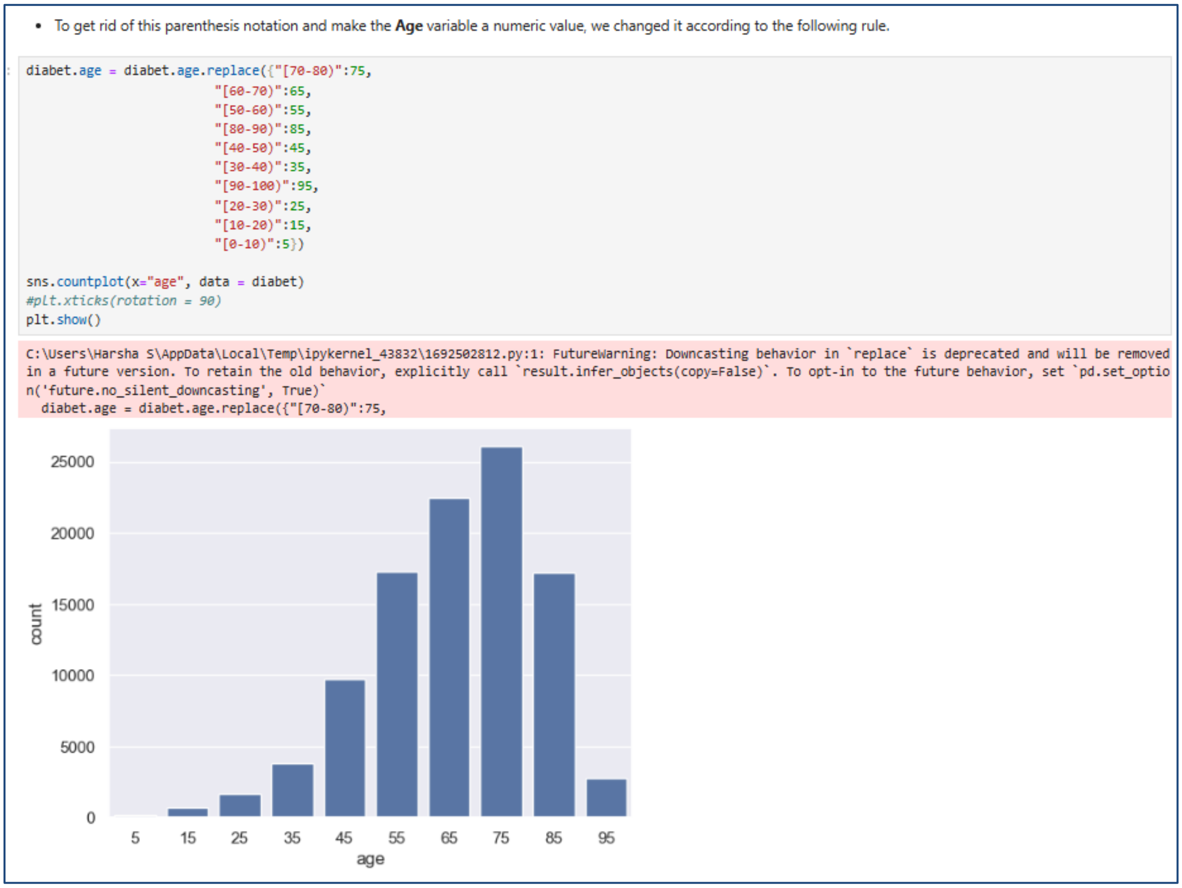
Figure 10:Readmitted Probability based on Gender

Inference:

There is one null value for gender, we remove that row.

We see that the distribution of both genders is almost near equal, but with females being slightly more in the case of readmits when compared to their male counterparts- hence females are slightly more susceptible to readmission risk.

Age:

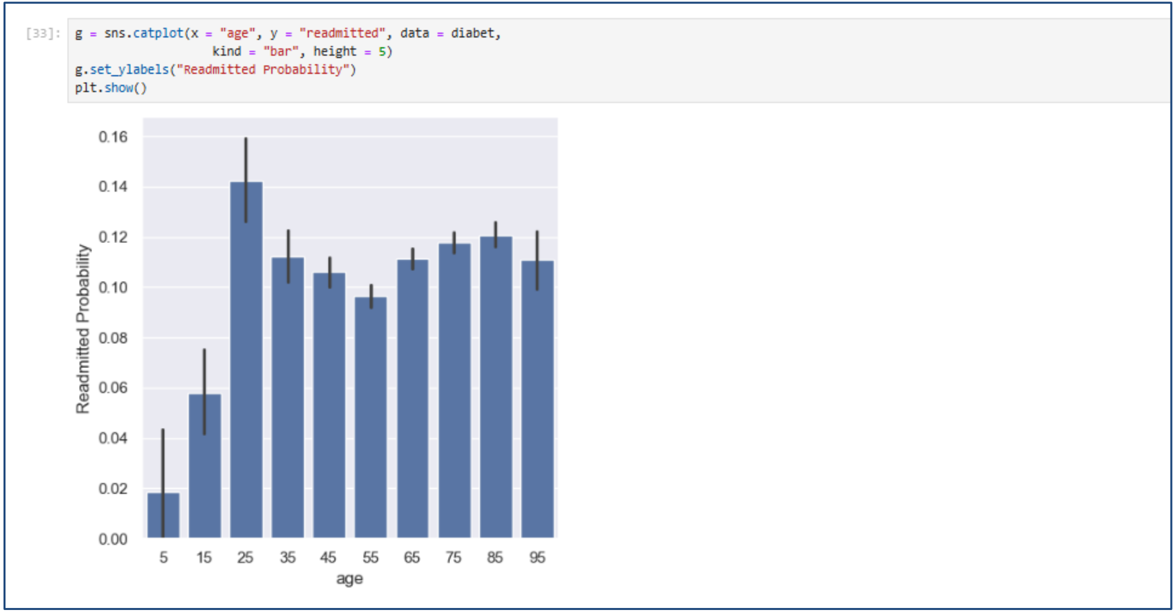


- To get rid of this parenthesis notation and make the **Age** variable a numeric value, we changed it according to the following rule.

```
diabet.age = diabet.age.replace({"[70-80)":75,
                          "[60-70)":65,
                          "[50-60)":55,
                          "[80-90)":85,
                          "[40-50)":45,
                          "[30-40)":35,
                          "[90-100)":95,
                          "[20-30)":25,
                          "[10-20)":15,
                          "[0-10)":5})

sns.countplot(x="age", data = diabet)
#plt.xticks(rotation = 90)
plt.show()
```

```
C:\Users\Harsha S\AppData\Local\Temp\ipykernel_43832\1692502812.py:1: FutureWarning: Downcasting behavior in `replace` is deprecated and will be removed
in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_optio
n('future.no_silent_downcasting', True)`
  diabet.age = diabet.age.replace({"[70-80)":75,
```

Figure 11: Age based histogram distribution plot



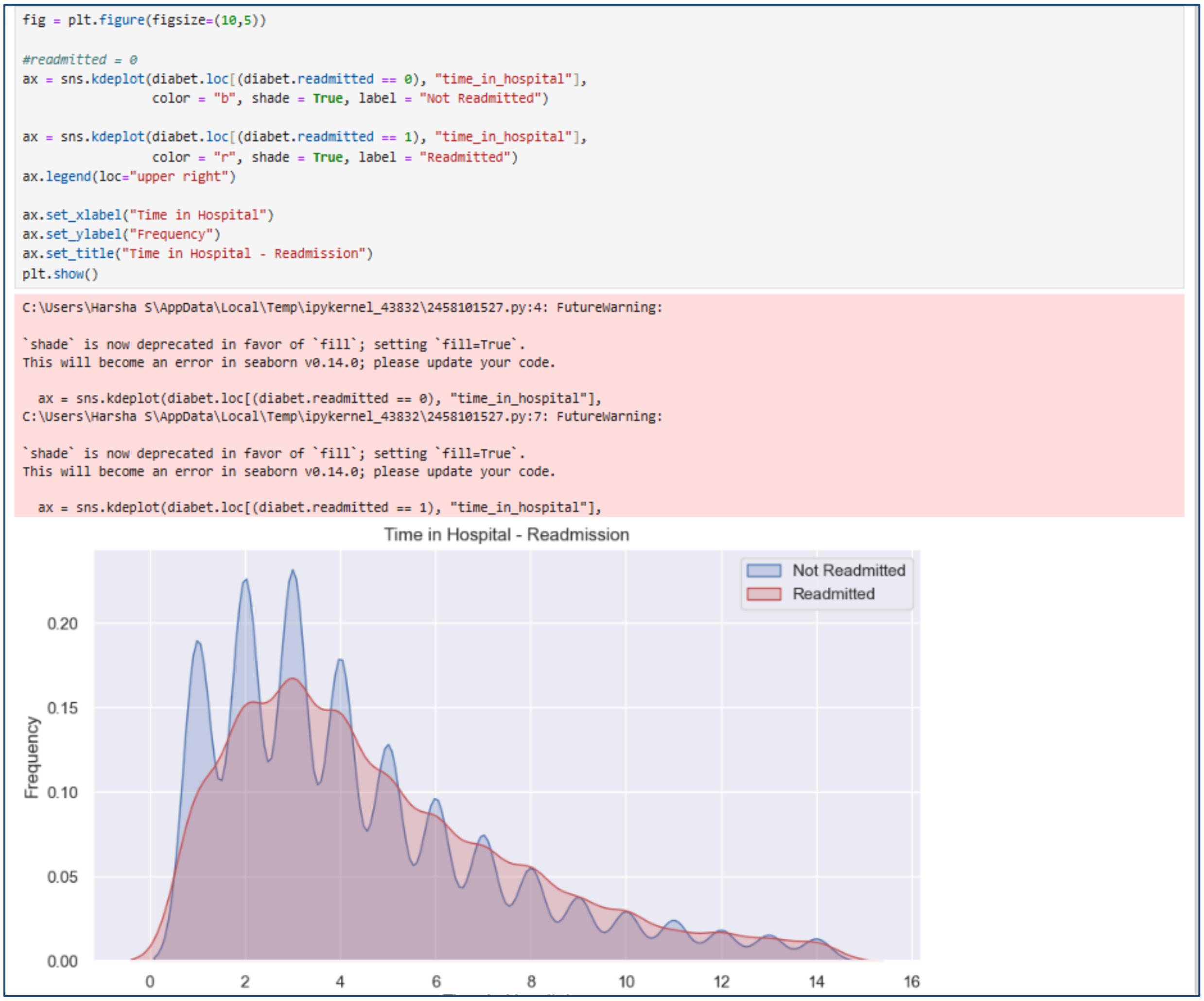
```
[33]: g = sns.catplot(x = "age", y = "readmitted", data = diabet,
                      kind = "bar", height = 5)
      g.set_ylabels("Readmitted Probability")
      plt.show()
```

Figure 12: Readmitted Probability Vs. Age Distribution

Inference:

Inference:
We can see that the population of readmits is significantly higher in the age group of 25.
Also, we can see that readmits are higher in the adult population of above 25 years of age.

Hospital stay:



```
fig = plt.figure(figsize=(10,5))

#readmitted = 0
ax = sns.kdeplot(diabet.loc[(diabet.readmitted == 0), "time_in_hospital"],
                 color = "b", shade = True, label = "Not Readmitted")

ax = sns.kdeplot(diabet.loc[(diabet.readmitted == 1), "time_in_hospital"],
                 color = "r", shade = True, label = "Readmitted")
ax.legend(loc="upper right")

ax.set_xlabel("Time in Hospital")
ax.set_ylabel("Frequency")
ax.set_title("Time in Hospital - Readmission")
plt.show()
```

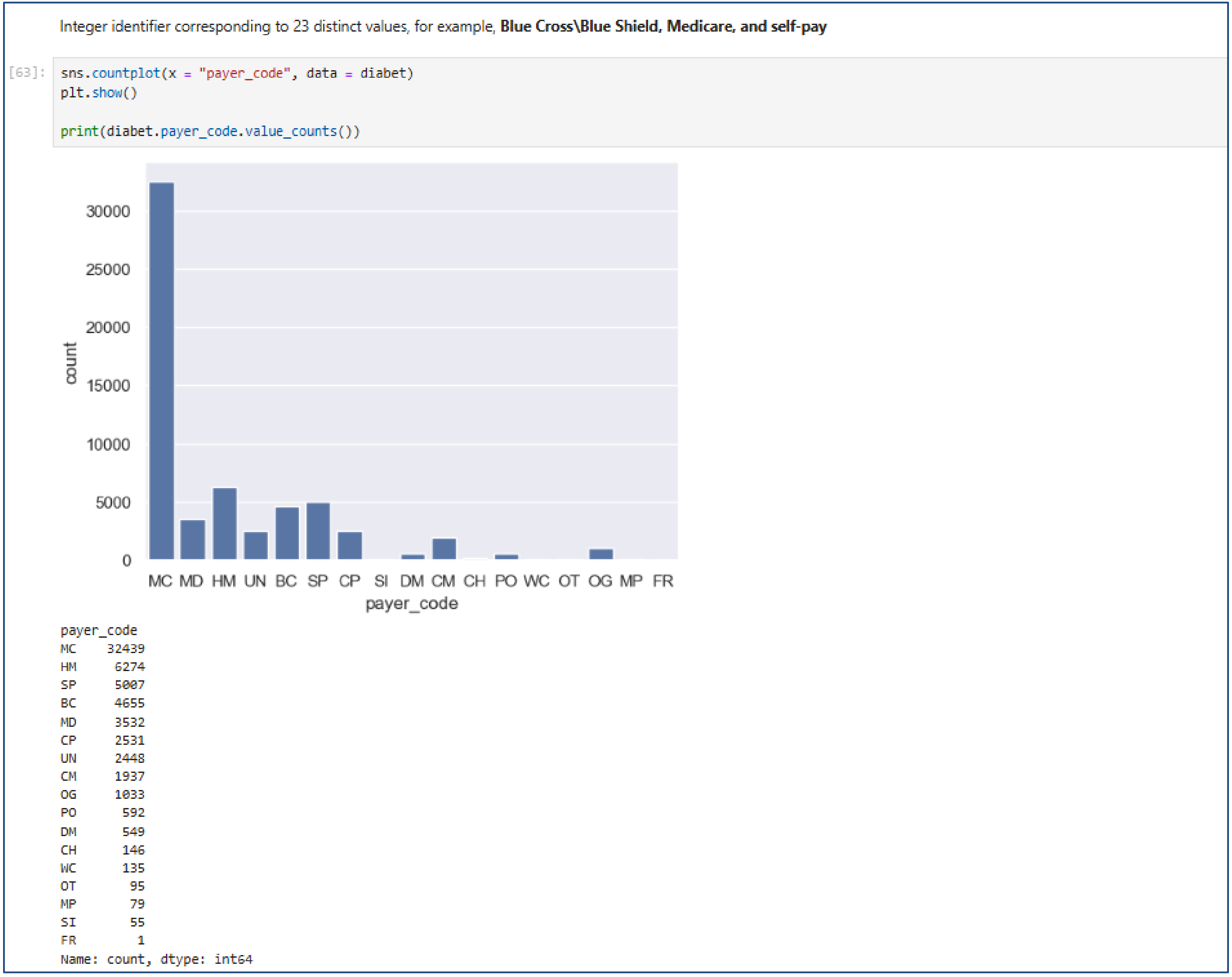
```
C:\Users\Harsha S\AppData\Local\Temp\ipykernel_43832\2458101527.py:4: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  ax = sns.kdeplot(diabet.loc[(diabet.readmitted == 0), "time_in_hospital"],
C:\Users\Harsha S\AppData\Local\Temp\ipykernel_43832\2458101527.py:7: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  ax = sns.kdeplot(diabet.loc[(diabet.readmitted == 1), "time_in_hospital"],
```

Figure 13: KDE Plot for Hospital Stay Duration and Readmittion probability

Inference:

We can see that patients mostly stayed In hospital for 2-3 days.

