Hospital Readmissions Reduction Program

Predictive Analytics

Content

Business Case: Analysis of a Hospital patient's data.

- Analyze the cost of post-discharge care plans versus their effects on patient readmissions
- Design a predictive model predicting the patient readmission related metrics

Data Acquisition: PCCI HRRP patient dataset was obtained in xlsx format

Data Visualization/ Exploratory Data Analysis (EDA): Get a better understanding of the problem statement with graphs and charts, descriptive statistics using **Tableau**

Data Preparation: Data preprocessing, Feature extraction, Feature engineering, Feature scaling and selection.

Predictive Analysis with Machine Learning: Find the suitable machine learning algorithm, train, score, evaluate and tune the prediction model.

Python is used for Data Preparation, EDA, Predictive Analysis with Machine learning

Data Visualization - Tableau

Tableau is an interactive data visualization tool used for Exploratory Data Analysis (EDA), where charts/graphs are plotted for dimensions (qualitative values) against measures (quantitative values) and dependent variables (readmit30) to get insights and understand their data. Exploratory Data Analysis (EDA) is an approach to analyzing datasets to summarize their statistical characteristics, often with visual methods.

Tableau is quick, simple, user-friendly, intuitive, can handle lot of data, provide statistical calculations on datasets

EDA:

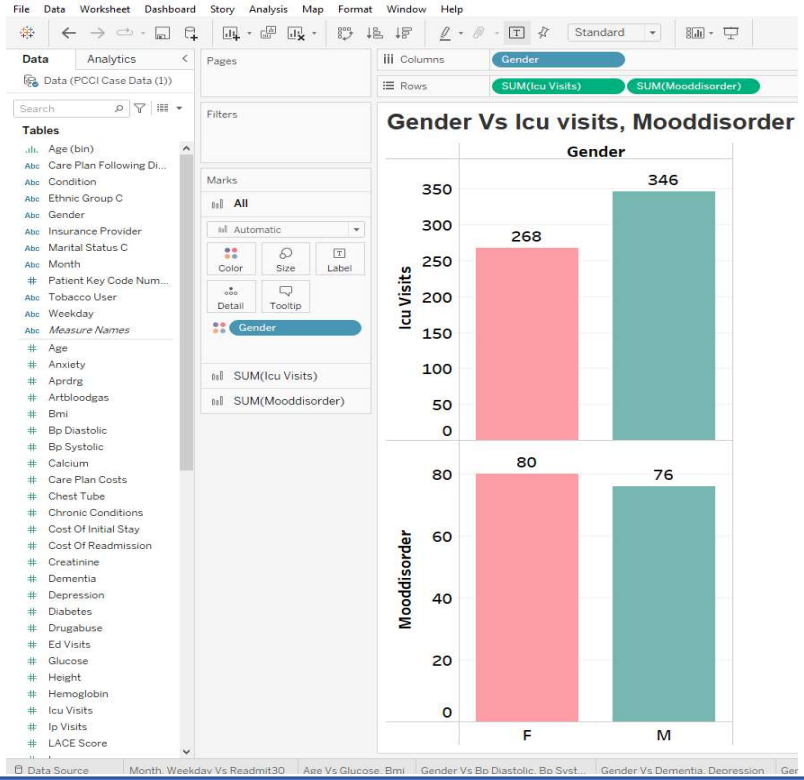
- Get a better understanding of data that may not be analyzed by standard data science algorithms.
- Understanding data patterns that may be skipped by typical machine learning algorithms.
- Drawing charts and graphs for better understanding from different angles and projects the results.
- To get a better understanding of the problem statement, visually.
- To find the hidden trends and relationship between variables.
- Assess and validate your assumptions on the variables, whether the variables help answer business problem or not.
- Screen for noise variables, missing data, outliers, etc. Find which variables need imputation, preprocessing

Tableau

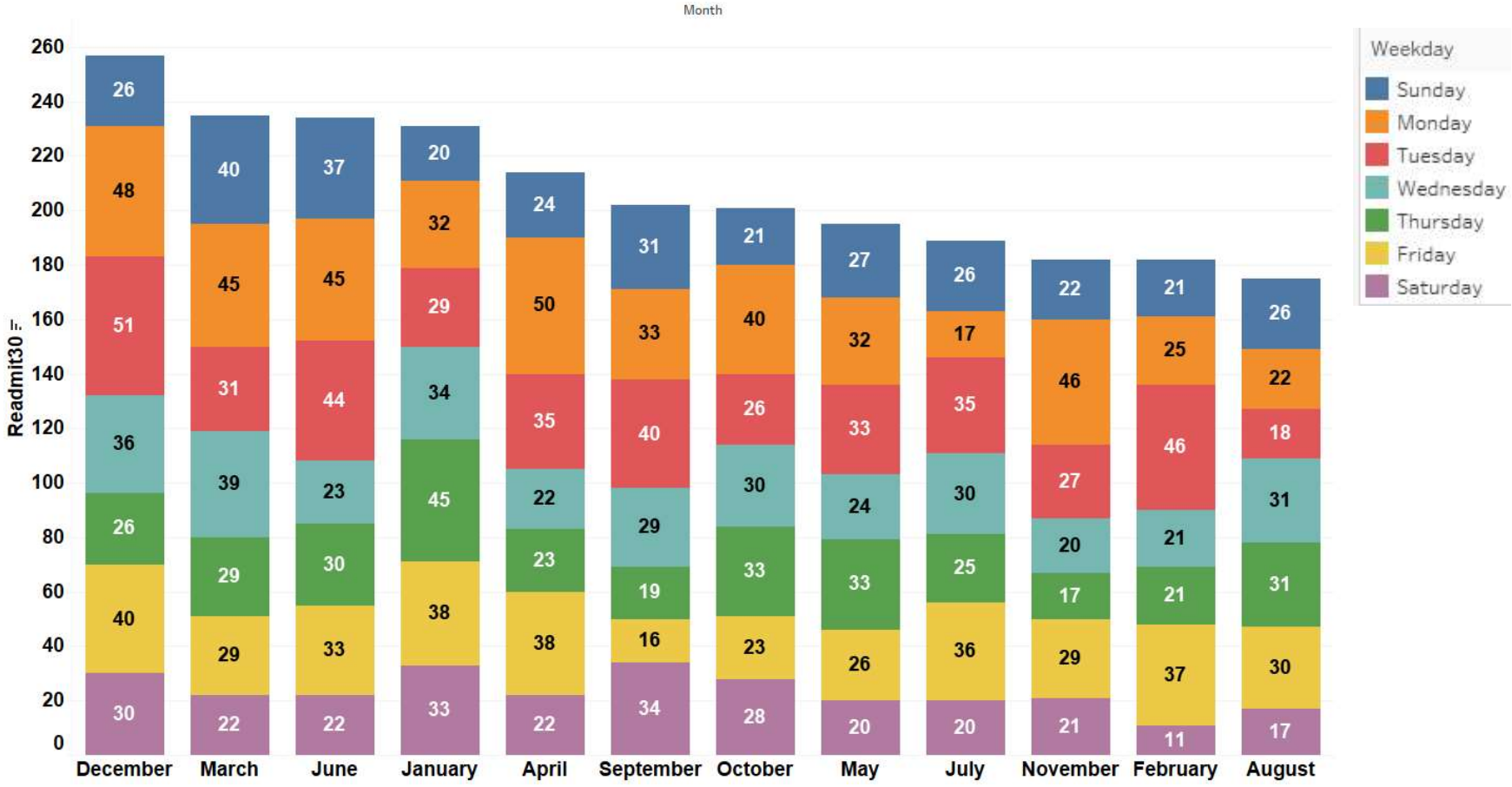
Chart Views

1. Text tables
2. Heat maps
3. Highlight tables
4. Symbol maps
5. Maps
6. Pie charts
7. Horizontal bars
8. Stacked bars
9. Side-by-side bars
10. Tree maps
11. Circle views
12. Side-by-side circles
13. Lines (continuous)
14. Lines (discrete)
15. Dual lines
16. Area charts (continuous)
17. Area charts (discrete)
18. Dual combination
19. Scatter plots
20. Histogram
21. Box and whisker plots
22. Gantt
23. Bullet graphs
24. Packed bubbles

