Diabetes Readmittance

SCS_3253_047 Machine Learning

Bailey Cameron

Irina Belaya

Nimaliny Krishnan

Understanding the problem

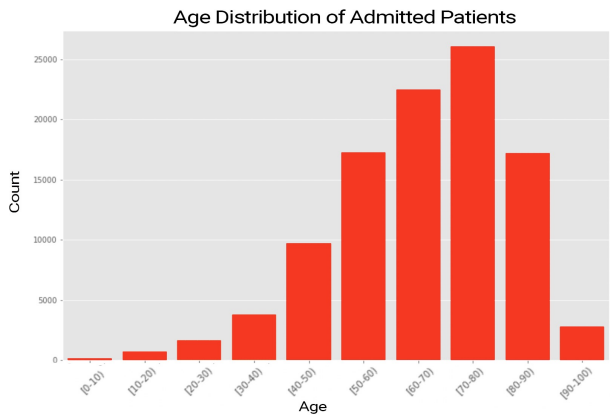
- Diabetes is one of the **most expensive** chronic diseases.
- **High risk of readmission** for patients with diabetes.
- ↓ Readmission rates for diabetic patients = ↓ Medical cost ↑ Health care quality

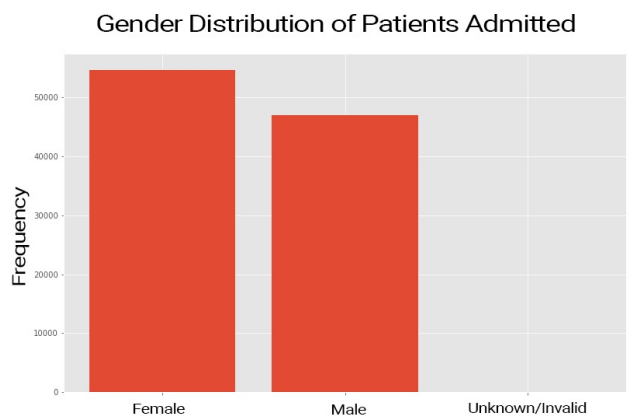
Objective:
To predict the likelihood of a diabetic patient being readmitted

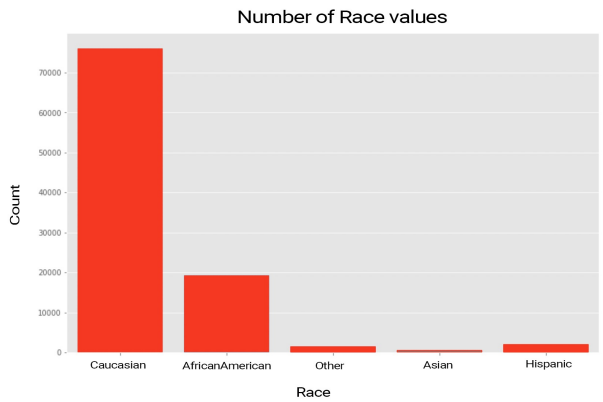
About the data

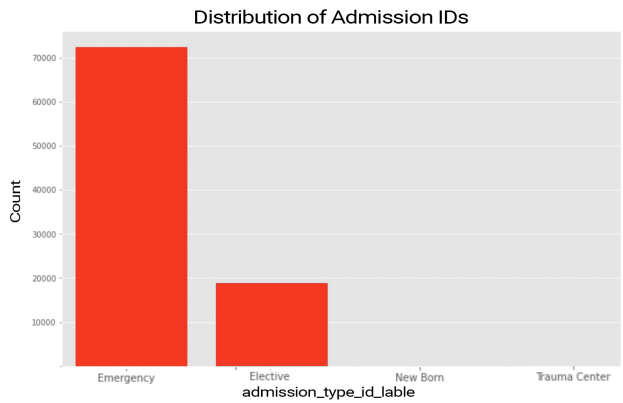
- 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks.
- 101 766 records with 71 515 of unique diabetic patients.
- 16 773 patients have been admitted more than once.
- 50 variables related to demographic, medication information, hospital interaction information