Payer Code:

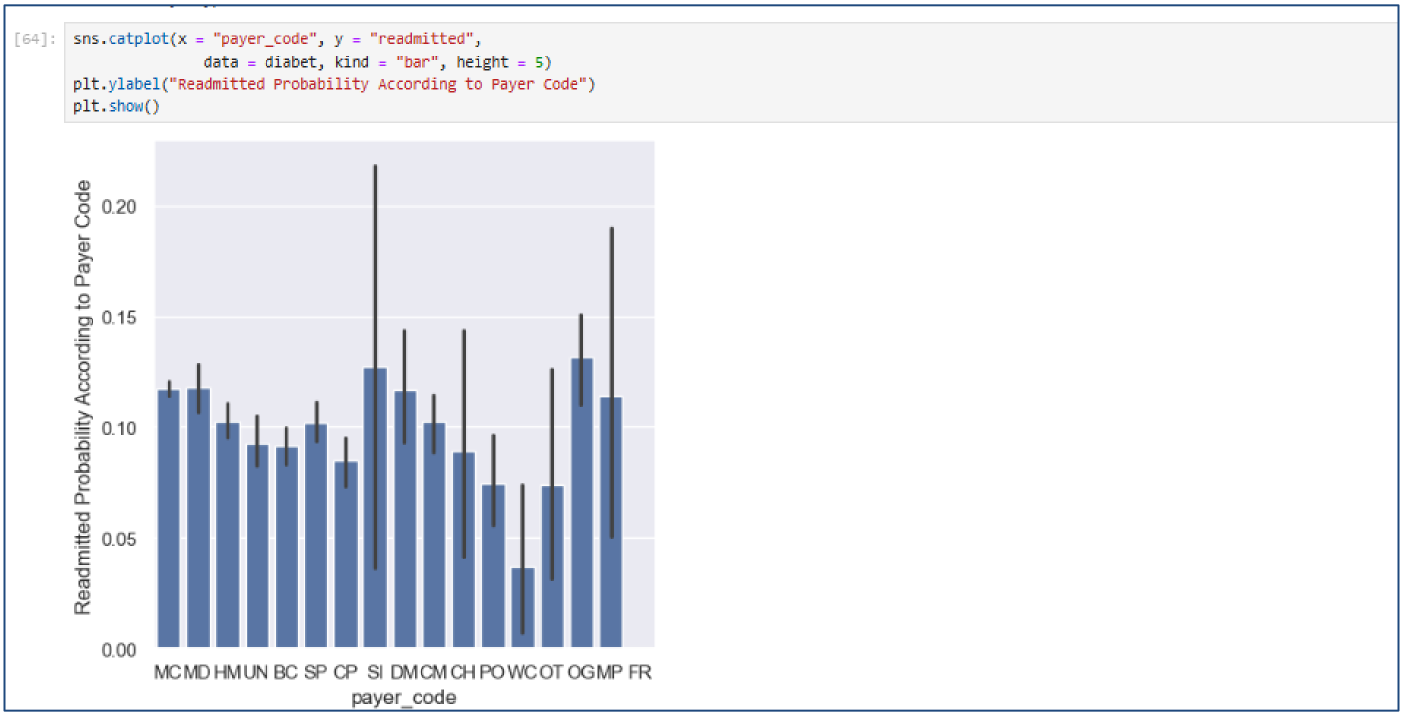


Integer identifier corresponding to 23 distinct values, for example, **Blue Cross\Blue Shield, Medicare, and self-pay**

```
[63]: sns.countplot(x = "payer_code", data = diabet)
      plt.show()

      print(diabet.payer_code.value_counts())
```

```
payer_code
MC    32439
HM     6274
SP     5007
BC     4655
MD     3532
CP     2531
UN     2448
CM     1937
OG     1033
PO      592
DM      549
CH      146
WC      135
OT       95
MP       79
SI       55
FR        1
Name: count, dtype: int64
```

Figure 14: Count of Patients in each Payer Code Category



```
[64]: sns.catplot(x = "payer_code", y = "readmitted",
                  data = diabet, kind = "bar", height = 5)
      plt.ylabel("Readmitted Probability According to Payer Code")
      plt.show()
```

Readmitted Probability According to Payer Code
0.20
0.15
0.10
0.05
0.00
MC MD HM UN BC SP CP SI DM CM CH PO WC OT OG MP FR
payer_code

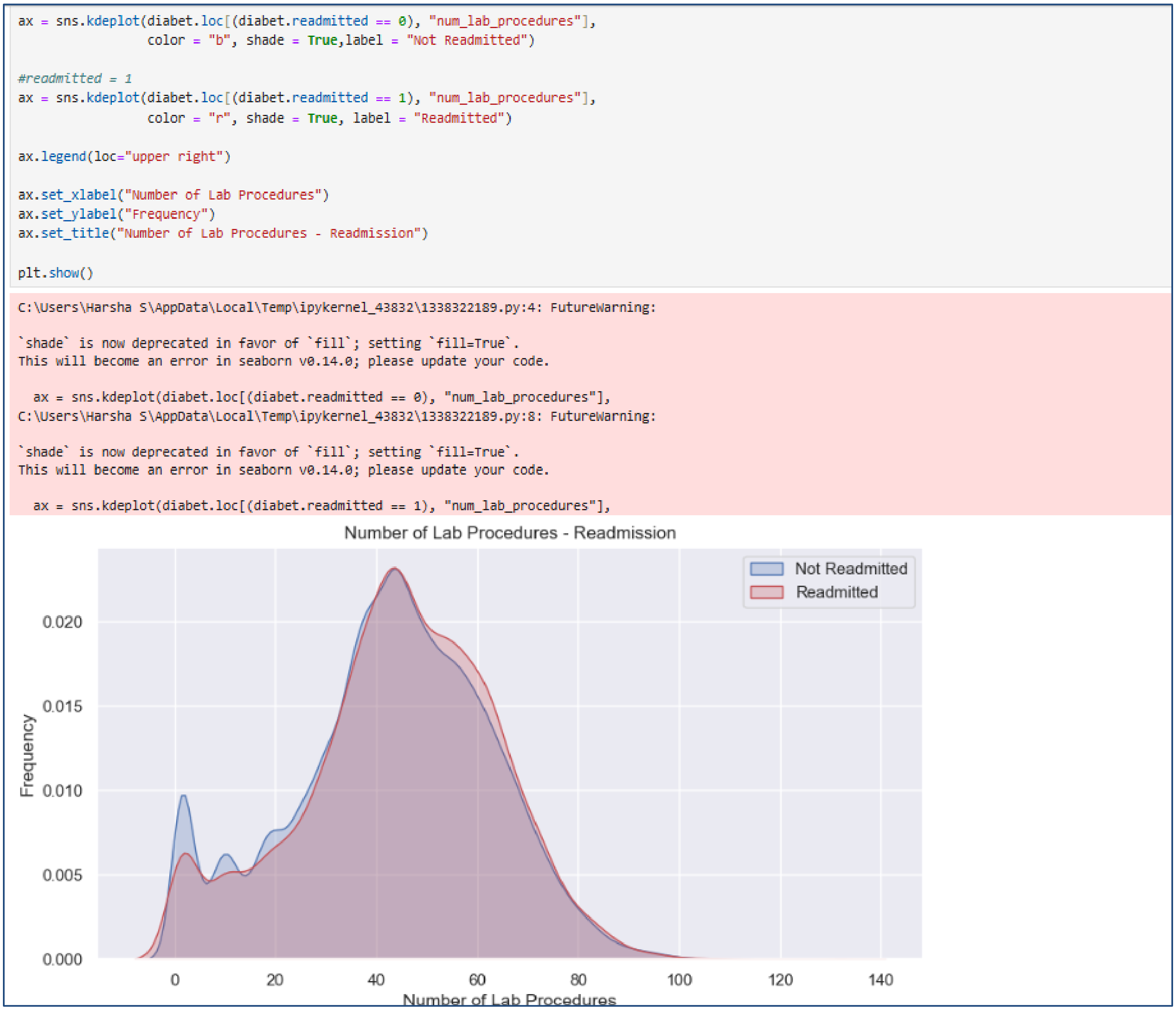
Figure 15: Readmitted Probability for each Payer Code

Inference:

We see that patients with payer code SI have the highest value of readmits, even though MC has the highest number of patients occurrences with it as payer code.

Number of Lab Procedures

Number of lab tests performed during the encounter



```
ax = sns.kdeplot(diabet.loc[(diabet.readmitted == 0), "num_lab_procedures"],
                 color = "b", shade = True,label = "Not Readmitted")

#readmitted = 1
ax = sns.kdeplot(diabet.loc[(diabet.readmitted == 1), "num_lab_procedures"],
                 color = "r", shade = True, label = "Readmitted")

ax.legend(loc="upper right")

ax.set_xlabel("Number of Lab Procedures")
ax.set_ylabel("Frequency")
ax.set_title("Number of Lab Procedures - Readmission")

plt.show()
```

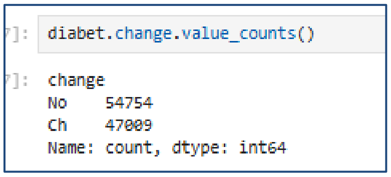
```
C:\Users\Harsha S\AppData\Local\Temp\ipykernel_43832\1338322189.py:4: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  ax = sns.kdeplot(diabet.loc[(diabet.readmitted == 0), "num_lab_procedures"],
C:\Users\Harsha S\AppData\Local\Temp\ipykernel_43832\1338322189.py:8: FutureWarning:

`shade` is now deprecated in favor of `fill`; setting `fill=True`.
This will become an error in seaborn v0.14.0; please update your code.

  ax = sns.kdeplot(diabet.loc[(diabet.readmitted == 1), "num_lab_procedures"],
```

Figure 16: KDE Plot showing distribution of Readmission/Non-Readmitted Count with Respect to the Number of Lab Procedures

Inferences:

We can see that the average number of lab procedures performed is 45 roughly in the case of both readmits and non-readmits.

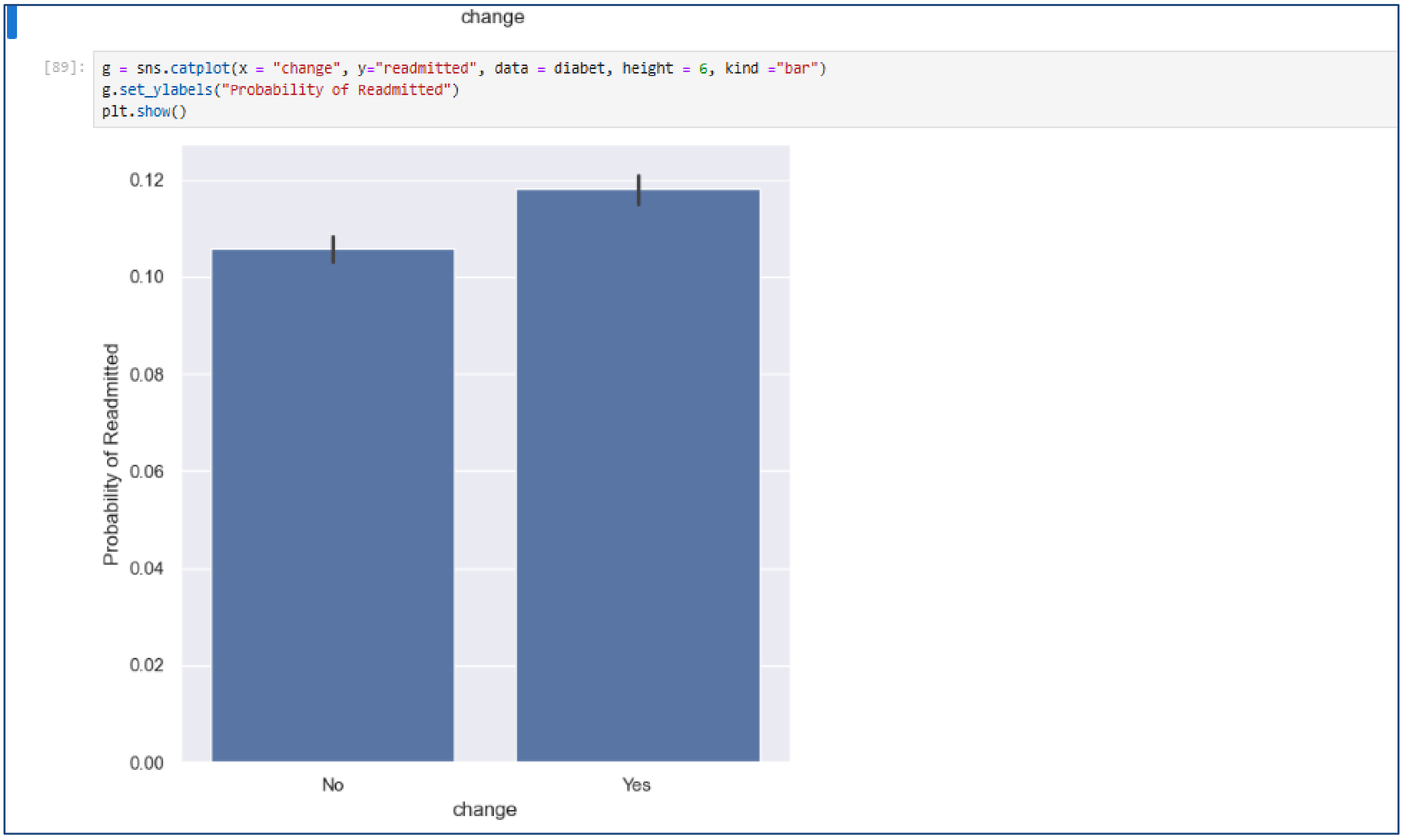
Change
Indicates if there was a change in diabetic medications (either dosage or generic name). Values:
“change”
“no change”



```
diabet.change.value_counts()
```

```
change
No    54754
Ch    47009
Name: count, dtype: int64
```