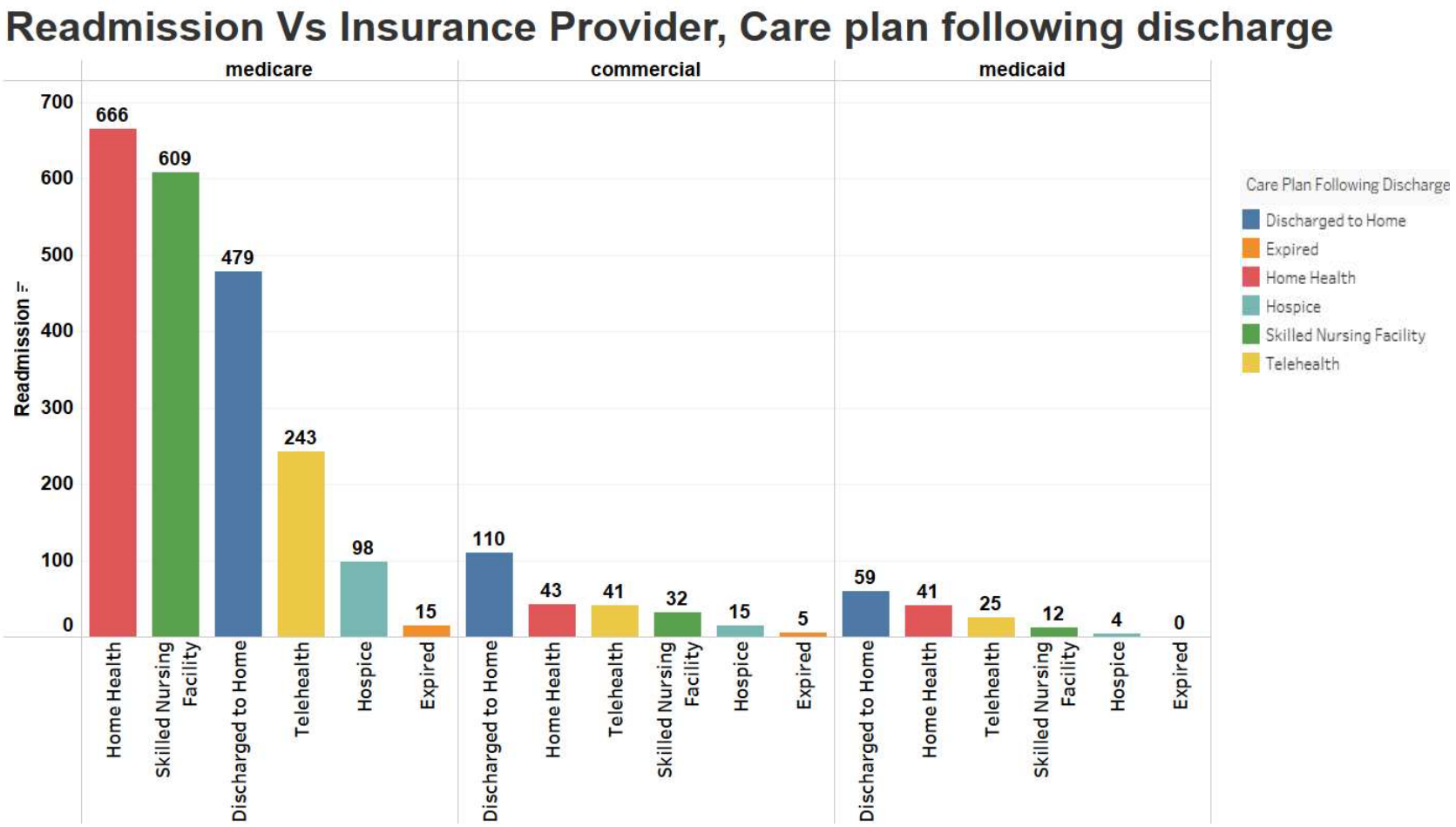
Data Pane, Marks card and Worksheet



Month, Weekday Vs Readmit30



Month and weekdays do not significantly affect the Readmission. They must be noise variables.



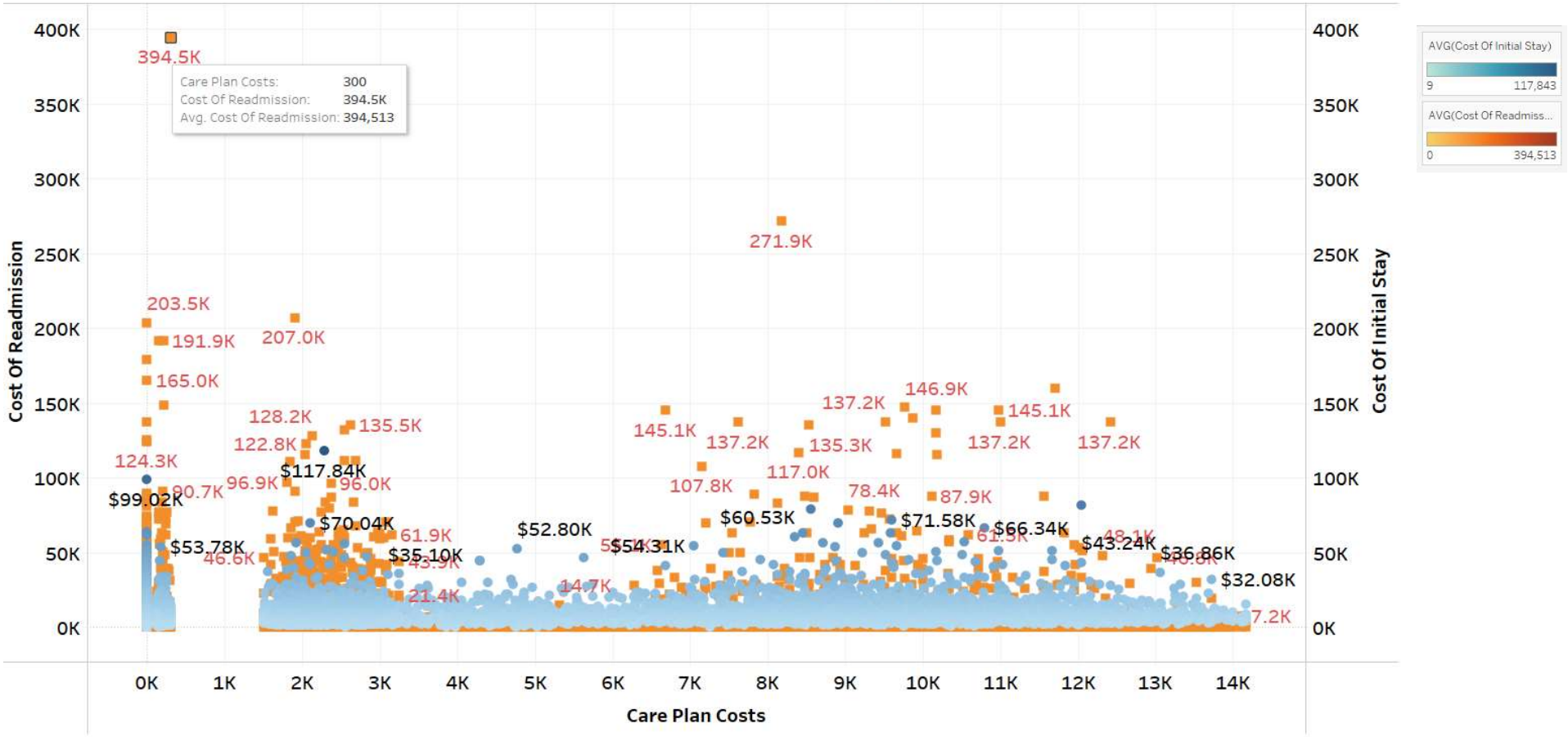
Patients who were on Medicare and had 'Home Health' as Care plan following discharge, were the most readmitted to hospital within 30 days, compared to other plans.