Exploring the Data

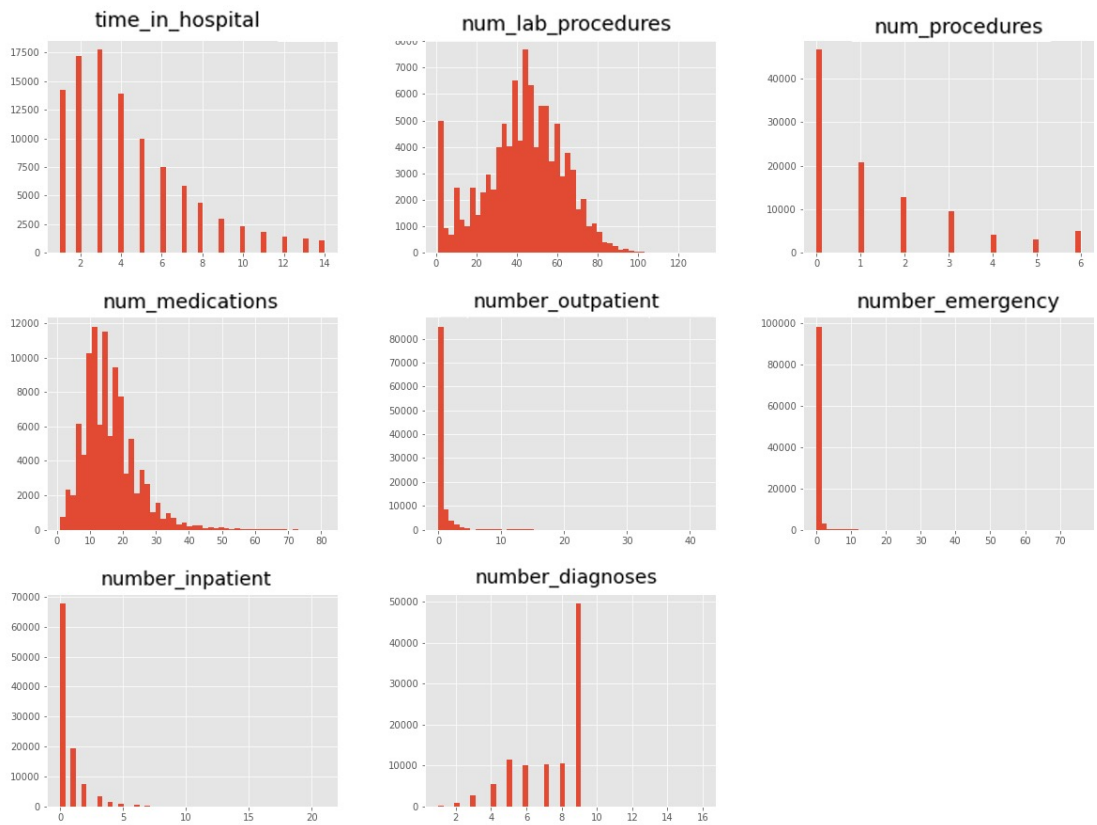




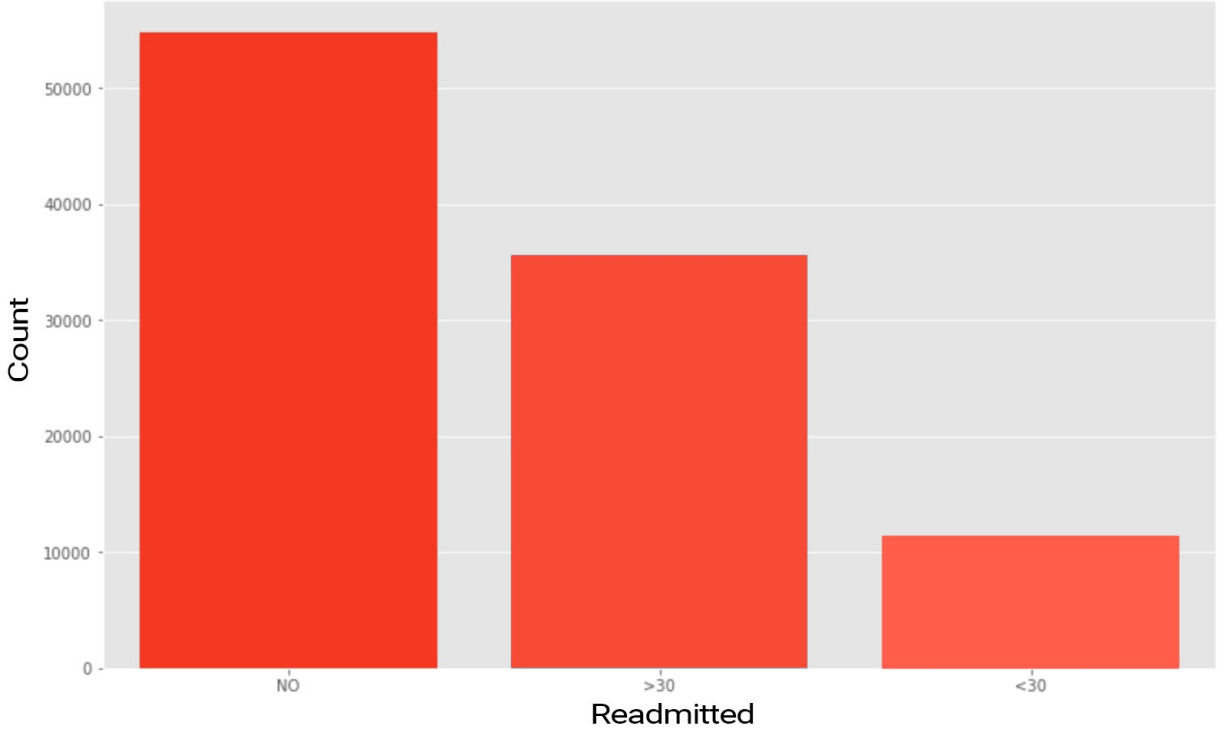




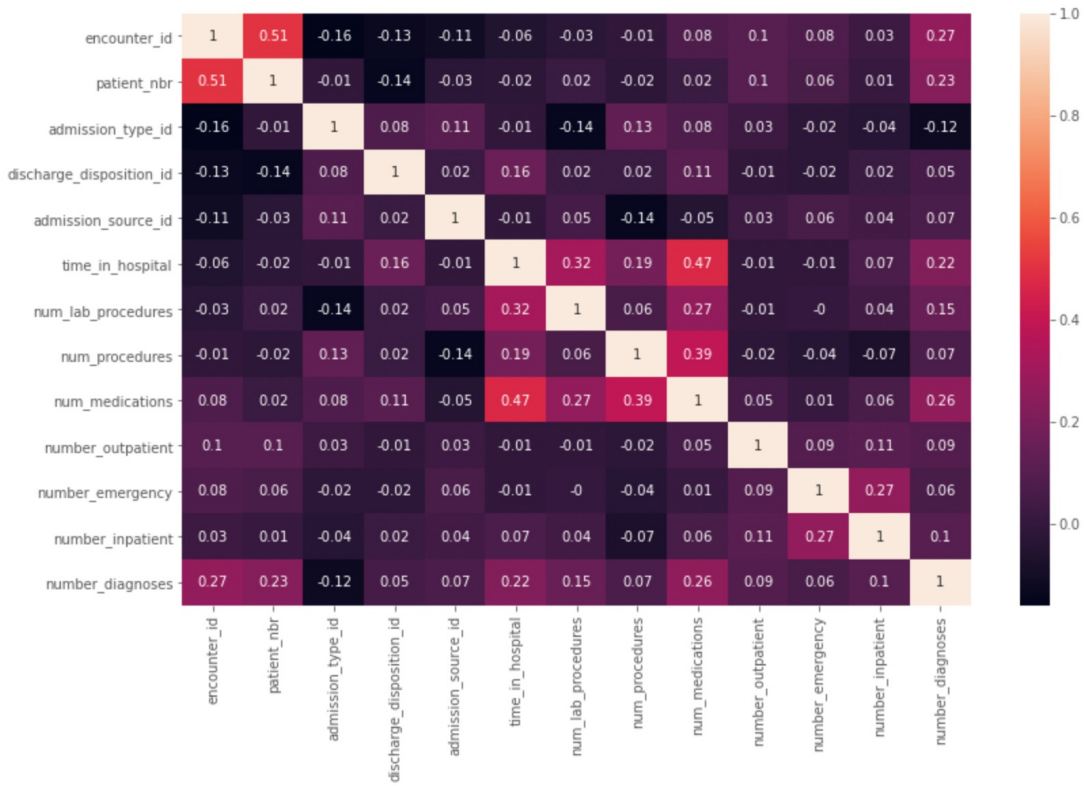
Exploring the Data: Numeric data distribution



Exploring the Data: Target Values



Exploring the Data: Variable Correlation



Data Cleaning

```
[27]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 101766 entries, 0 to 101765
Data columns (total 50 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   encounter_id              101766 non-null  int64
 1   patient_nbr               101766 non-null  int64
 2   race                      99493 non-null   object
 3   gender                    101766 non-null  object
 4   age                       101766 non-null  object
 5   weight                    3197 non-null    object
 6   admission_type_id         101766 non-null  int64
 7   discharge_disposition_id  101766 non-null  int64
 8   admission_source_id       101766 non-null  int64
 9   time_in_hospital          101766 non-null  int64
 10  payer_code                61510 non-null   object
 11  medical_specialty         51817 non-null   object
 12  num_lab_procedures        101766 non-null  int64
 13  num_procedures            101766 non-null  int64
 14  num_medications           101766 non-null  int64
 15  number_outpatient         101766 non-null  int64
 16  number_emergency          101766 non-null  int64
 17  number_inpatient          101766 non-null  int64
 18  diag_1                    101745 non-null  object
 19  diag_2                    101408 non-null  object
 20  diag_3                    100343 non-null  object
 21  number_diagnoses          101766 non-null  int64
 22  max_glu_serum             101766 non-null  object
```

```
[83]: (df1.isna().sum()/len(df1)*100).round(2)
```

```
[83]: encounter_id                  0.00
      patient_nbr                   0.00
      race                          2.23
      gender                        0.00
      age                           0.00
      weight                       96.86
      admission_type_id             0.00
      discharge_disposition_id      0.00
      admission_source_id           0.00
      time_in_hospital              0.00
      payer_code                   39.56
      medical_specialty            49.08
      num_lab_procedures            0.00
      num_procedures                0.00
      num_medications               0.00
      number_outpatient             0.00
      number_emergency              0.00
      number_inpatient              0.00
      diag_1                        0.02
      diag_2                        0.35
      diag_3                        1.40
      number_diagnoses              0.00
      max_glu_serum                 0.00
```