Figure 17: Change in Medication (Change or No Change) Count



```
[89]: g = sns.catplot(x = "change", y="readmitted", data = diabet, height = 6, kind ="bar")
      g.set_ylabels("Probability of Readmitted")
      plt.show()
```

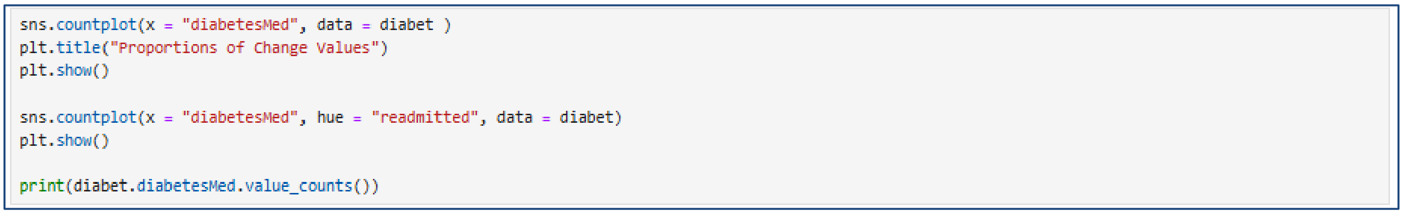
Figure 18: Readmission probability Vs Change in Medication

Inference:

When there is change is medication there are slightly higher chances of readmits.

Diabetes medications:

Indicates if there was any diabetic medication prescribed. Values: “yes” and “no”



```
sns.countplot(x = "diabetesMed", data = diabet )
plt.title("Proportions of Change Values")
plt.show()

sns.countplot(x = "diabetesMed", hue = "readmitted", data = diabet)
plt.show()

print(diabet.diabetesMed.value_counts())
```

Figure 19: Code for Diabetes Medication Column Visualisation

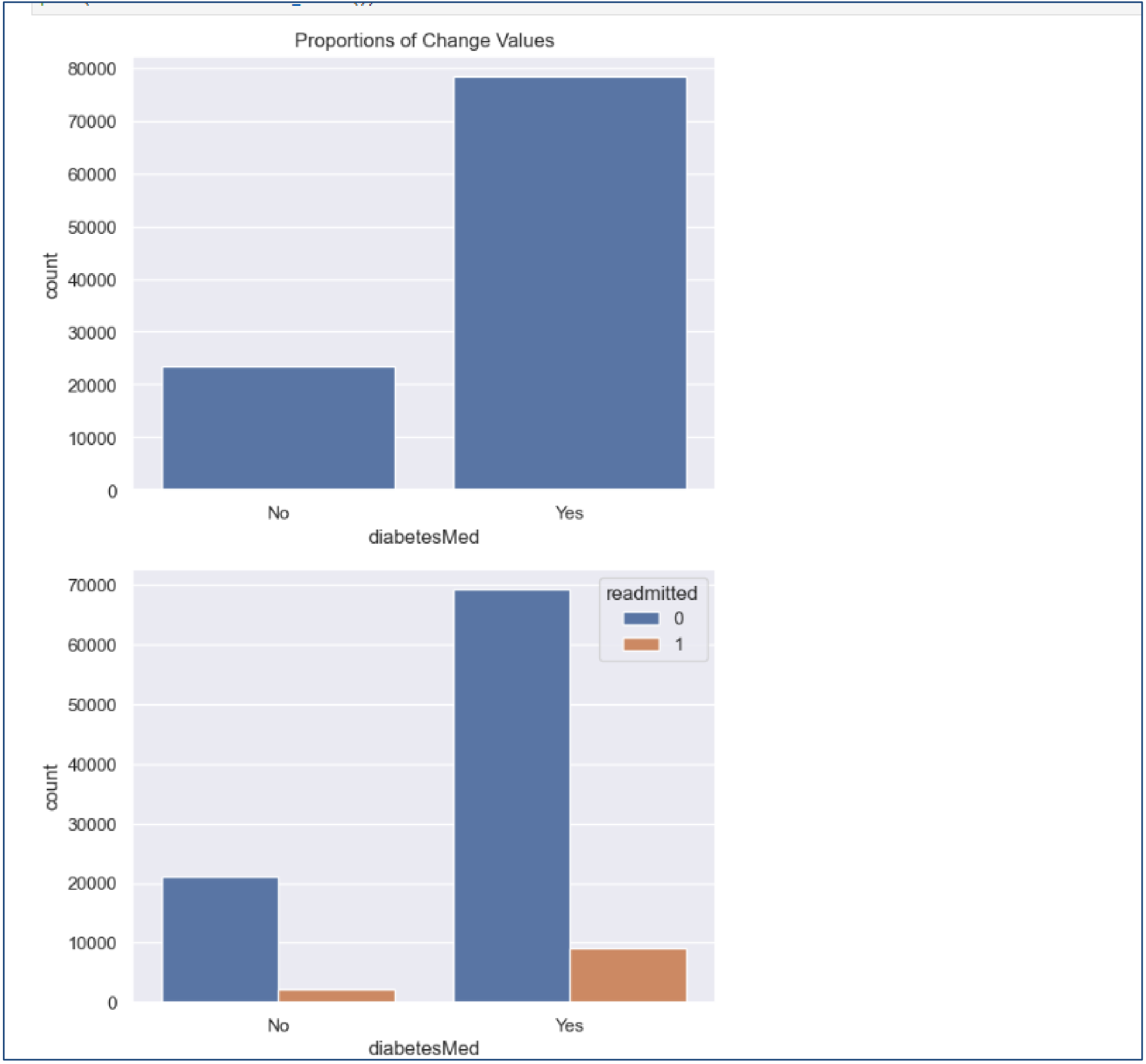
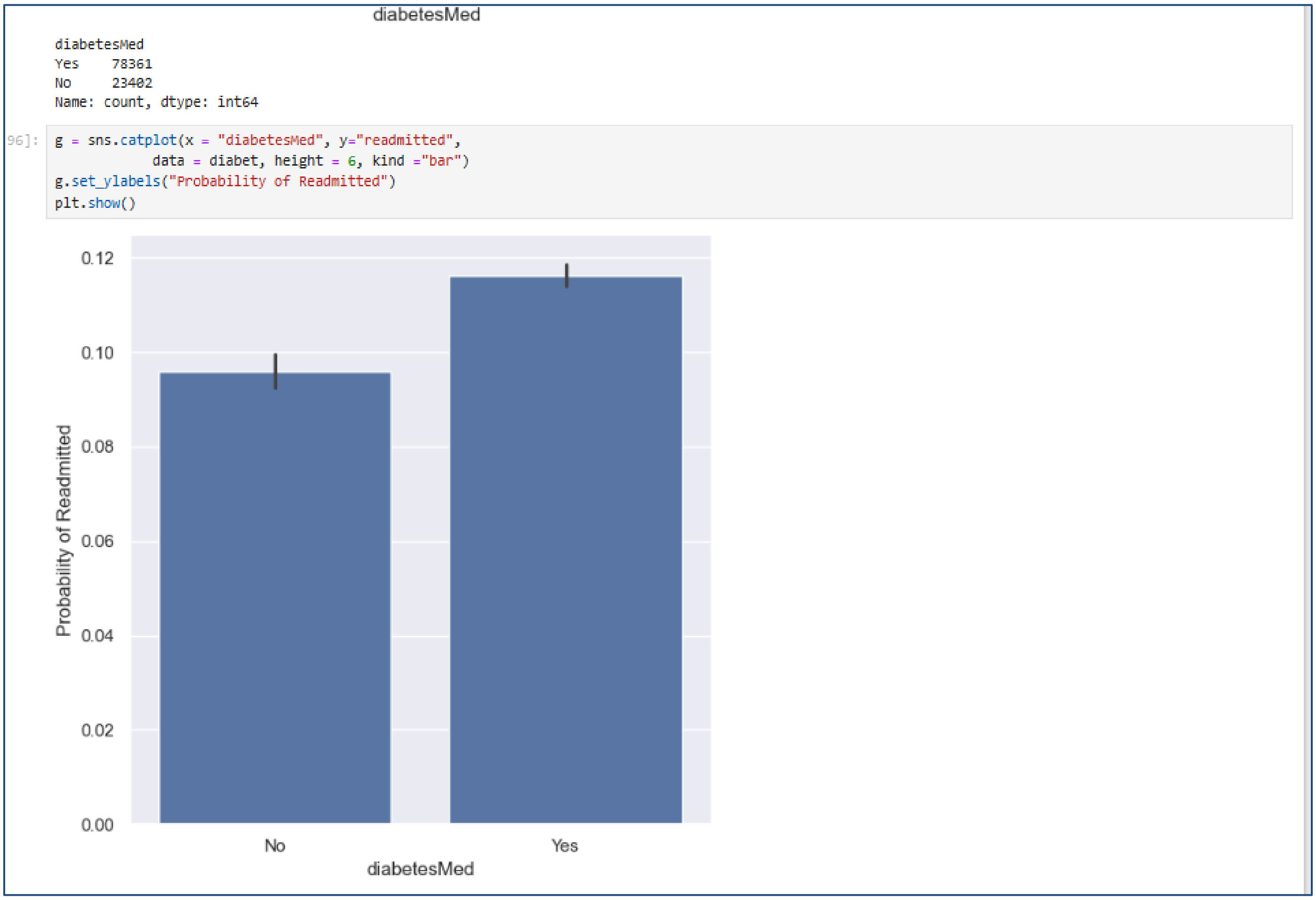


Figure 20: Distribution of patients who were given Diabetes Medication and those who were not,



```
diabetesMed
Yes    78361
No     23402
Name: count, dtype: int64
```

```
g = sns.catplot(x = "diabetesMed", y="readmitted",
            data = diabet, height = 6, kind ="bar")
g.set_ylabels("Probability of Readmitted")
plt.show()
```

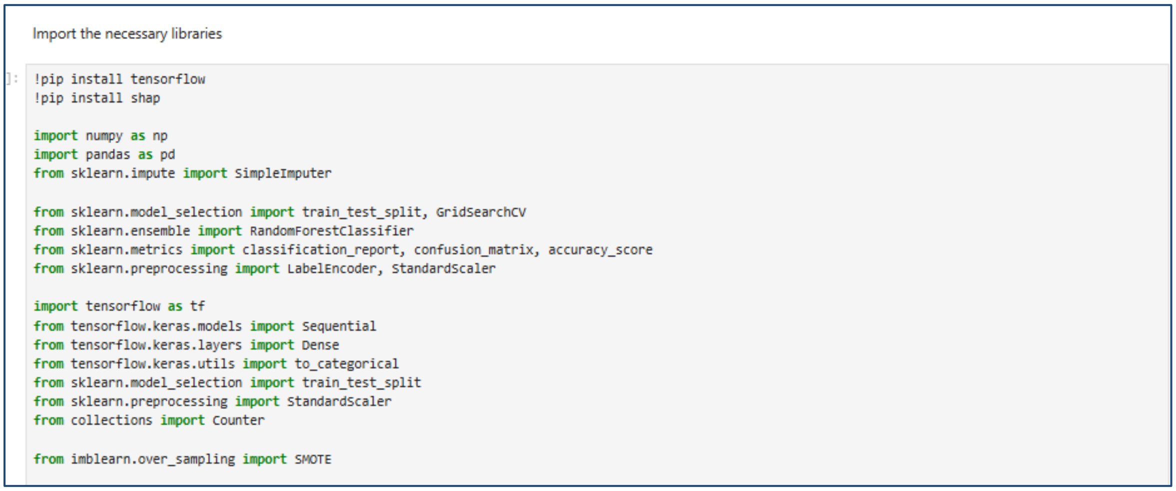
Figure 21: Readmission probability of people who received and did not receive Diabetic Medication

Inferences:

We see that the people given Diabetes medication is higher than those who didn’t get any diabetes medication.

The probability of readmit is more in the case of people who received medication.