Care Plan Following Discharge Vs Readmission,Cost of Initial Stay, Cost of readmission

Patients who were discharged to **Skilled nursing facility** had high average cost of initial stay, high average cost of readmission.

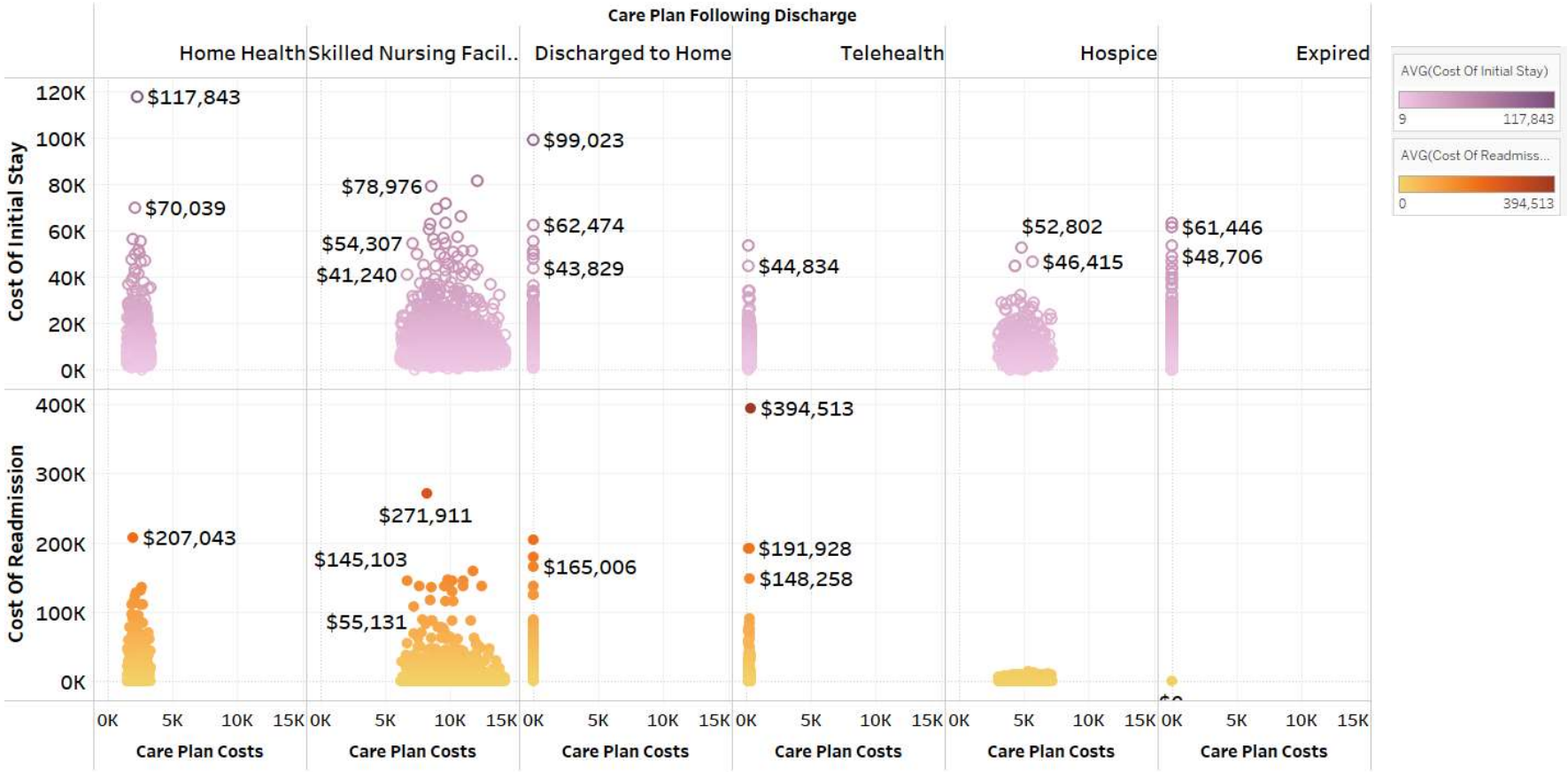
Patients who had **Home Health** as Care plan following discharge, were the most readmitted to hospital within 30 days, compared to other plans.

Care Plan costs Vs Cost of Initial Stay, Cost of Readmission



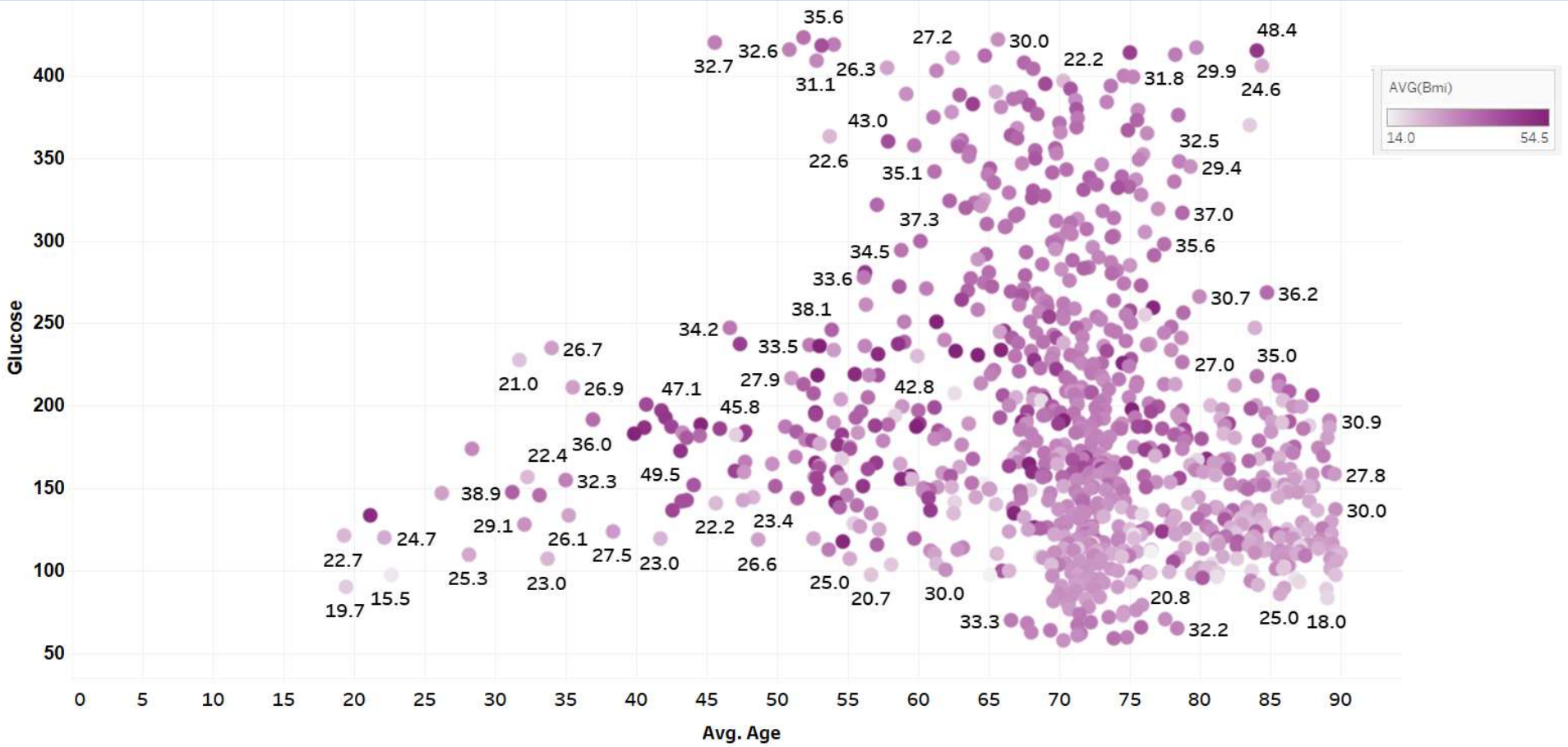
$394K in ‘Cost of Readmissions’ and $117K in ‘Cost of initial stay’ are far away from other values, so they must be outliers.

