**Readmitted Prediction Capstone**

***Abstract***—In this machine learning project, I apply machine leaning concepts with hospital data. The data is from UCI Machine Learning Repository, aiming to use different feature of the patients to predict whether they will readmitted or not. There are a total of 49 features used in this task for the prediction.

I.Introduction

To estimate the readmitted, this report will recognize the most relevant factors at first and then use the selected factors to build different models: Logistic regression, Random Forest, KNN, SVC, naive bayes, Decision Tree and xgboost. Then I will use the accuracy of different model to evaluate different models and make analysis. Finally, I will print out the best predicted value as the final result.

II.Data Description

The task is to estimate the readmitted based on the following features:

1)Gender: the gender of the patient.

2)Age: the age of the patient.

3)Weight: the weight of the patient.

4)Admissiion\_type\_id, discharge\_disposition\_id, admission\_source\_id, payer\_code, num\_lab\_reproduce, num\_procedures, num\_medications, number\_outpatient, number\_emergency, number\_inpatient, number\_diagnoses: id and numbers related with the intersection

5)diag\_1, diag\_2, diag\_3: three kind of diagnosis.

6)max\_glu\_serum, A1Cresult, metformin, repaglinide, nateglinide, chlorpropamide, glimepiride, acetohexamide, glipizide, glyburide, tolbutamide, pioglitazone, rosiglitazone, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, insulin, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metformin-rosiglitazone, metformin-pioglitazone: different health index of the patient

7)change: whether patient changes his or her medicine or not.

8)diabetesMed: whether the patient take the diabetes medicine or not

9)readmitted: whether and when the patient will readmitted

Before the analysis of data, I check the duplicate and missing value,encode the name of diag1, diag2, and diag3 into binomial label, encoding the name and usage of the medicines into numerical value. And also I dropped following columns: encounter\_id, patient\_nbr, payer\_code, number\_outpatient, number\_inpatient, number\_emergency, num\_lab\_procedures, num\_procedures, num\_medications which I believe is not so relavent with the prediction of readmitted to the patient.

After that I draw the distribution of each columns, columns of max\_glu\_serum, repaglinide, nateglinied, chlorproamide, acetohexamide, race, tolbutamide, acarbose, miglitol, troglitazone, tolazamide, examide, citoglipton, glyburide-metformin, glipizide-metformin, glimepiride-pioglitazone, metfomin-rosiglitazone, metformin-pioglitazone has only one class of value or extremely imbalanced value, which may not be responsible for the different sort of target columns.

Another kind of columns dropped later is the columns with the same distribution in all the different data. Columns of glyburide and change are dropped because of this. Below are their distribution plot group by classess.