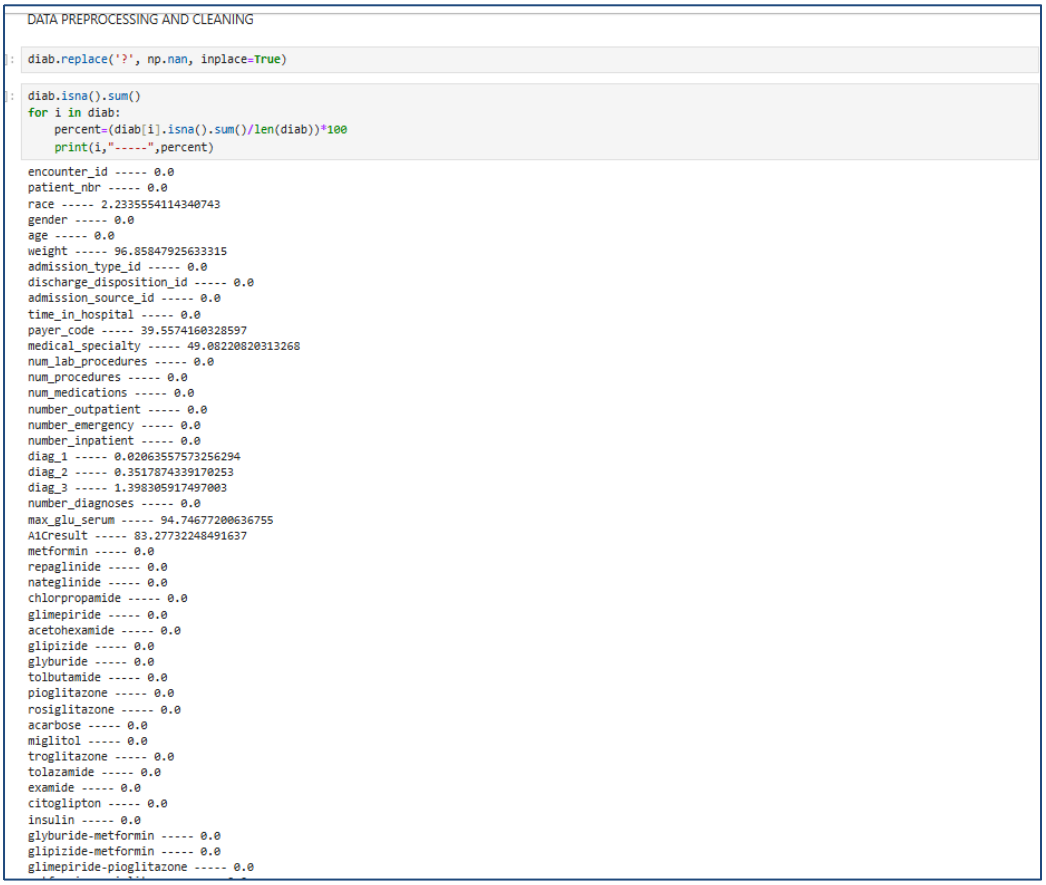
Data pre-processing



Import the necessary libraries

```
!pip install tensorflow
!pip install shap

import numpy as np
import pandas as pd
from sklearn.impute import SimpleImputer

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix, accuracy_score
from sklearn.preprocessing import LabelEncoder, StandardScaler

import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.utils import to_categorical
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from collections import Counter

from imblearn.over_sampling import SMOTE
```

Figure 22: Importing LIbraries



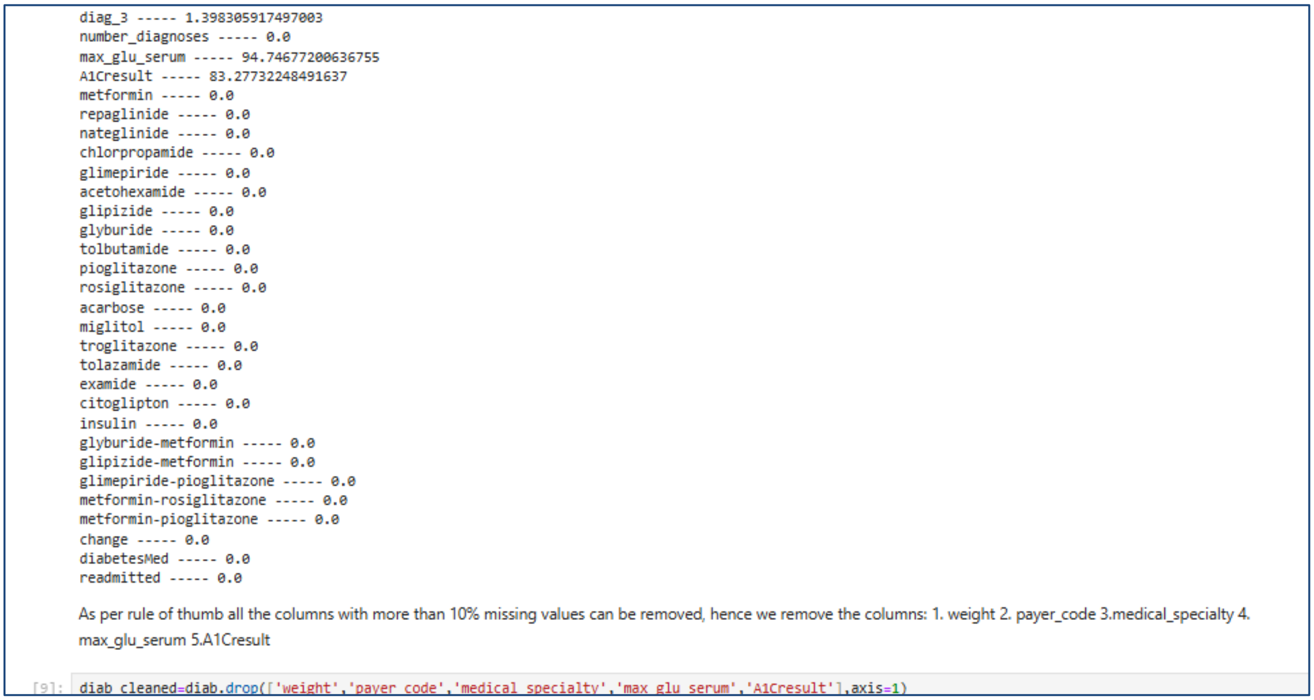
DATA PREPROCESSING AND CLEANING

```
diab.replace('?', np.nan, inplace=True)
```

```
diab.isna().sum()
for i in diab:
    percent=(diab[i].isna().sum()/len(diab))*100
    print(i,"-----",percent)
```

```
encounter_id ----- 0.0
patient_nbr ----- 0.0
race ----- 2.2335554114340743
gender ----- 0.0
age ----- 0.0
weight ----- 96.85847925633315
admission_type_id ----- 0.0
discharge_disposition_id ----- 0.0
admission_source_id ----- 0.0
time_in_hospital ----- 0.0
payer_code ----- 39.5574160328597
medical_specialty ----- 49.08220820313268
num_lab_procedures ----- 0.0
num_procedures ----- 0.0
num_medications ----- 0.0
number_outpatient ----- 0.0
number_emergency ----- 0.0
number_inpatient ----- 0.0
diag_1 ----- 0.02063557573256294
diag_2 ----- 0.3517874339170253
diag_3 ----- 1.398305917497003
number_diagnoses ----- 0.0
max_glu_serum ----- 94.74677200636755
A1Cresult ----- 83.27732248491637
metformin ----- 0.0
repaglinide ----- 0.0
nateglinide ----- 0.0
chlorpropamide ----- 0.0
glimepiride ----- 0.0
acetohexamide ----- 0.0
glipizide ----- 0.0
glyburide ----- 0.0
tolbutamide ----- 0.0
pioglitazone ----- 0.0
rosiglitazone ----- 0.0
acarbose ----- 0.0
miglitol ----- 0.0
troglitazone ----- 0.0
tolazamide ----- 0.0
examide ----- 0.0
citoglipton ----- 0.0
insulin ----- 0.0
glyburide-metformin ----- 0.0
glipizide-metformin ----- 0.0
glimepiride-pioglitazone ----- 0.0
```

Figure 23: Data preprocessing and cleaning (a)

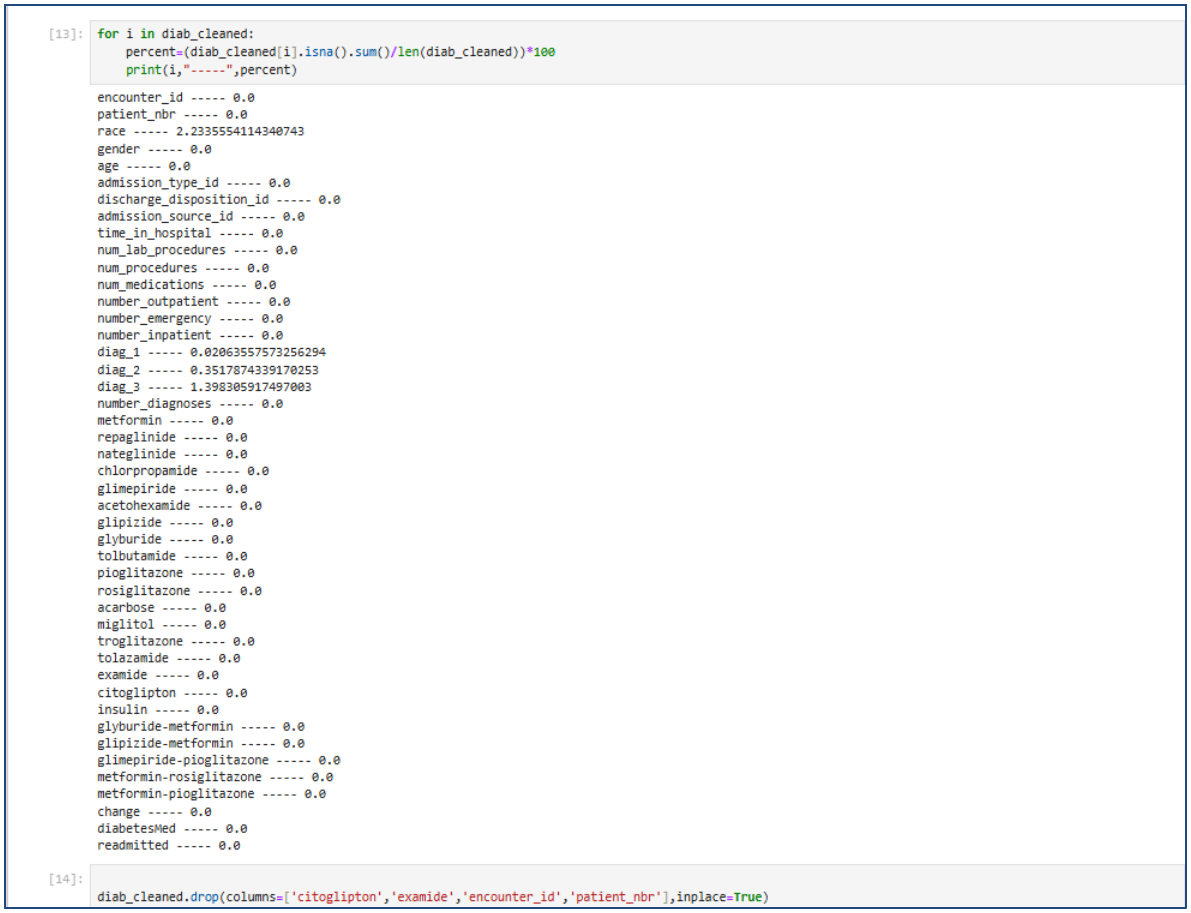


```
diag_3 ----- 1.398305917497003
number_diagnoses ----- 0.0
max_glu_serum ----- 94.74677200636755
A1Cresult ----- 83.27732248491637
metformin ----- 0.0
repaglinide ----- 0.0
nateglinide ----- 0.0
chlorpropamide ----- 0.0
glimepiride ----- 0.0
acetohexamide ----- 0.0
glipizide ----- 0.0
glyburide ----- 0.0
tolbutamide ----- 0.0
pioglitazone ----- 0.0
rosiglitazone ----- 0.0
acarbose ----- 0.0
miglitol ----- 0.0
troglitazone ----- 0.0
tolazamide ----- 0.0
examide ----- 0.0
citoglipton ----- 0.0
insulin ----- 0.0
glyburide-metformin ----- 0.0
glipizide-metformin ----- 0.0
glimepiride-pioglitazone ----- 0.0
metformin-rosiglitazone ----- 0.0
metformin-pioglitazone ----- 0.0
change ----- 0.0
diabetesMed ----- 0.0
readmitted ----- 0.0
```

As per rule of thumb all the columns with more than 10% missing values can be removed, hence we remove the columns: 1. weight 2. payer_code 3.medical_specialty 4. max_glu_serum 5.A1Cresult

```
[9]: diab_cleaned=diab.drop(['weight','payer_code','medical_specialty','max_glu_serum','A1Cresult'],axis=1)
```

Figure 24: Data preprocessing and cleaning (b)

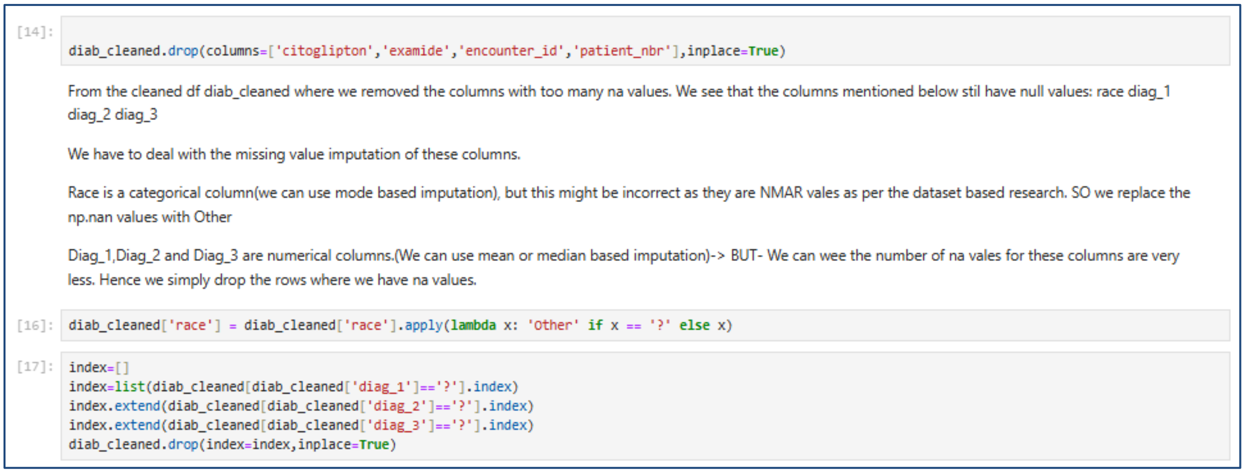


```
[13]: for i in diab_cleaned:
          percent=(diab_cleaned[i].isna().sum()/len(diab_cleaned))*100
          print(i,"-----",percent)
```

```
encounter_id ----- 0.0
patient_nbr ----- 0.0
race ----- 2.2335554114340743
gender ----- 0.0
age ----- 0.0
admission_type_id ----- 0.0
discharge_disposition_id ----- 0.0
admission_source_id ----- 0.0
time_in_hospital ----- 0.0
num_lab_procedures ----- 0.0
num_procedures ----- 0.0
num_medications ----- 0.0
number_outpatient ----- 0.0
number_emergency ----- 0.0
number_inpatient ----- 0.0
diag_1 ----- 0.02063557573256294
diag_2 ----- 0.3517874339170253
diag_3 ----- 1.398305917497003
number_diagnoses ----- 0.0
metformin ----- 0.0
repaglinide ----- 0.0
nateglinide ----- 0.0
chlorpropamide ----- 0.0
glimepiride ----- 0.0
acetohexamide ----- 0.0
glipizide ----- 0.0
glyburide ----- 0.0
tolbutamide ----- 0.0
pioglitazone ----- 0.0
rosiglitazone ----- 0.0
acarbose ----- 0.0
miglitol ----- 0.0
troglitazone ----- 0.0
tolazamide ----- 0.0
examide ----- 0.0
citoglipton ----- 0.0
insulin ----- 0.0
glyburide-metformin ----- 0.0
glipizide-metformin ----- 0.0
glimepiride-pioglitazone ----- 0.0
metformin-rosiglitazone ----- 0.0
metformin-pioglitazone ----- 0.0
change ----- 0.0
diabetesMed ----- 0.0
readmitted ----- 0.0
```

```
[14]:
      diab_cleaned.drop(columns=['citoglipton','examide','encounter_id','patient_nbr'],inplace=True)
```

Figure 25: Data preprocessing and cleaning (c)



```
[14]:
diab_cleaned.drop(columns=['citoglipton','examide','encounter_id','patient_nbr'],inplace=True)
```

From the cleaned df diab_cleaned where we removed the columns with too many na values. We see that the columns mentioned below stil have null values: race diag_1 diag_2 diag_3

We have to deal with the missing value imputation of these columns.