Care Plan Following Discharge, Care Plan Costs Vs Cost of Initial Stay, Cost of Readmission



'Cost of Readmission' is less for higher 'Care Plan costs', except for 'Skilled Nursing Facility' plan

Age Vs Glucose, Bmi – Scatter plots



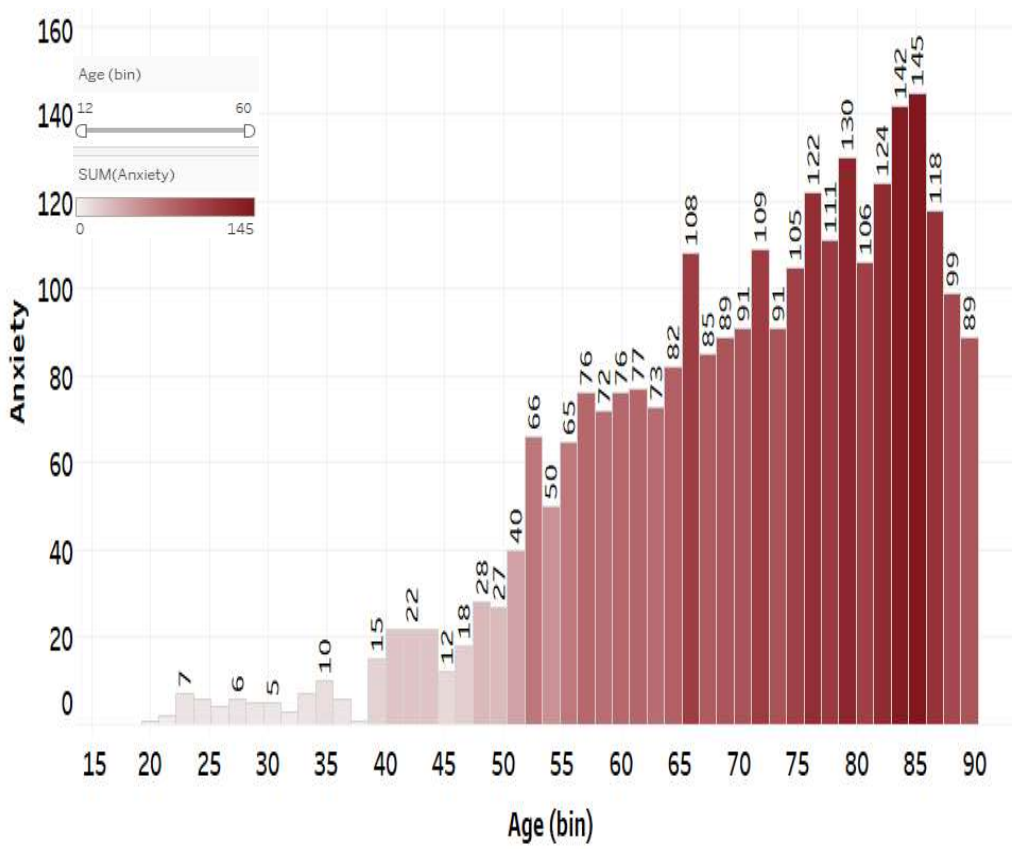
Glucose levels and Bmi increase with age. There are more people with high glucose and high Bmi, above age 40.

Age Vs BMI – treemap

Age Vs Anxiety

Packed bubbles

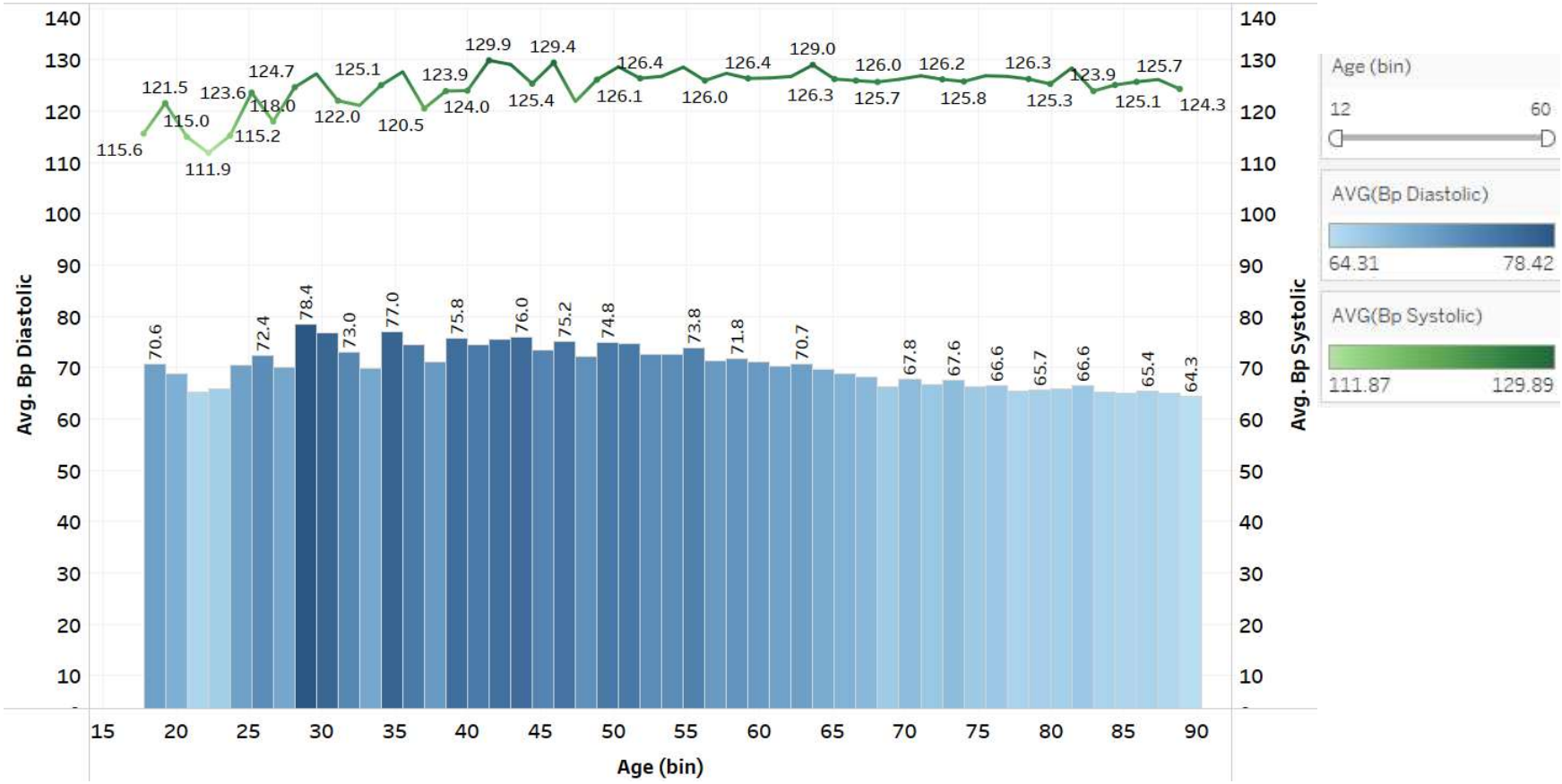
Horizontal bars



Anxiety is mostly present for patients between age 40 to 90. Most patients aged 85 has the maximum count for anxiety