1. Drop column: **weight**
2. Drop rows with missing values for: **race**
3. Drop column: **medical specialty of admitting physician**
4. Drop column: **payer code**
5. Drop **diag_1**, fill in **diag_2, diag_3** with nulls

Data Cleaning

[36]:
```
# get top 5 diagnosis codes
df3['diag_1'].value_counts(ascending=False)*100/len(df3)
```

[36]:
```
428    6.774703
414    6.439938
786    3.958863
410    3.536638
486    3.443145
         ...
318    0.001005
217    0.001005
649    0.001005
906    0.001005
V60    0.001005
Name: diag_1, Length: 714, dtype: float64
```

OneHotEncoder was created for all categorical variables

diag_2_cat_428	diag_2_cat_486	diag_2_cat_786	diag_2_cat_Other
0	0	0	1
0	0	0	1
0	0	0	1
0	0	0	1
0	0	0	1

Numeric variables were standardized

Test and training sets

```
from sklearn.model_selection import train_test_split
train_set, test_set = train_test_split(df3, test_size=0.3, random_state=42)
```

```
target = 'readmitted'
features = list(train_set.columns)
features = [f for f in features if f != target]
```

```
# Train set
X_tr = train_set[features]
y_tr = train_set[[target]]

# Test set
X_te = test_set[features]
y_te = test_set[[target]]
```

Models: Binary Classifier

- The target (readmittance) was converted into a binary variable ‘yes’ representing that the patient was readmitted and ‘no’ representing that patient was not readmitted.

```
y_train_no = (y_tr != 'NO').values.ravel()
y_test_no = (y_te != 'NO').values.ravel()
```

Models: Binary Classifier 1

- **SGD Classification**
  - The model performed poorly with a score of 0.57

```
cross_val_score(sgd_clf, X_tr, y_train_no, cv=3, scoring="accuracy")
```

```
array([0.5650769 , 0.57794054, 0.57794054])
```

Tuning the model: Binary Classifier 1

- **ROC AUC score:** 0.68 (best alpha: 0.1)
-

Accuracy score	F1 score	Precision score	Recall score
0.63	0.59	0.34	0.69

- PCA results are approximately equivalent ROC AUC

Models: Binary Classifier 2

- **Random Forest**
  - The model performed with a score of 0.61

```
cross_val_score(forest_clf, X_tr, y_train_no, cv=3, scoring="accuracy")
```

```
array([0.60402395, 0.60607497, 0.60659199])
```

Tuning the model: Binary Classifier 2

- **ROC AUC score:** 0.68 (best n_estimators: 200)
-

Accuracy score	F1 score	Precision score	Recall score
0.63	0.63	0.55	0.62

Next step: Refining Data Cleaning

- Removed duplicate patient encounters and keep only first
- Dropped columns: *encounter_id, patient_nbr, weight, medical_specialty, payer_code*
- Converted *age, glu_dict* (glucose level), *a1c_dict, med_dict* (list of all medications) values into numeric
- Remap *admission_type_id, discharge_disposition_id, admission_source_id*
- Encoded converted values (OneHotEncoder)

Next step: Obtaining new results

- **SGD Classification**

	Accuracy score	F1 score	Precision score	Recall score
Old Results	0.63	0.59	0.34	0.69
New Results	0.6256	0.5859	0.3424	0.6906

- **Random Forest**
  - max_depth: 15, n_estimators: 200

	Accuracy score	F1 score	Precision score	Recall score
Old Results	0.63	0.63	0.55	0.62
New Results	0.6424	0.6318	0.5123	0.6409

Key Results

We built a predictive model in order to identify diabetic patients who have higher likelihood of being readmitted to the hospital

Based on the analysis of 2 machine learning algorithms:

SDG Classification and Random Forest

ROC curve (accuracy scores)

The best model is **Random Forest**, which yielded an accuracy of 64.2%.

How the model can be used

- **Model can be used to identify future diabetes patients who may be at risk of readmittance to the hospital**
  - **While the model is still being refined, the data can be used as an indicator.**
- **Long-term goals of the model will be to improve patient care and reduce hospital costs.**