Race is a categorical column(we can use mode based imputation), but this might be incorrect as they are NMAR vales as per the dataset based research. SO we replace the np.nan values with Other

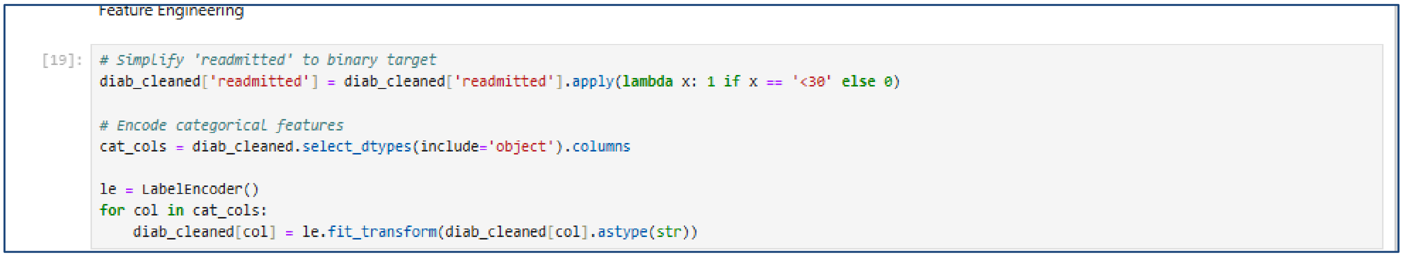
Diag_1,Diag_2 and Diag_3 are numerical columns.(We can use mean or median based imputation)-> BUT- We can wee the number of na vales for these columns are very less. Hence we simply drop the rows where we have na values.

```
[16]: diab_cleaned['race'] = diab_cleaned['race'].apply(lambda x: 'Other' if x == '?' else x)
```

```
[17]: index=[]
      index=list(diab_cleaned[diab_cleaned['diag_1']=='?'].index)
      index.extend(diab_cleaned[diab_cleaned['diag_2']=='?'].index)
      index.extend(diab_cleaned[diab_cleaned['diag_3']=='?'].index)
      diab_cleaned.drop(index=index,inplace=True)
```

Figure 26: Data preprocessing and cleaning (d)

Feature Engineering:



Feature Engineering

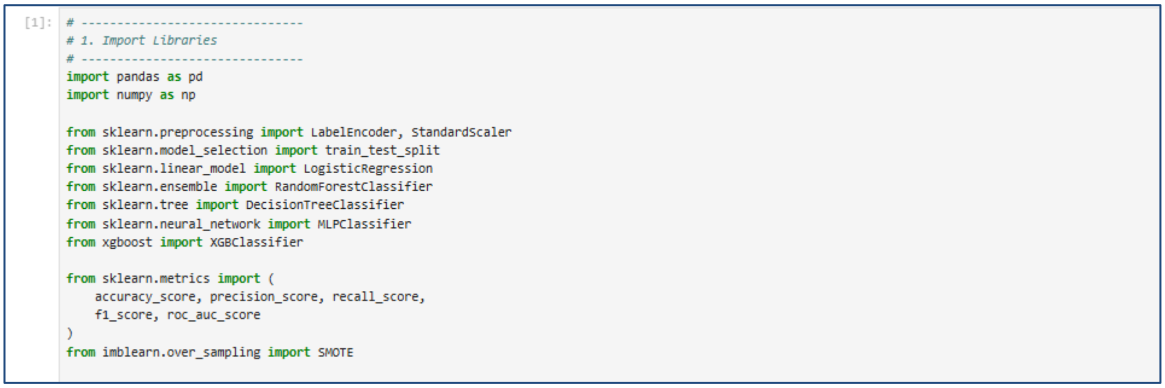
```
[19]: # Simplify 'readmitted' to binary target
      diab_cleaned['readmitted'] = diab_cleaned['readmitted'].apply(lambda x: 1 if x == '<30' else 0)

      # Encode categorical features
      cat_cols = diab_cleaned.select_dtypes(include='object').columns

      le = LabelEncoder()
      for col in cat_cols:
          diab_cleaned[col] = le.fit_transform(diab_cleaned[col].astype(str))
```

Figure 27: Feature Engineering

6. MODEL TRAINING AND COMPARISON:

Our aim as a part of this section would be to train models: Logistic Regression, Random Forest Classifier, Decision tree Classifier and Neural Network (MLP Classifier) and understand with a comparative study on which model would be more suited for the readmission risk prediction problem statement for our Data set.

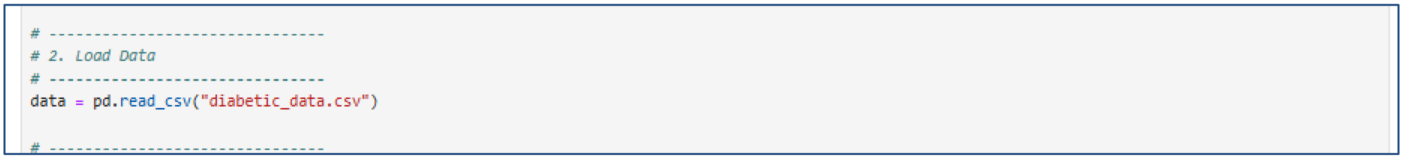
Below code imports the necessary libraries for the model training and testing.



```
[1]: # ------------------------------
     # 1. Import Libraries
     # ------------------------------
     import pandas as pd
     import numpy as np

     from sklearn.preprocessing import LabelEncoder, StandardScaler
     from sklearn.model_selection import train_test_split
     from sklearn.linear_model import LogisticRegression
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.tree import DecisionTreeClassifier
     from sklearn.neural_network import MLPClassifier
     from xgboost import XGBClassifier

     from sklearn.metrics import (
         accuracy_score, precision_score, recall_score,
         f1_score, roc_auc_score
     )
     from imblearn.over_sampling import SMOTE
```

Figure 28: Model Training libraries importing

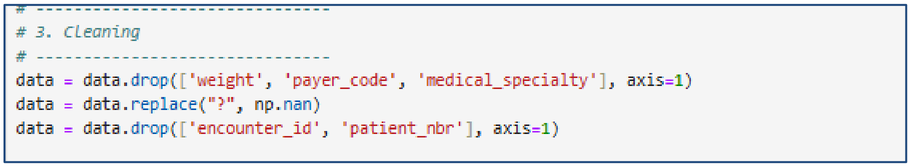
The below code is used to load the dataset:



```
# ------------------------------
# 2. Load Data
# ------------------------------
data = pd.read_csv("diabetic_data.csv")

# ------------------------------
```

Figure 29: Read CSV file

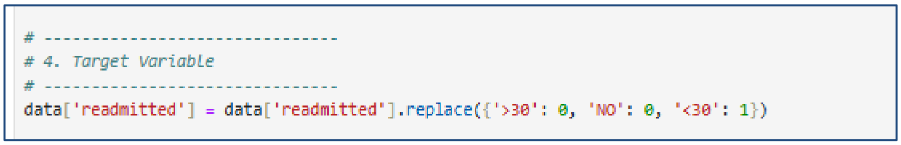
Based on our our dataset analysis which we did in previous section, we can see that certain columns can be removed to excess null values present in them and hence we are removing them. Whatever remaining "?" values there are in the remaining columns we replace with np.nan and we also drop the columns with PII data (encounter_id and patient_nbr):



```
# ------------------------------
# 3. Cleaning
# ------------------------------
data = data.drop(['weight', 'payer_code', 'medical_specialty'], axis=1)
data = data.replace("?", np.nan)
data = data.drop(['encounter_id', 'patient_nbr'], axis=1)
```

Figure 30: Data Cleaning

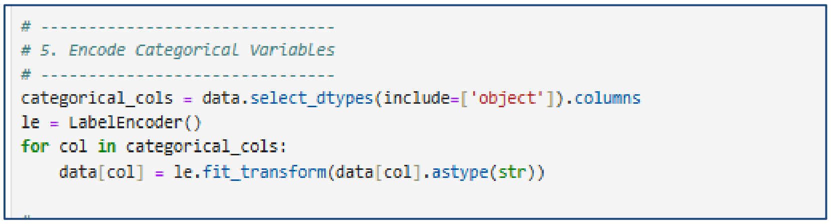
Our problem statement is to find the readmission probability of a patient within 30 days of discharge. The output/target variable – readmitted has three possible values: <30, >30 and No. We are keeping <30 as 1 and both >30 and No as 0.



```
# ------------------------------
# 4. Target Variable
# ------------------------------
data['readmitted'] = data['readmitted'].replace({'>30': 0, 'NO': 0, '<30': 1})
```

Figure 31: Target Variable Mapping

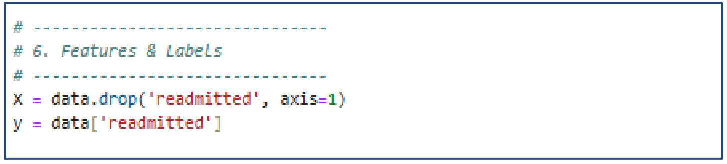
Categorical variables such as race, gender, and admission type were converted into numerical form using Label Encoding. This ensured that all features were model-compatible and allowed the algorithm to interpret categorical data correctly.



```
# ------------------------------
# 5. Encode Categorical Variables
# ------------------------------
categorical_cols = data.select_dtypes(include=['object']).columns
le = LabelEncoder()
for col in categorical_cols:
    data[col] = le.fit_transform(data[col].astype(str))
```

Figure 32: Encoding Categorical Variables

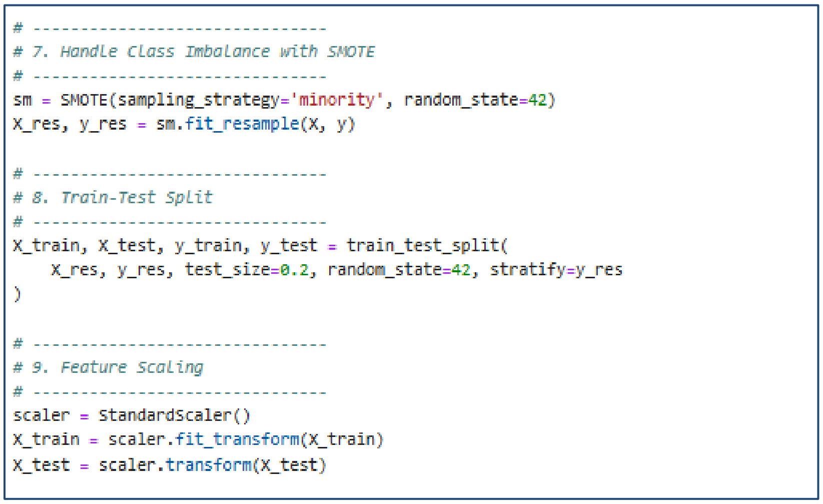
Splitting into data frames for input variable and target variables X and Y respectively.



```
# ------------------------------
# 6. Features & Labels
# ------------------------------
X = data.drop('readmitted', axis=1)
y = data['readmitted']
```

Figure 33: Feature and Labels

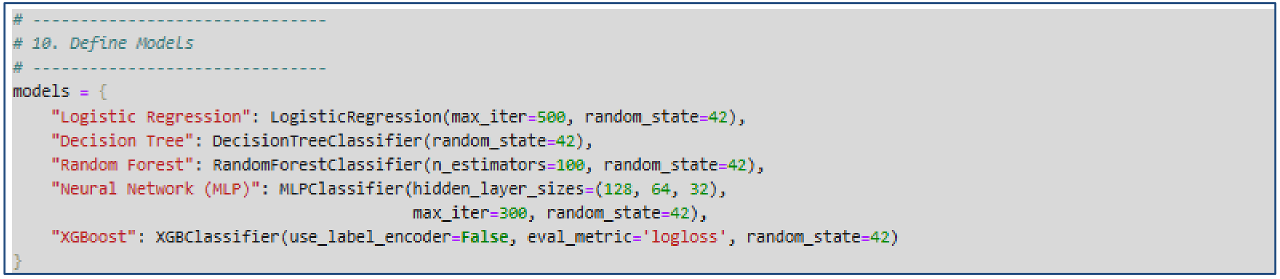
To address class imbalance, SMOTE was applied to generate synthetic minority samples. The balanced dataset was then split into training and testing sets using an 80–20 ratio with stratification. Finally, StandardScaler was applied to normalize the feature values, ensuring fair contribution of all variables during model training.



```
# ------------------------------
# 7. Handle Class Imbalance with SMOTE
# ------------------------------
sm = SMOTE(sampling_strategy='minority', random_state=42)
X_res, y_res = sm.fit_resample(X, y)

# ------------------------------
# 8. Train-Test Split
# ------------------------------
X_train, X_test, y_train, y_test = train_test_split(
    X_res, y_res, test_size=0.2, random_state=42, stratify=y_res
)

# ------------------------------
# 9. Feature Scaling
# ------------------------------
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

Figure 34: Class imbalance handling

This block defines a set of candidate models for hospital readmission prediction.