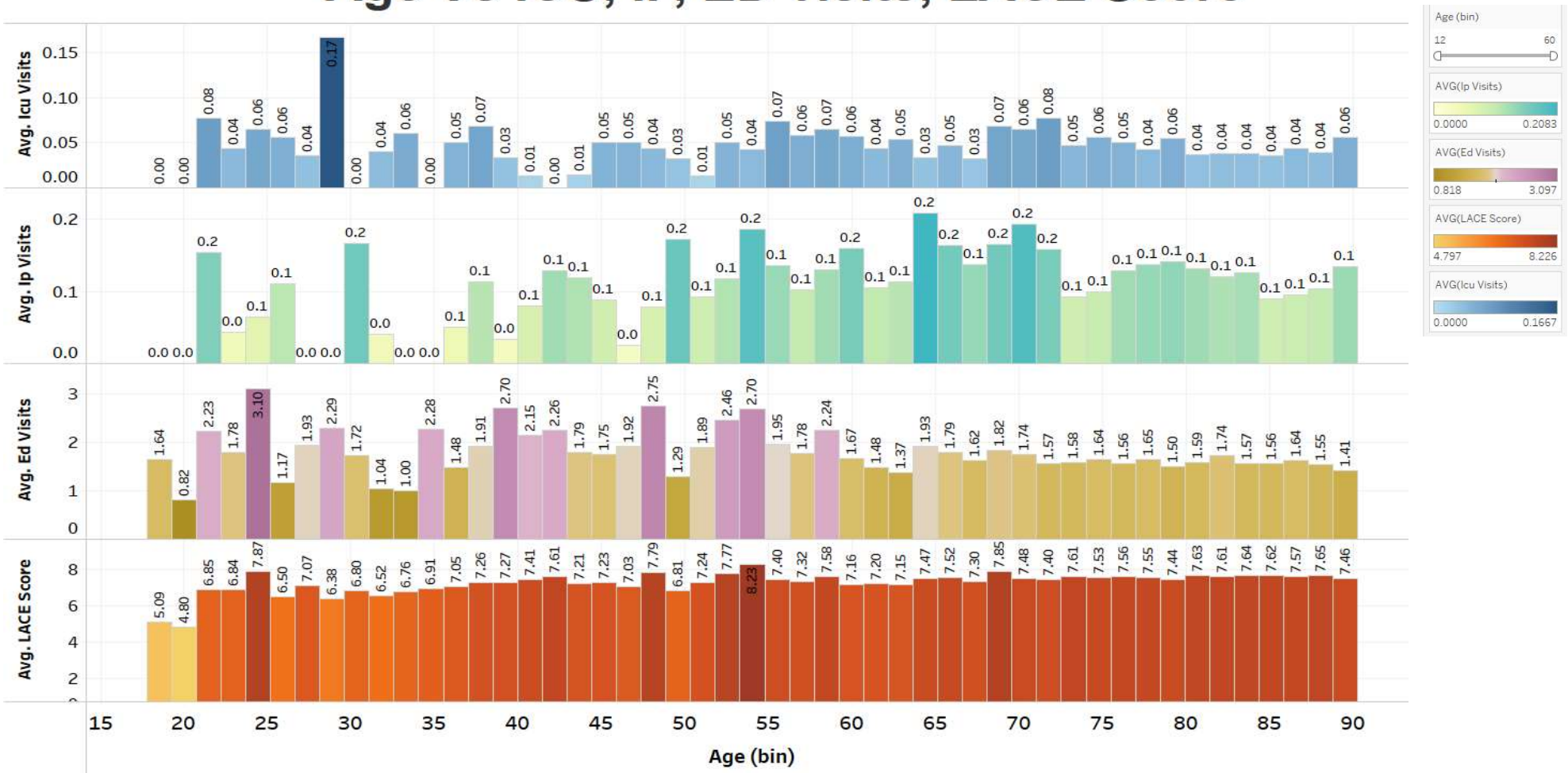
Age Vs Bp Diastolic, Bp Systolic

Dual, Synchronize axis



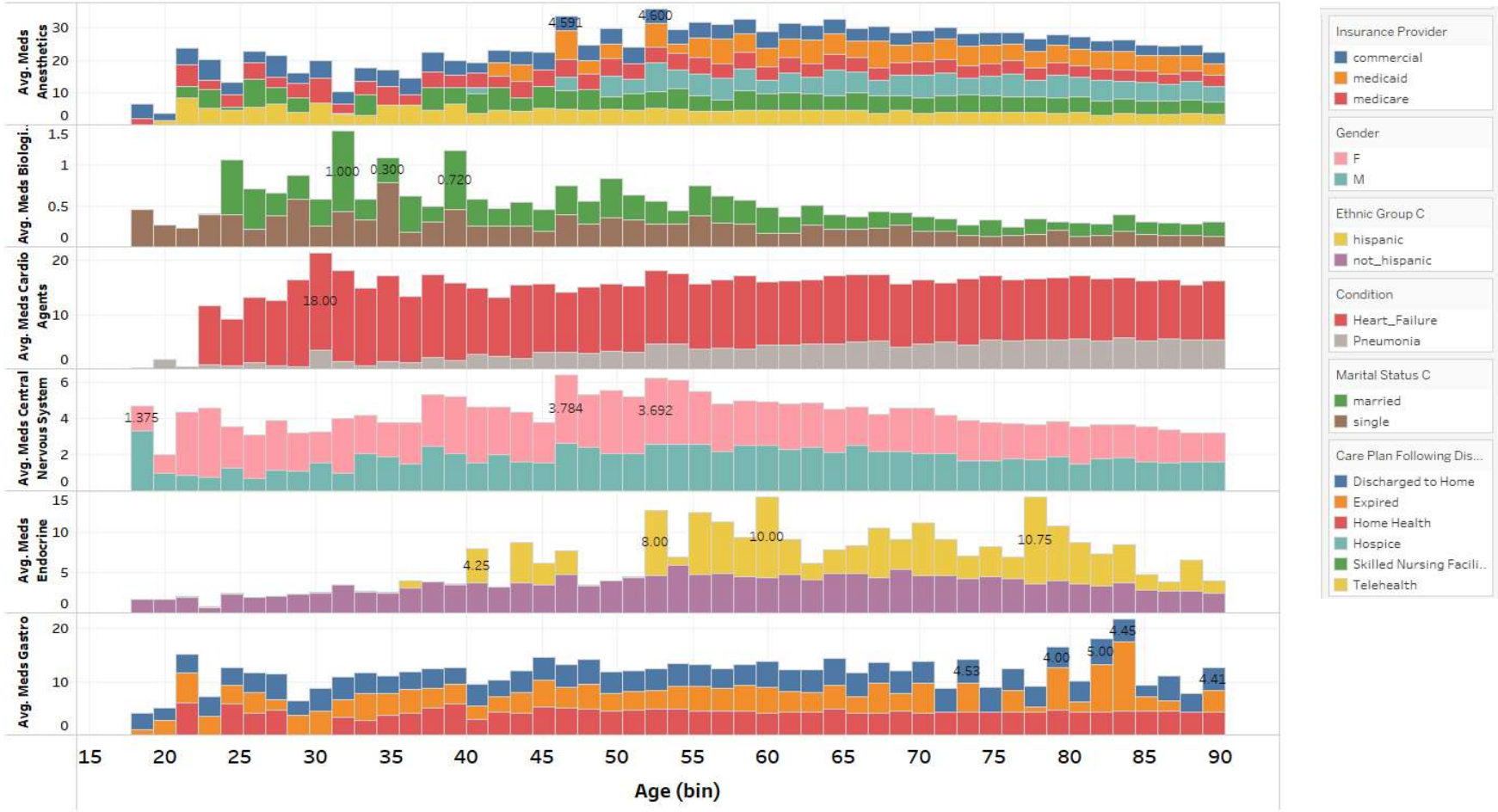
No linear relation between Age and Bp Diastolic and Bp Systolic.