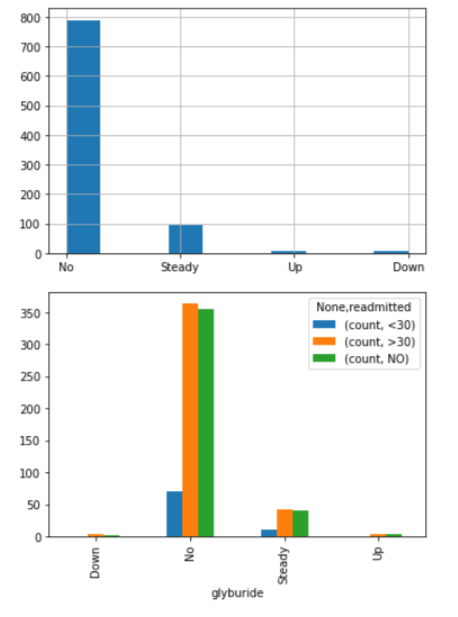
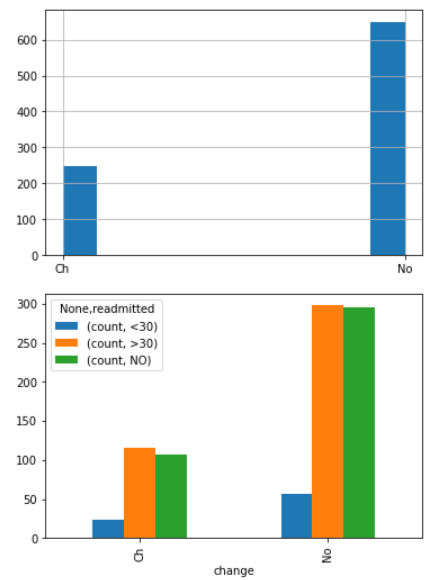
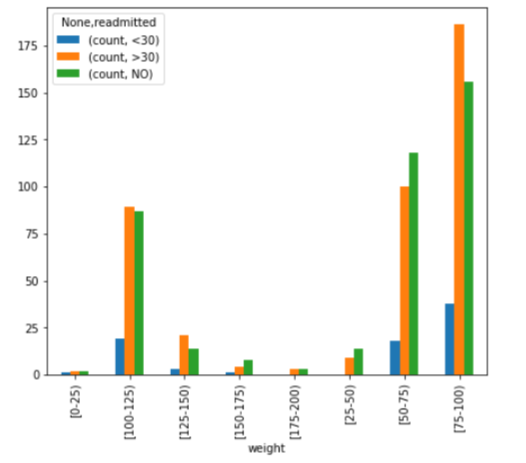
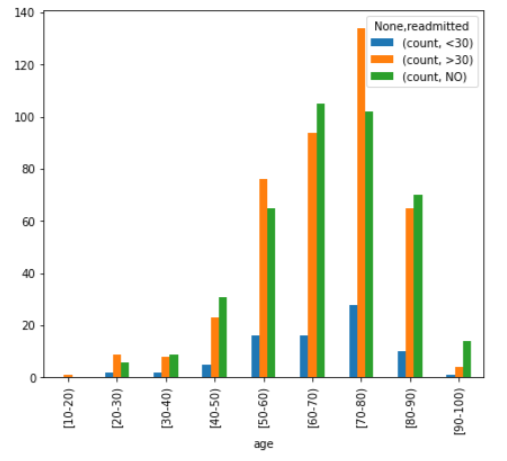
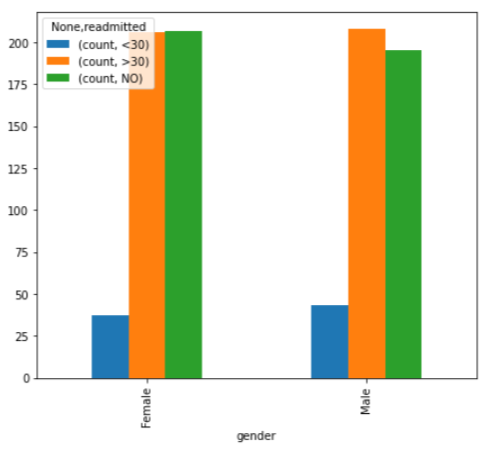
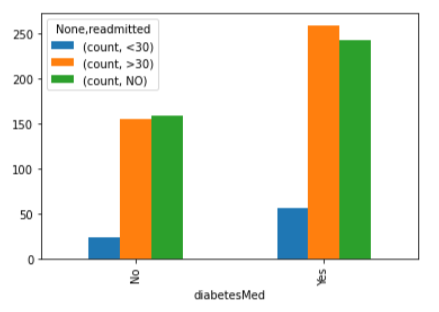
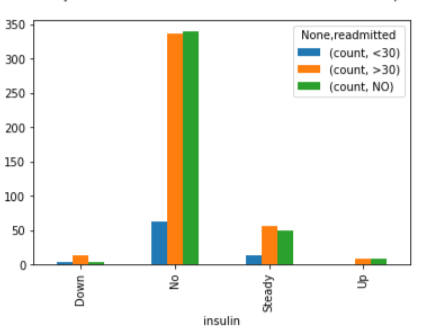
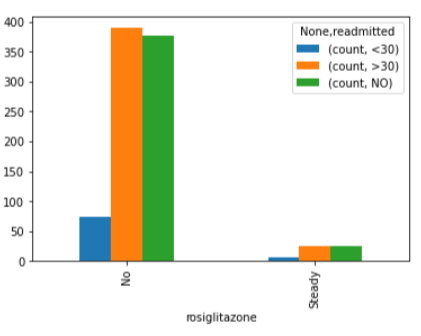
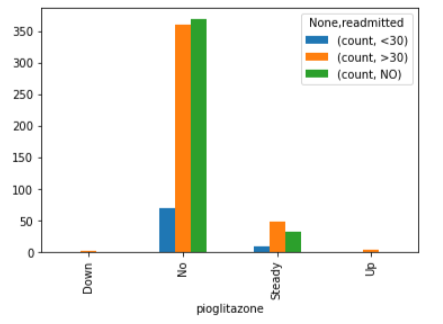
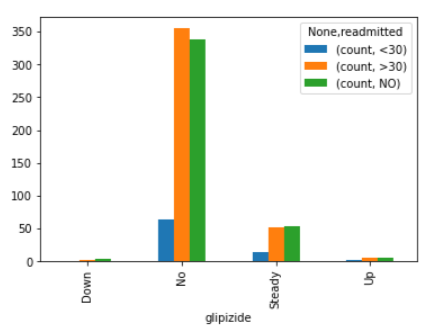
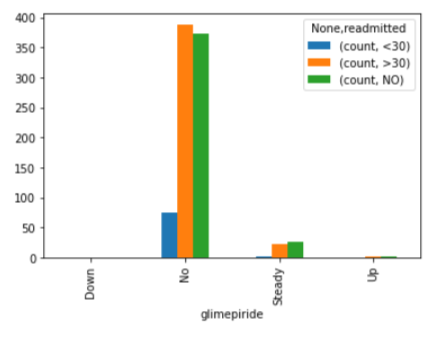
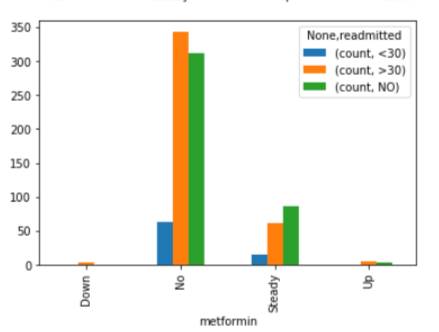
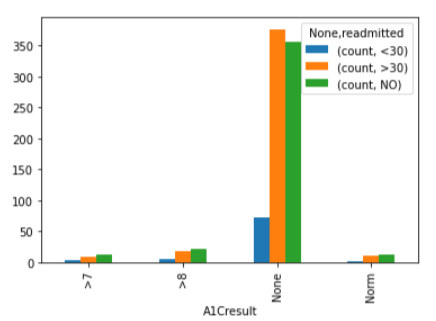
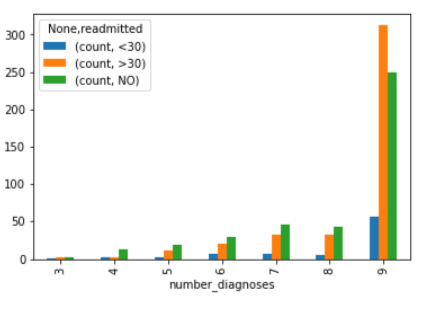
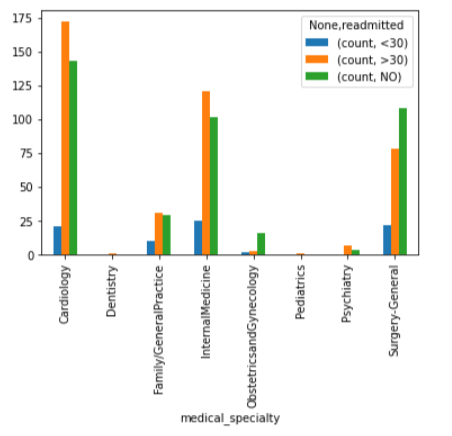
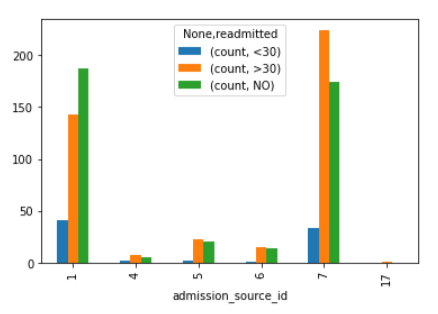
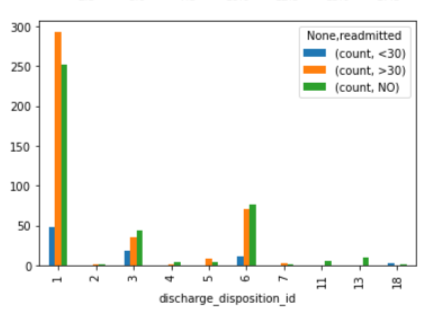
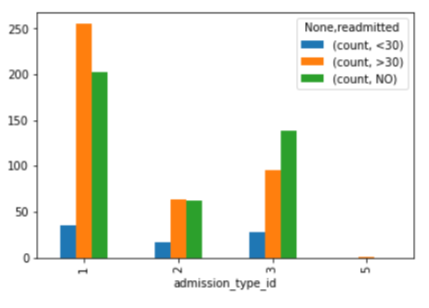
 