```
# ------------------------------
# 10. Define Models
# ------------------------------
models = {
    "Logistic Regression": LogisticRegression(max_iter=500, random_state=42),
    "Decision Tree": DecisionTreeClassifier(random_state=42),
    "Random Forest": RandomForestClassifier(n_estimators=100, random_state=42),
    "Neural Network (MLP)": MLPClassifier(hidden_layer_sizes=(128, 64, 32),
                                         max_iter=300, random_state=42),
    "XGBoost": XGBClassifier(use_label_encoder=False, eval_metric='logloss', random_state=42)
}
```

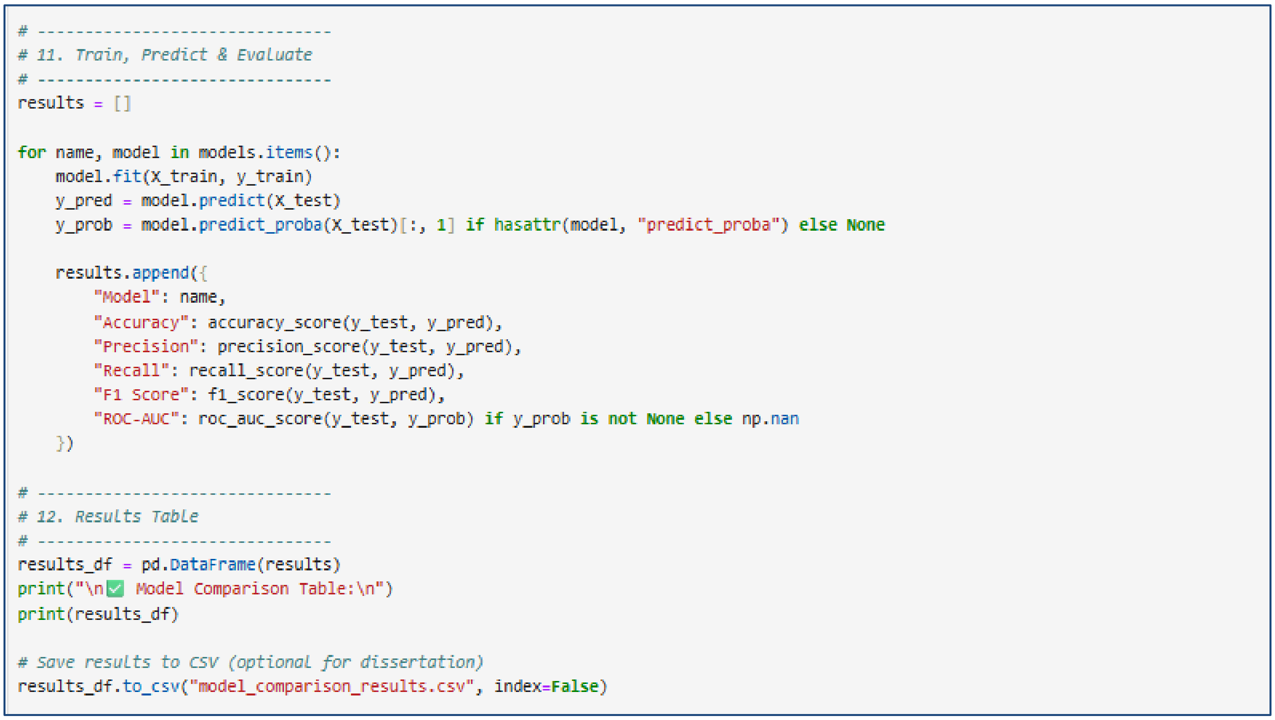
Figure 35: Model definition

- Logistic Regression
  - A baseline statistical model, useful for binary classification tasks like readmission prediction.
  - Helps interpret feature importance through coefficients.
- Decision Tree
  - A simple tree-based classifier that splits data into decision rules.
  - Easy to interpret but may overfit.
- Random Forest
  - An ensemble of multiple decision trees trained on random subsets of data and features.
  - Provides robustness and higher accuracy while reducing overfitting.
- Neural Network (MLP – Multi-Layer Perceptron)
  - A deep learning approach with hidden layers (128 → 64 → 32).
  - Captures complex nonlinear relationships in the data.
- XGBoost (Extreme Gradient Boosting)
  - A powerful gradient boosting algorithm that builds trees sequentially.
  - Known for high accuracy and efficiency in structured/tabular data.
  - Widely used in healthcare predictive modeling competitions.

Why this is important

- Each algorithm has strengths:
  - Logistic Regression → interpretability.
  - Tree-based models → handle categorical/nonlinear data well.
  - Neural Networks → capture complex interactions.
  - XGBoost → high predictive performance.
- This comparative study helps identify the most suitable model for real-world hospital readmission prediction.

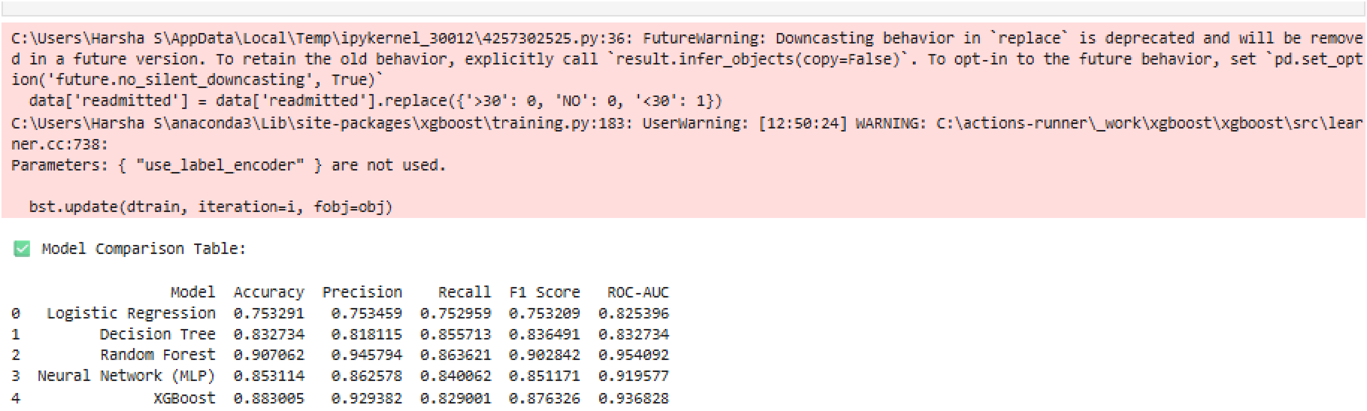
Each model was trained on the training dataset and evaluated on the test set. Multiple metrics including Accuracy, Precision, Recall, F1 Score, and ROC-AUC were computed to provide a holistic performance comparison. The results were consolidated into a comparison table and exported for reporting. This enabled the identification of the most suitable model for predicting hospital readmissions among diabetic patients.



```
# ------------------------------
# 11. Train, Predict & Evaluate
# ------------------------------
results = []

for name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    y_prob = model.predict_proba(X_test)[:, 1] if hasattr(model, "predict_proba") else None

    results.append({
        "Model": name,
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred),
        "Recall": recall_score(y_test, y_pred),
        "F1 Score": f1_score(y_test, y_pred),
        "ROC-AUC": roc_auc_score(y_test, y_prob) if y_prob is not None else np.nan
    })

# ------------------------------
# 12. Results Table
# ------------------------------
results_df = pd.DataFrame(results)
print("\n✅ Model Comparison Table:\n")
print(results_df)

# Save results to CSV (optional for dissertation)
results_df.to_csv("model_comparison_results.csv", index=False)
```

Figure 36: Train, predict and evaluate model

RESULT:



```
C:\Users\Harsha S\AppData\Local\Temp\ipykernel_30012\4257302525.py:36: FutureWarning: Downcasting behavior in `replace` is deprecated and will be remove
d in a future version. To retain the old behavior, explicitly call `result.infer_objects(copy=False)`. To opt-in to the future behavior, set `pd.set_opt
ion('future.no_silent_downcasting', True)`
  data['readmitted'] = data['readmitted'].replace({'>30': 0, 'NO': 0, '<30': 1})
C:\Users\Harsha S\anaconda3\Lib\site-packages\xgboost\training.py:183: UserWarning: [12:50:24] WARNING: C:\actions-runner\_work\xgboost\xgboost\src\lear
ner.cc:738:
Parameters: { "use_label_encoder" } are not used.

  bst.update(dtrain, iteration=i, fobj=obj)

✅ Model Comparison Table:

                  Model  Accuracy  Precision    Recall  F1 Score   ROC-AUC
0   Logistic Regression  0.753291   0.753459  0.752959  0.753209  0.825396
1         Decision Tree  0.832734   0.818115  0.855713  0.836491  0.832734
2         Random Forest  0.907062   0.945794  0.863621  0.902842  0.954092
3  Neural Network (MLP)  0.853114   0.862578  0.840062  0.851171  0.919577
4               XGBoost  0.883005   0.929382  0.829001  0.876326  0.936828
```

Figure 37: Model Comparison Results

<u>Explanation of Each Metric:</u>

- Accuracy → overall correctness of predictions.
- Precision → of the patients predicted as readmitted, how many were actually readmitted (reduces false alarms).
- Recall (Sensitivity) → of the patients who were actually readmitted, how many were correctly predicted (important to avoid missing high-risk cases).
- F1 Score → balance between precision and recall.
- ROC-AUC → model's ability to distinguish between readmitted vs. not readmitted across thresholds (higher = better discrimination).

Model-wise Interpretation:

- Logistic Regression
  Accuracy: 75.3%
  Balanced precision (0.753) and recall (0.753).
  ROC-AUC: 0.82, showing decent separability.
  Simple, interpretable baseline model.

- Decision Tree
  Accuracy: 83.3%, higher than Logistic Regression.
  Recall: 0.856, meaning it catches more actual readmissions.
  Precision slightly lower (0.818), meaning more false positives compared to RF/XGBoost.
  Useful for interpretability but can overfit.

- Random Forest
  Highest Accuracy: 90.7%
  Precision: 0.946 → very few false positives.
  Recall: 0.864 → strong ability to detect readmissions.
  F1: 0.903, best overall balance.
  ROC-AUC: 0.95, strongest separability.
  Best-performing model overall.

- Neural Network (MLP)
  Accuracy: 85.3%
  Precision: 0.863, Recall: 0.840, F1: 0.851.
  ROC-AUC: 0.92, second best after Random Forest/XGBoost.
  Good performance but requires more computation and less interpretable.

- XGBoost
  Accuracy: 88.3%
  Precision: 0.929, slightly less than Random Forest but very strong.
  Recall: 0.829, a bit lower (misses some readmissions compared to RF/Tree).
  ROC-AUC: 0.937, very high discrimination power.
  Strong candidate, slightly more efficient than Random Forest for large datasets.

Final Takeaway:

- Best Model → Random Forest, due to its highest overall scores across all metrics, especially F1 and ROC-AUC.
- Close Second → XGBoost, with excellent precision and ROC-AUC, slightly weaker on recall.
- Baseline → Logistic Regression, interpretable but less accurate.

Our Choice:

Although Random Forest achieved the best predictive performance, we opt for XGBoost as the final model due to its stronger compatibility with SHAP explainability, computational efficiency, and consistent interpretability of features. This makes it better suited for real-world deployment in a clinical decision-support setting, where transparency and trust are essential.

7. FLOWCHART FOR EXPLANATION OF THE WORKFLOW IMPLEMENTED:

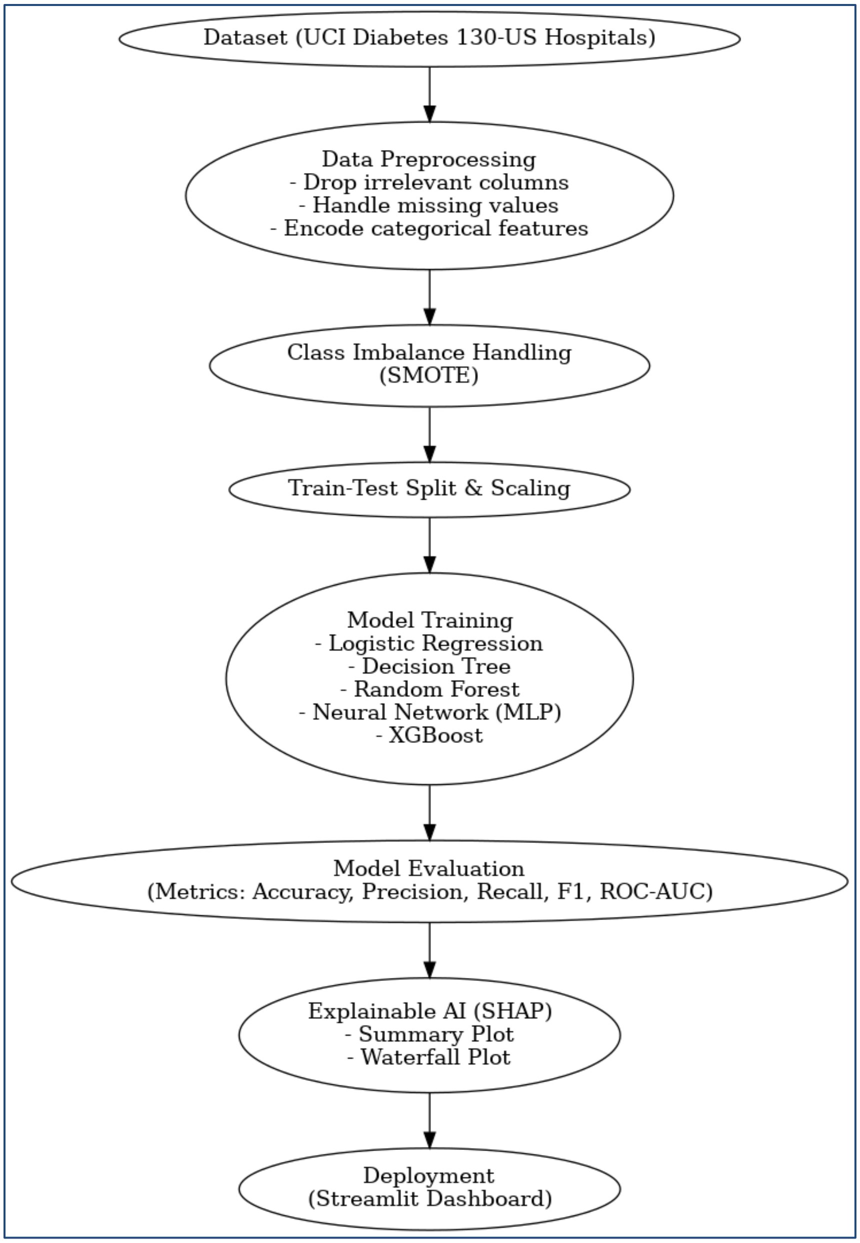
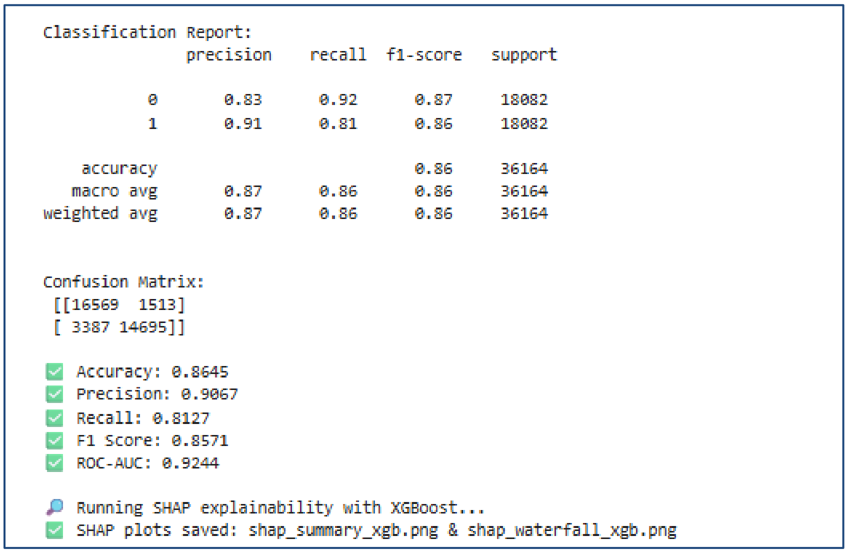


Figure 38: Workflow Flowchart of Solution

This is the end-to-end flow diagram of the workflow which we have implemented, till now steps up to model evaluation have been covered.

Now, the next step is to take a deeper dive into the results of the finalised model- XGBOOST.

This is the result which we have obtained after training and testing the model:



```
Classification Report:
               precision    recall  f1-score   support

           0       0.83      0.92      0.87     18082
           1       0.91      0.81      0.86     18082

    accuracy                           0.86     36164
   macro avg       0.87      0.86      0.86     36164
weighted avg       0.87      0.86      0.86     36164


Confusion Matrix:
 [[16569  1513]
 [ 3387 14695]]

✅ Accuracy: 0.8645
✅ Precision: 0.9067
✅ Recall: 0.8127
✅ F1 Score: 0.8571
✅ ROC-AUC: 0.9244

🔎 Running SHAP explainability with XGBoost...
✅ SHAP plots saved: shap_summary_xgb.png & shap_waterfall_xgb.png
```

Figure 39: Classification Report and results for XGBoost

The XGBoost model achieved an overall accuracy of **86.5%** and ROC-AUC **0.924**, indicating strong discrimination between readmitted and non-readmitted patients.

For the readmission class the model attained **precision = 0.91** and **recall = 0.81** (F1 = 0.86), which means that 91% of predicted readmissions were correct while the model detected 81% of actual readmissions (≈19% false negatives).

The confusion matrix (TN=16,569; FP=1,513; FN=3,387; TP=14,695) shows a favorable balance between false alarms and missed cases.

8. SHAP SUMMARY PLOT OF 200 RANDOM SAMPLES

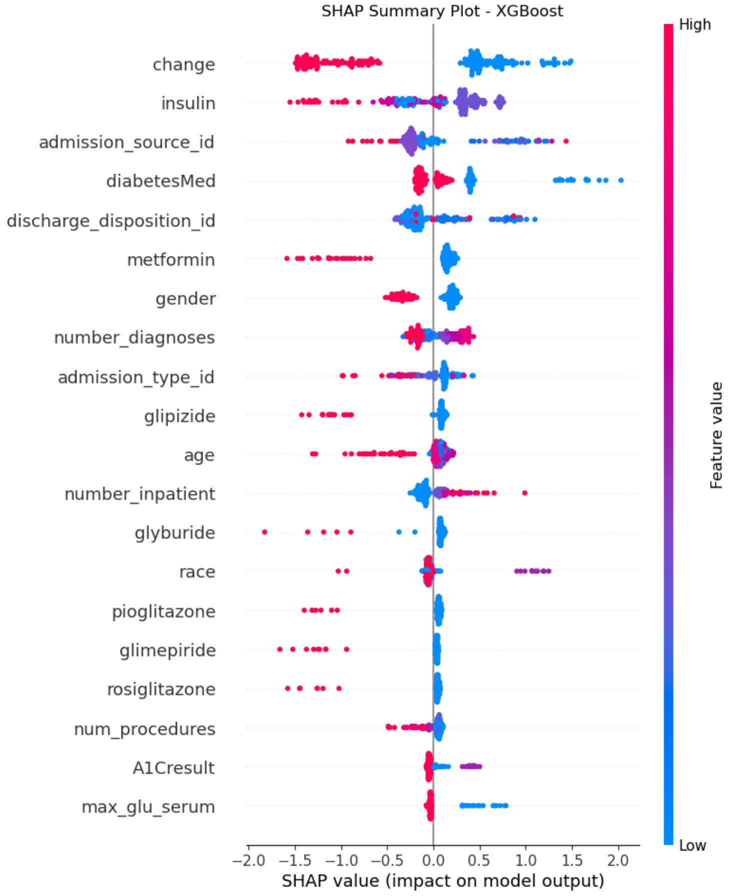


Figure 40: SHAP Summary plot

The basis of the above SHAP Summary Plot is as follows:

- Y-axis (features): The most important features are listed from top to bottom, ranked by their average impact on the model's output. Here, features like change, insulin, and admission_source_id are the most influential in predicting readmission.
- X-axis (SHAP value): Measures the direction and magnitude of impact on the prediction.
    - Positive SHAP value → pushes prediction towards higher probability of readmission.
    - Negative SHAP value → pushes prediction towards lower probability (not readmitted).
- Colors: Encode the feature value (red = high, blue = low). This shows *how* the value of a feature influences the prediction.

<u>2. Key feature-level insights</u>

- Change (top feature):
  Patients whose medications were changed (change=Yes, red) strongly push predictions toward readmission (positive SHAP). If no change (change=No, blue), the contribution is negative (lower readmission risk).
- Insulin:
  High/active insulin usage (red) increases predicted readmission risk. Low/no insulin (blue) reduces risk. This aligns clinically — insulin therapy often indicates more severe diabetes management needs.

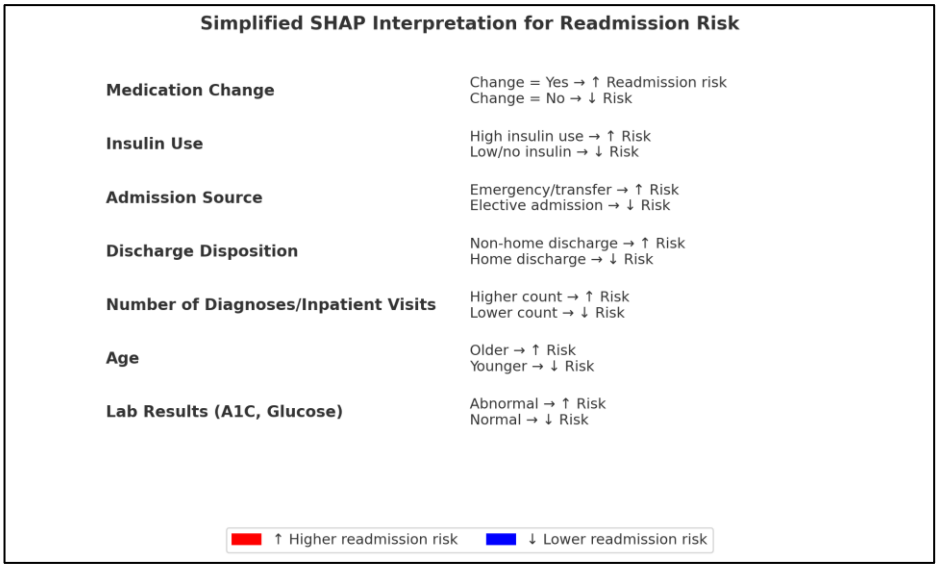
- Admission source ID:
Certain admission sources (e.g., emergency referrals, transfers) increase readmission risk. Others (e.g., elective admissions) reduce it.
- DiabetesMed:
Being on diabetes medication (red) increases risk compared to those not on medication, since it may indicate chronic/severe cases.
- Discharge disposition ID:
Where a patient is discharged to (e.g., home, nursing facility, rehab) heavily influences readmission. Non-home discharges often increase risk.
- Metformin, glipizide, glyburide, pioglitazone, etc. (drug variables):
These drugs individually contribute smaller but non-trivial shifts. For example, high values (red, indicating "drug prescribed") sometimes increase risk, reflecting clinical severity.
- Number of diagnoses & inpatient visits:
More diagnoses/inpatient visits (red = high values) push the model toward predicting readmission. Fewer = safer.
- Age:
Higher age (red) slightly increases risk compared to younger patients.
- Race & Gender:
Appear with modest influence. Important to monitor here for potential bias, as SHAP reveals model reliance.
- Lab results (A1C result, max_glu_serum):
Abnormal/"high" results (red) slightly increase readmission risk.

3. Clinical interpretation

- The model is most sensitive to medication changes, insulin use, admission context, and discharge status.
- These factors make sense: medication changes and discharge settings are closely tied to patient stability and follow-up needs.
- Drug-specific features and lab results play smaller but additive roles in prediction.
- The model's behavior largely aligns with clinical intuition — a key justification for trusting the XGBoost + SHAP pipeline.

4. Overall Conclusion

The SHAP summary plot ranks features by their contribution to predicting readmission. The most influential factors included **medication changes**, **insulin therapy**, and **admission source**, with high values of these features pushing the model toward predicting readmission. Discharge disposition, diabetes medications, and the number of diagnoses also played major roles. Clinical variables such as **age**, **lab results (A1C, glucose serum)**, and the **use of specific diabetes drugs** contributed more modestly. The directionality of SHAP values aligns with clinical expectations — for instance, patients with frequent medication changes or multiple inpatient visits are at higher risk of readmission — thereby supporting the model's interpretability and trustworthiness.



Simplified SHAP Interpretation for Readmission Risk

Medication Change	Change = Yes → ↑ Readmission risk
	Change = No → ↓ Risk

Insulin Use	High insulin use → ↑ Risk Low/no insulin → ↓ Risk
Admission Source	Emergency/transfer → ↑ Risk Elective admission → ↓ Risk
Discharge Disposition	Non-home discharge → ↑ Risk Home discharge → ↓ Risk
Number of Diagnoses/Inpatient Visits	Higher count → ↑ Risk Lower count → ↓ Risk
Age	Older → ↑ Risk Younger → ↓ Risk
Lab Results (A1C, Glucose)	Abnormal → ↑ Risk Normal → ↓ Risk

🟥 ↑ Higher readmission risk 🟦 ↓ Lower readmission risk

Figure 41: SHAP Summary plot simplified interpretation

9. SHAP WATERFALL PLOT FOR A TEST SAMPLE

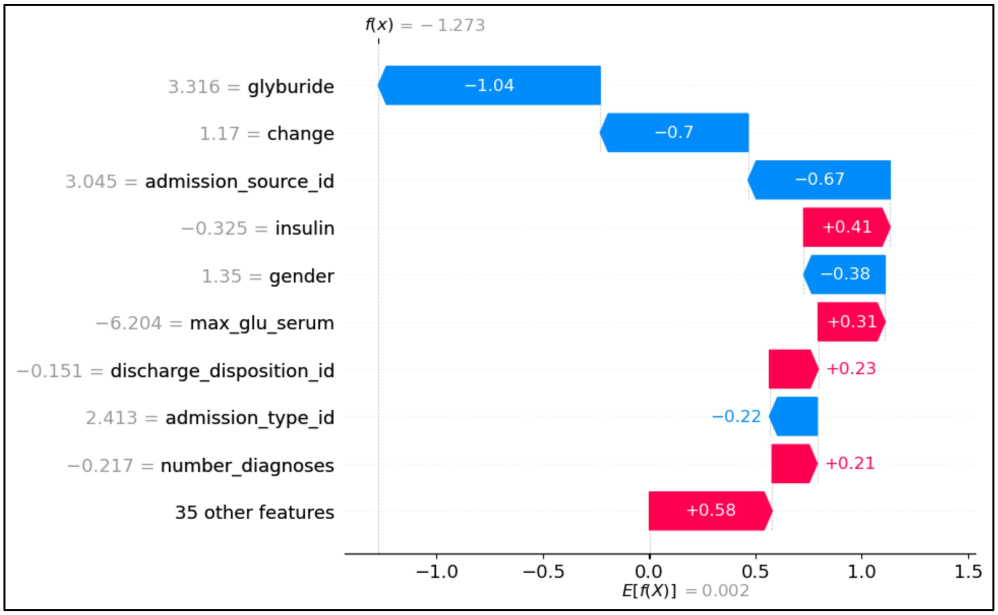


Figure 42: SHAP Waterfall Plot

Explanation of how a waterfall plot for SHAP values works is as follows:

- x-axis: contribution of each feature to the prediction.
  - Negative values (blue) push the prediction toward “no readmission”.
  - Positive values (red) push the prediction toward “readmission”.
- Center line (E[f(x)] = 0.002): the base value (average model prediction across all patients).
- Right end (f(x) = –1.273): the final prediction for this patient, after adding contributions of all features.

Key Features in this Example:

1. Glyburide (–1.04, blue)
   - Strongly decreases readmission risk for this patient.
   - Suggests the patient’s usage of glyburide lowered the predicted probability.
2. Change in Medication (–0.70, blue)
   - No/low change in medication reduced risk.
3. Admission Source (–0.67, blue)
   - The type of admission source contributed to lowering risk.
4. Insulin (+0.41, red)
   - Presence of insulin therapy increased readmission risk.
5. Gender (+0.31, red)
   - Patient’s gender slightly pushed risk upward.
6. Max Glucose Serum (–0.23, blue)
   - Normal or low glucose serum reduced risk.
7. Other Features (+0.58, red)
   - Collectively, other small factors increased risk a bit.

<u>Interpretation:</u>

- The overall prediction is negative (–1.273) → the model predicts this patient is unlikely to be readmitted within 30 days.

- The biggest protective factor is glyburide usage, followed by medication stability and admission source.
- The main risk factors for this patient are insulin therapy, gender, and other combined features.

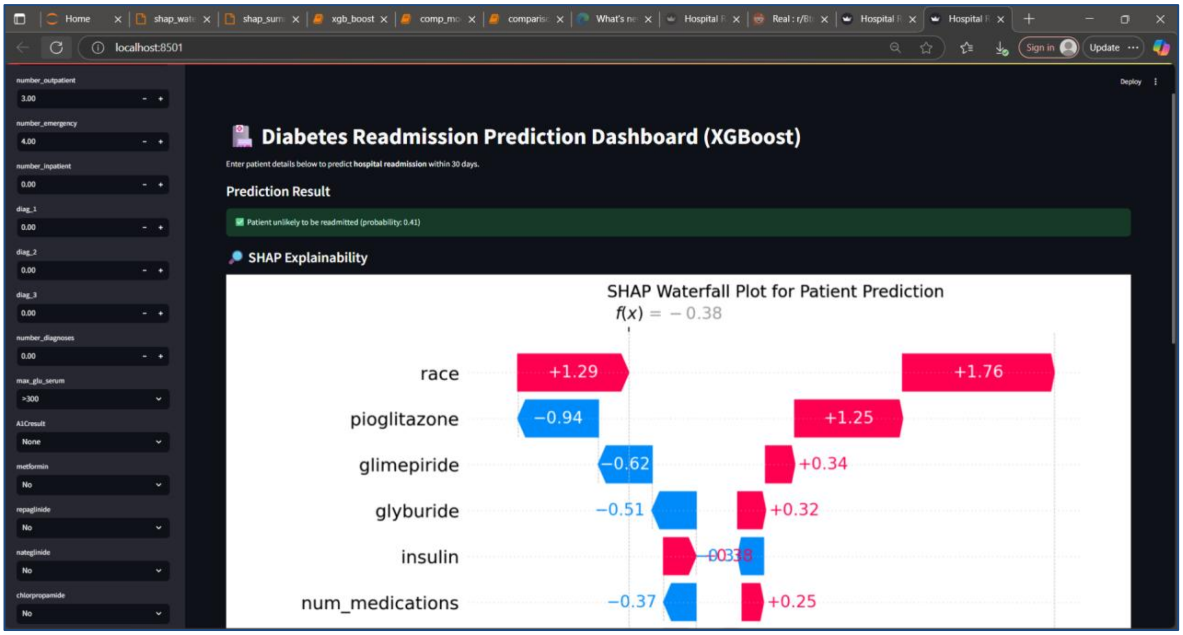
Overall Inference:

The SHAP waterfall plot explains individual patient predictions by showing how each feature pushes the risk of readmission up or down. For this patient, glyburide usage and stable medication significantly reduced the risk, while insulin use and demographic factors slightly increased it. This kind of patient-level interpretability makes XGBoost explainable for clinicians.

10. DASHBOARD IMPLEMENTATION AND OUTPUT

The trained XGBoost model was deployed using **Streamlit** as an interactive dashboard for clinicians. The system allows users to enter patient details (demographics, diagnoses, medications, and lab results) through dropdowns and numeric inputs. These inputs are preprocessed using stored **LabelEncoders** and a **StandardScaler**, after which the model predicts the probability of hospital readmission within 30 days.

The dashboard displays the prediction result (high-risk or low-risk) along with a **SHAP waterfall plot** to explain feature-level contributions for each individual patient. This ensures transparency and builds clinician trust. The application can be run locally or hosted on cloud platforms for hospital-wide use, making it both practical and scalable.