Age Vs ICU, IP, ED visits, LACE Score



It is seen from the horizontal bar charts, there is no linear correlation between Age and IP, ED, IP visits and LACE score

Age Vs Meds - Anesthetics, Biological, Cardio Agents, Central Nervous System, Endocrine, Gastro



Mark type

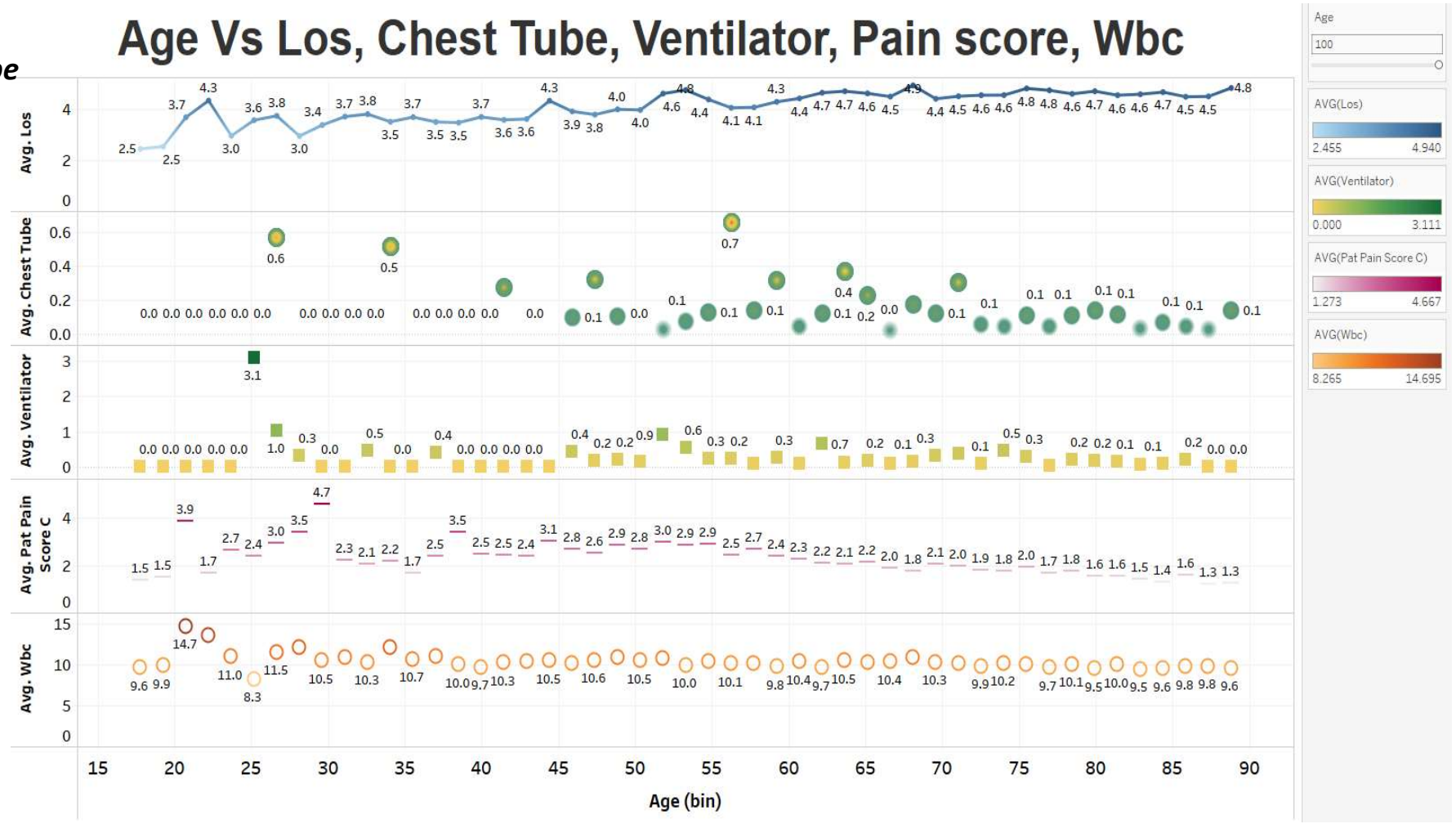
Line

Density

Square

Gantt bar

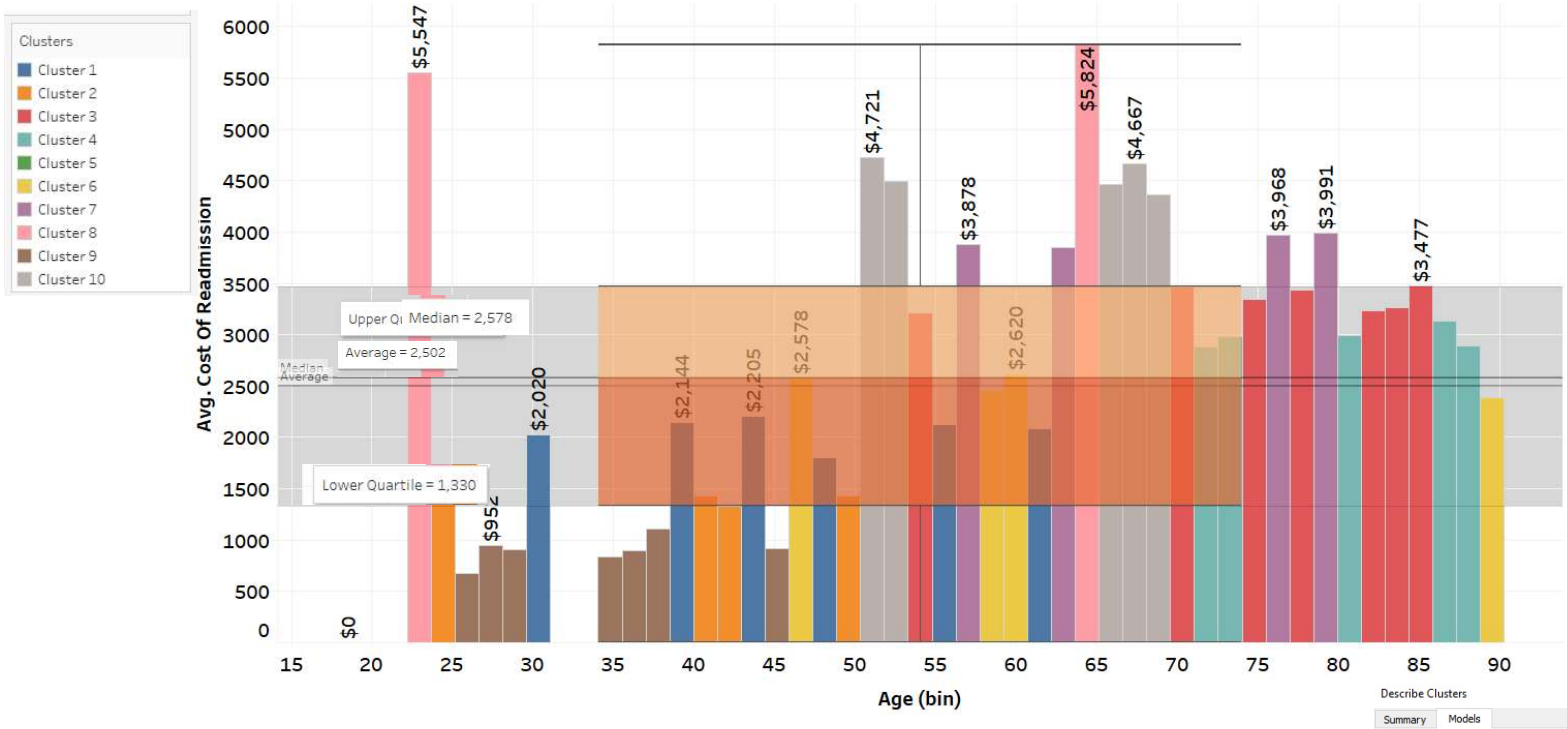
Shape



Different, there is no linear correlation between Age and Los, Chest Tube, Ventilator, Pain score, Wbc

Age Vs Cost of Readmission

Clustering, Analytics – Median, Average, Reference band



Average: 2502$, Median: 2578$, Interquartile range: 1330$ - 3470$

Describe Clusters

Summary | Models

Inputs for Clustering

Variables: Avg. Cost Of Readmission
Level of Detail: Age (bin)
Scaling: Normalized