Fig.1. Distribution of glyburide(left) and change(right)

For the columns left in the models, they have the distribution in different class which can be related to the target column.





In these columns, different class has different performance on the target, in this way they maybe relative to the target column.

III.Model Building

First, I split our dataset into training and testing sets. I use the training data to train our model for predicting. Then the testing data will be used to verify the predicted price by the model. In this project, I randomly select 70% of the raw data as the training set, and the remaining 30% of the data is the testing set. That is, the training set contains 623 observations, while the testing set contains 268 observations. The amount of data is quite small which means the model can be easily overfitting.

Secondly, I use standard scaler scale the training data, using Logistic regression, Random Forest, KNN, SVM, naive bayes, XGboost and Decision Tree to train the model. And also using the scale to normalize the training data, which find out the following accuracy score of the training set and some of the test set:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Logistic Regression | Random Forest | KNN | SVM | Naive bayes | Decision Tree | XGboost |
| 0.54896 | 0.60353 | 1 | 0.53933 | 0.52809 | 1 | 0.38363 |
|  |  | 0.22761 |  | 0.4834 | 0.46269 |  |

According to our evaluation results, logistic regression, random forest, SVM, Naive Bayes and XGboost show underfitting with a accuracy score less than 0.7, while KNN and Decision Tree show overfitting on training data set with a high accuracy score at 1 and low accuracy score on test set. These bad performance of the models may because of the lack of training data, the result could be better with a large sum of data set and less amount of variables.