number_outpatient: 3.00

number_emergency: 4.00

number_inpatient: 0.00

diag_1: 0.00

diag_2: 0.00

diag_3: 0.00

number_diagnoses: 0.00

max_glu_serum: >300

A1Cresult: None

metformin: No

repaglinide: No

nateglinide: No

chlorpropamide: No

Diabetes Readmission Prediction Dashboard (XGBoost)

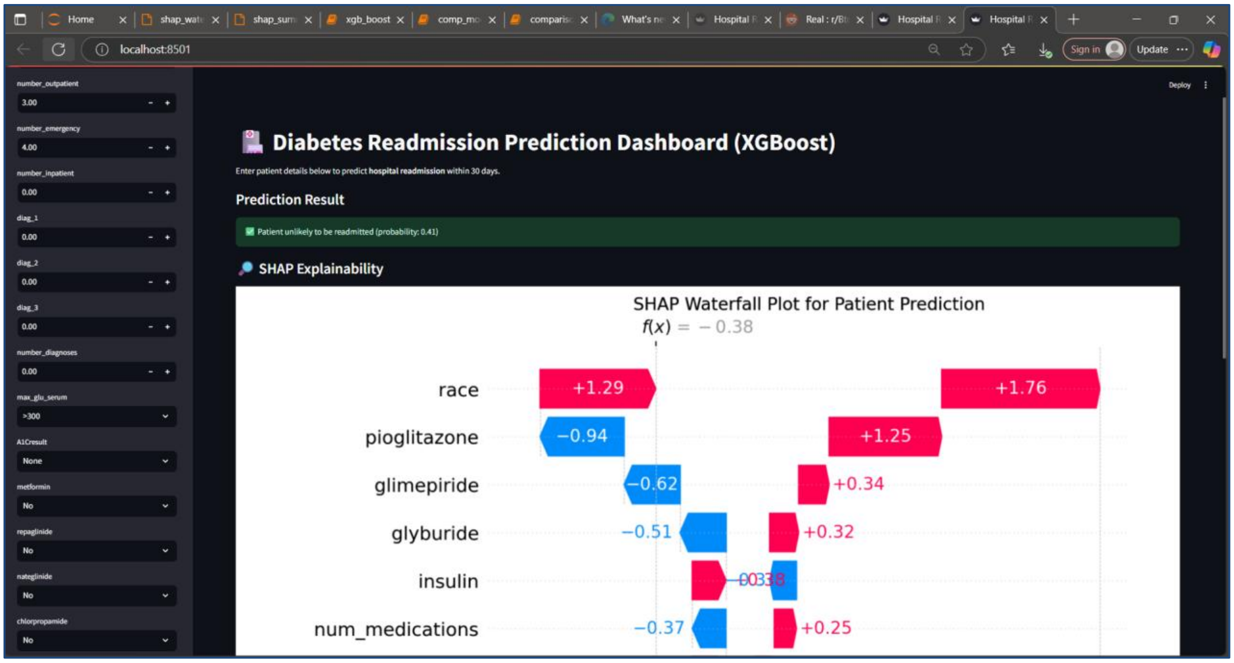
Enter patient details below to predict **hospital readmission** within 30 days.

Prediction Result

✓ Patient unlikely to be readmitted (probability: 0.41)

SHAP Explainability

Figure 43: Dashboard showing unlikely pediction



number_outpatient: 3.00

number_emergency: 4.00

number_inpatient: 0.00

diag_1: 0.00

diag_2: 0.00

diag_3: 0.00

number_diagnoses: 0.00

max_glu_serum: >300

A1Cresult: None

metformin: No

repaglinide: No

nateglinide: No

chlorpropamide: No

Diabetes Readmission Prediction Dashboard (XGBoost)

Enter patient details below to predict **hospital readmission** within 30 days.

Prediction Result

✓ Patient unlikely to be readmitted (probability: 0.41)

SHAP Explainability

Figure 44: Dashboard showing readmission probability

11. CONCLUSION:

From the dissertation project implemented, we have identified the appropriate dataset which is an Open source dataset from UCI and performed exploratory data analysis. Based on the data analysis, we could observe that the data set is imbalanced with a majority class being that of the non-readmitted, thus the data set is biased. To overcome this we are making use of SMOTE to handle data imbalances.

This dissertation presented an AI-powered predictive modeling framework for hospital readmission among diabetic patients, with a focus on explainability and clinical usability. The study systematically compared multiple machine learning models — Logistic Regression, Decision Tree, Random Forest, Neural Network (MLP), and XGBoost — after rigorous data preprocessing, SMOTE-based class balancing, and feature scaling.

Model Performance

The comparative evaluation (Table 1) revealed that ensemble methods outperformed traditional models:

Model	Accuracy	Precision	Recall	F1 Score	ROC-AUC
Logistic Regression	0.753	0.753	0.753	0.753	0.825
Decision Tree	0.833	0.818	0.856	0.836	0.833
Random Forest	0.907	0.946	0.864	0.903	0.954
Neural Network (MLP)	0.853	0.863	0.840	0.851	0.920
XGBoost	0.883	0.929	0.829	0.876	0.937

While Random Forest achieved the highest accuracy and ROC-AUC, XGBoost was chosen as the final model due to its compatibility with SHAP explainability, computational efficiency, and interpretability, which are critical in clinical applications.

Explainability with SHAP

A major contribution of this work is the integration of SHAP (SHapley Additive exPlanations) to make model decisions transparent.

- SHAP Summary Plot highlighted that medication change, insulin therapy, admission source, discharge disposition, and diabetes medications were the top contributors to readmission risk.
- SHAP Waterfall Plots provided patient-level interpretability by showing how each feature increased or decreased the probability of readmission.

This aligns with clinical reasoning — for instance, frequent medication changes or insulin dependency often indicate unstable conditions, justifying higher readmission risk.

Dashboard Deployment

To bridge the gap between research and clinical usability, the trained XGBoost model was deployed using a Streamlit-based dashboard.

- Clinicians can input patient details (categorical & numerical) via dropdowns and fields.
- The system outputs the probability of 30-day readmission.
- A SHAP Waterfall Plot is displayed alongside predictions, ensuring transparent and trustworthy decision support.

This dashboard makes the solution scalable, interactive, and usable in real-world hospital settings.
Final Remarks
This dissertation demonstrates that AI combined with Explainable AI (XAI) can significantly improve predictive modeling in healthcare while maintaining clinical trust, transparency, and ethical responsibility.

Research Contribution:

1. Comparative evaluation of five ML models for diabetic readmission prediction.
2. Integration of SHAP for both global and local interpretability.
3. Deployment of a clinically meaningful, explainable dashboard for real-time use.

Impact:

- Improved early identification of high-risk patients.
- Potential reduction in unnecessary hospitalizations.
- Enhanced collaboration between clinicians and AI systems.

In conclusion, the project successfully balances predictive performance with interpretability and usability, offering a practical AI-powered decision support tool for reducing diabetic patient readmissions.

12. FUTURE WORK:

While this dissertation demonstrates the effectiveness of AI-powered hospital readmission prediction with Explainable AI, there remain several avenues for further improvement and exploration.

- **Advanced Deep Learning Models:** Future studies could investigate the use of Recurrent Neural Networks (RNNs) or Transformer-based architectures to capture temporal patterns in-patient admissions and lab results.
- **Federated Learning:** Implementing federated learning can enable model training across multiple hospitals without sharing sensitive patient data, thereby enhancing both privacy and generalizability.
- **Integration of Additional Data Sources**: Incorporating unstructured data such as physician notes, imaging data, and real-time wearable device data may further improve predictive performance.
- **Bias and Fairness Analysis**: A deeper study on model fairness with respect to race, gender, and socioeconomic factors could help ensure equitable healthcare predictions.
- **Clinical Trials and Validation:** Deploying the dashboard in a real hospital setting and conducting prospective validation with clinicians will be essential to test real-world usability and impact.

These directions aim to build upon the current work by improving accuracy, interpretability, fairness, and deployment readiness, ultimately supporting better clinical decision-making and reducing avoidable readmissions

13. REFERENCES

- Data set Citation-
  J. Clore, K. Cios, J. DeShazo, and B. Strack. "Diabetes 130-US Hospitals for Years 1999-2008," UCI Machine Learning Repository, 2014. [Online]. Available: https://doi.org/10.24432/C5230J.

- Predicting 30-Day Readmission Rates for Diabetic Patients Using Machine Learning Models
  Citation: Li, C. (2024). "Machine learning-based readmission risk prediction for diabetic patients." *Applied and Computational Engineering*, 46, 45-59.
  https://www.ewadirect.com/proceedings/ace/article/view/10780
  Keywords: Diabetes, Hospital Readmission, Machine Learning, XGBoost, Logistic Regression, GBDT, Decision Tree, Random Forest, Deep Neural Network.
  Year: 2024
  Algorithm / Technique / Model / Method Used: XGBoost, Logistic Regression, Gradient Boosting Decision Tree (GBDT), Decision Tree, Random Forest, Deep Neural Network.

- Machine Learning-Based Predictions of Mortality and Readmission in Type 2 Diabetes Patients in the ICU
  Citation: Author(s). (2024). "Machine Learning-Based Predictions of Mortality and Readmission in Type 2 Diabetes Patients in the ICU." *Applied Sciences*, 14(18), 8443.
  https://www.mdpi.com/2076-3417/14/18/8443
  Keywords: Type 2 Diabetes, ICU, Hospital Readmission, Machine Learning, Electronic Health Records.
  Year: 2024
  Algorithm / Technique / Model / Method Used: Bagging, AdaBoost, Gaussian Naïve Bayes, Logistic Regression, Multilayer Perceptron (MLP), Support Vector Classifier (SVC).

14. CHECKLIST FOR THE FINAL REPORT

SI. No.	Checklist Item	Yes/No
1	Is the Cover page in proper format?	Yes
2	Is the Title page in proper format?	Yes
3	Is the Certificate from the Supervisor in proper format? Has it been signed?	Yes
4	Is Abstract included in the Report? Is it properly written?	Yes
5	Does the Table of Contents' page include chapter page numbers?	Yes
6	Is Introduction included in the report? Is it properly written?	Yes
7	Are the Pages numbered properly?	Yes
8	Are the Figures numbered properly?	Yes
9	Are the Tables numbered properly?	Yes
10	Are the Captions for the Figures and Tables proper?	Yes
11	Are the Appendices numbered?	Yes
12	Does the Report have Conclusions/ Recommendations of the work?	Yes
13	Are References/ Bibliography given in the Report?	Yes
14	Have the References been cited in the Report?	Yes
15	Is the citation of References/ Bibliography in proper format?	Yes