Summary Diagnostics

Number of Clusters: 10
Number of Points: 49
Between-group Sum of Squares: 3.2426
Within-group Sum of Squares: 0.01603
Total Sum of Squares: 3.2586

Clusters	Number of Items	Centers: Avg. Cost Of Readmission
Cluster 1	6	2065.9
Cluster 2	4	1459.3
Cluster 3	7	3346.3
Cluster 4	5	2973.2
Cluster 5	5	0.0
Cluster 6	4	2512.3
Cluster 7	4	3920.6
Cluster 8	2	5685.6
Cluster 9	7	898.35
Cluster 10	5	4541.0
Not Clustered	0	

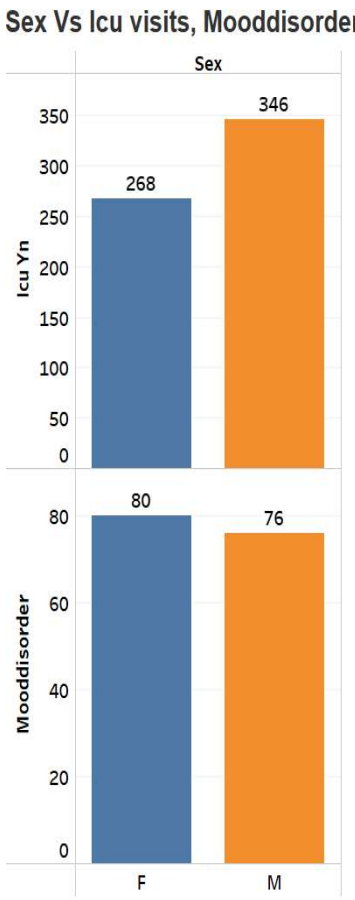
Describe Clusters

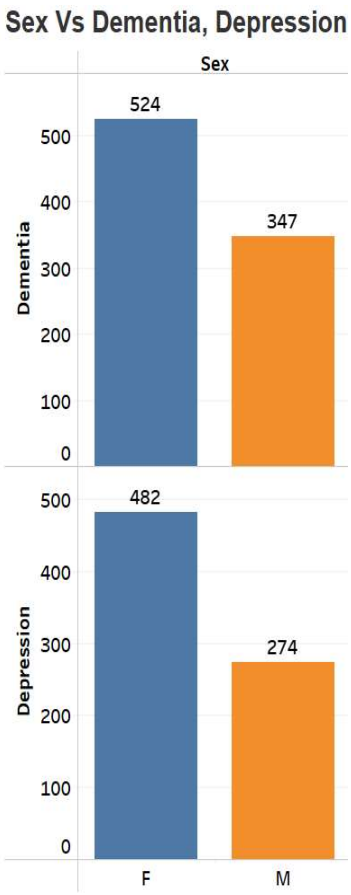
Summary | Models

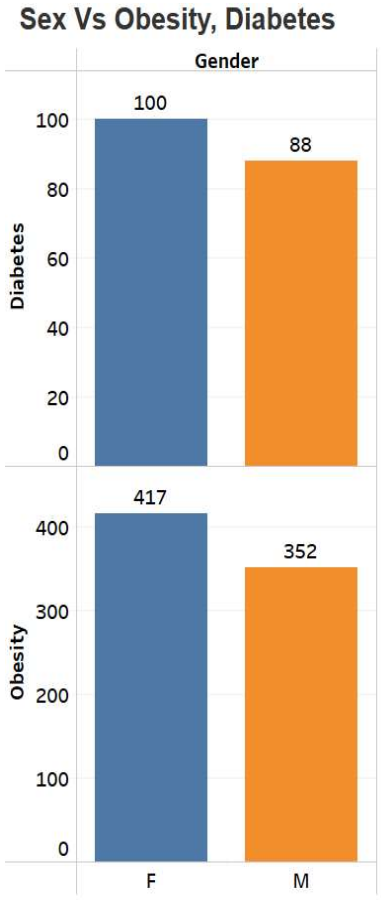
Analysis of Variance:

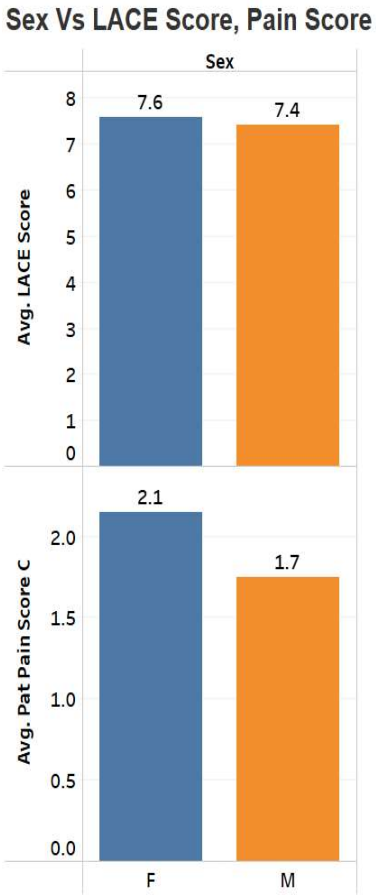
Variable	F-statistic	p-value	Model: Sum of Squares	Model: DF	Error: Sum of Squares	Error: DF
Avg. Cost Of Readmission	4.312	0.0006081	3.243	9	3.259	39

Gender Vs Features



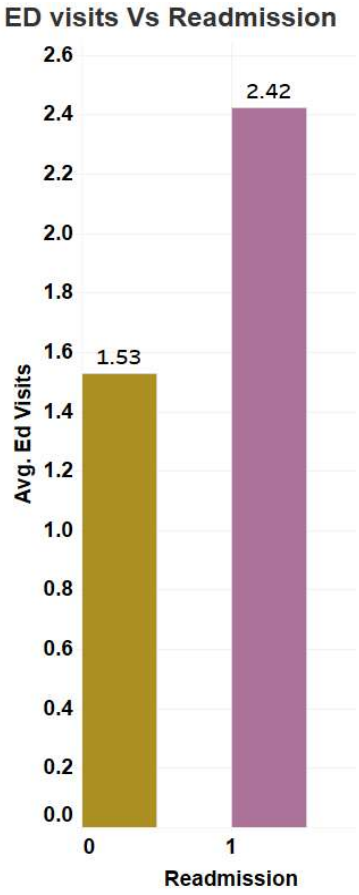


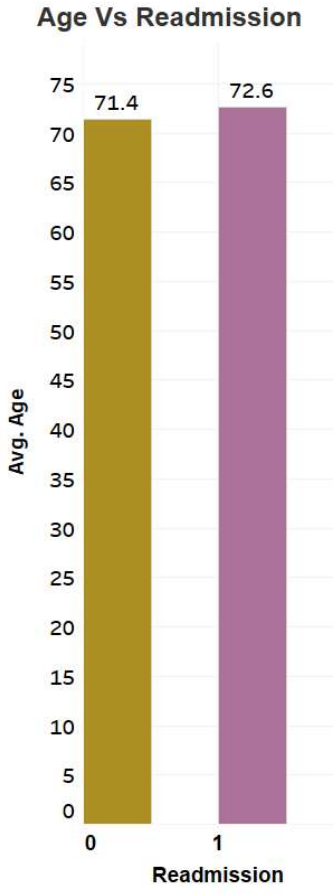


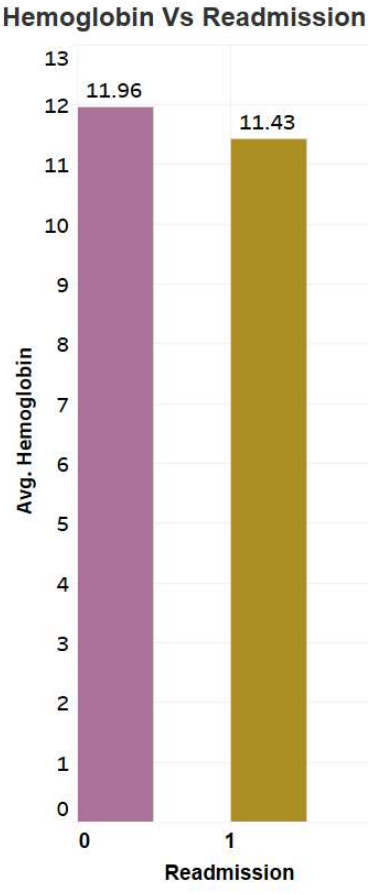


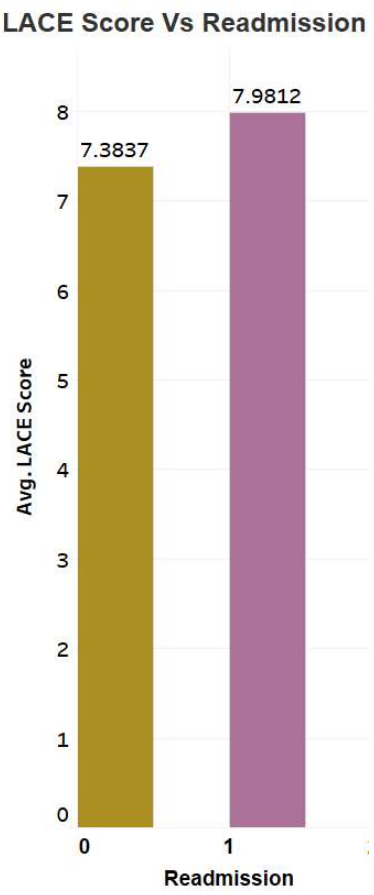
Men are admitted to ICU more than Women. More women suffer from Mood disorder, Dementia, Depression, Obesity, Diabetes than men.

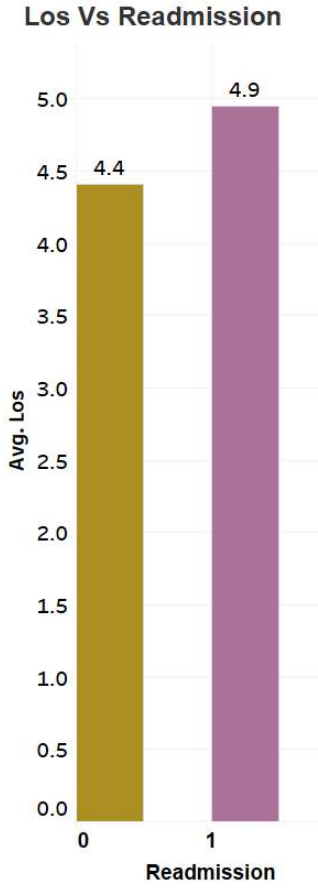
Readmission Vs Features











Patients who are readmitted have Average. ED visits - 2, Age - 72.6, Hemoglobin - 12, LACE score - 8, Length of stay - 5 days

Exploratory Data Analysis (EDA) - Python

The Features (attributes) are segregated into three categories:

Dependent Variable (Y): Variable that is being measured in the experiment. It changes as a result of the changes to the independent variables. Y values to predict:

Y : Readmit30– Classification model

Noise: Variable that does not affect the dependent variable.

Independent Variable or Predictor Variable (X): Variable whose change isn't affected by any other variable in the experiment.

Independent variable is the cause, and dependent variable is the effect.

Total number of features = 64
Total number of records = 12980

No duplicate values found
No missing values

Exploratory Data Analysis - Python

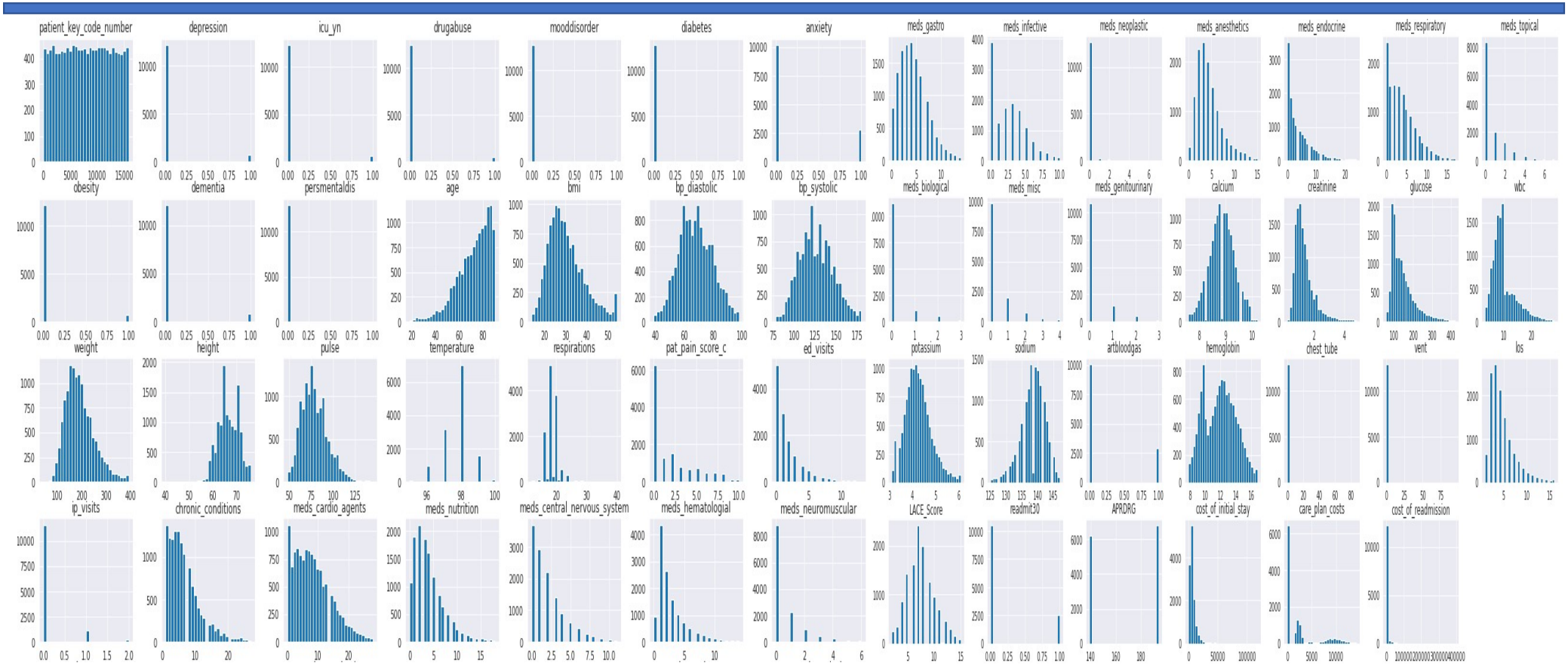
Python has open-source libraries that can automate the whole process of Exploratory Data Analysis and save a lot of time.

- **Autoviz** - performs automatic visualization of any dataset with one line
- **Pandas Profiling** - provides a very high-level API and allows a data scientist to generate a comprehensive HTML profile report . The report has five main sections: Overview, Variables, Interactions, Correlation, and Missing Values
- **Sweetviz** - generates visualizations which is useful in exploratory data analysis with just a few lines of codes. The library can be used to visualize the variables and comparing the dataset.
- **DataPrep** – dataprep.eda is the fastest and the easiest EDA tool. It allows you to understand a Pandas/Dask DataFrame with a few lines of code in seconds.
- **Dtale** - DTale is a Flask and React-based powerful tool which is used to analyze and visualize pandas data structure seamlessly. It supports different objects like Data Frame, Series, etc.

Top 10 Python Data Visualization Libraries

1. Matplotlib, 2. Seaborn, 3. Ggplot, 4. Plotly, 5. Geoplotlib, 6. Bokeh, 7. Folium, 8. Altair, 9. Pygal, 10. Gleam

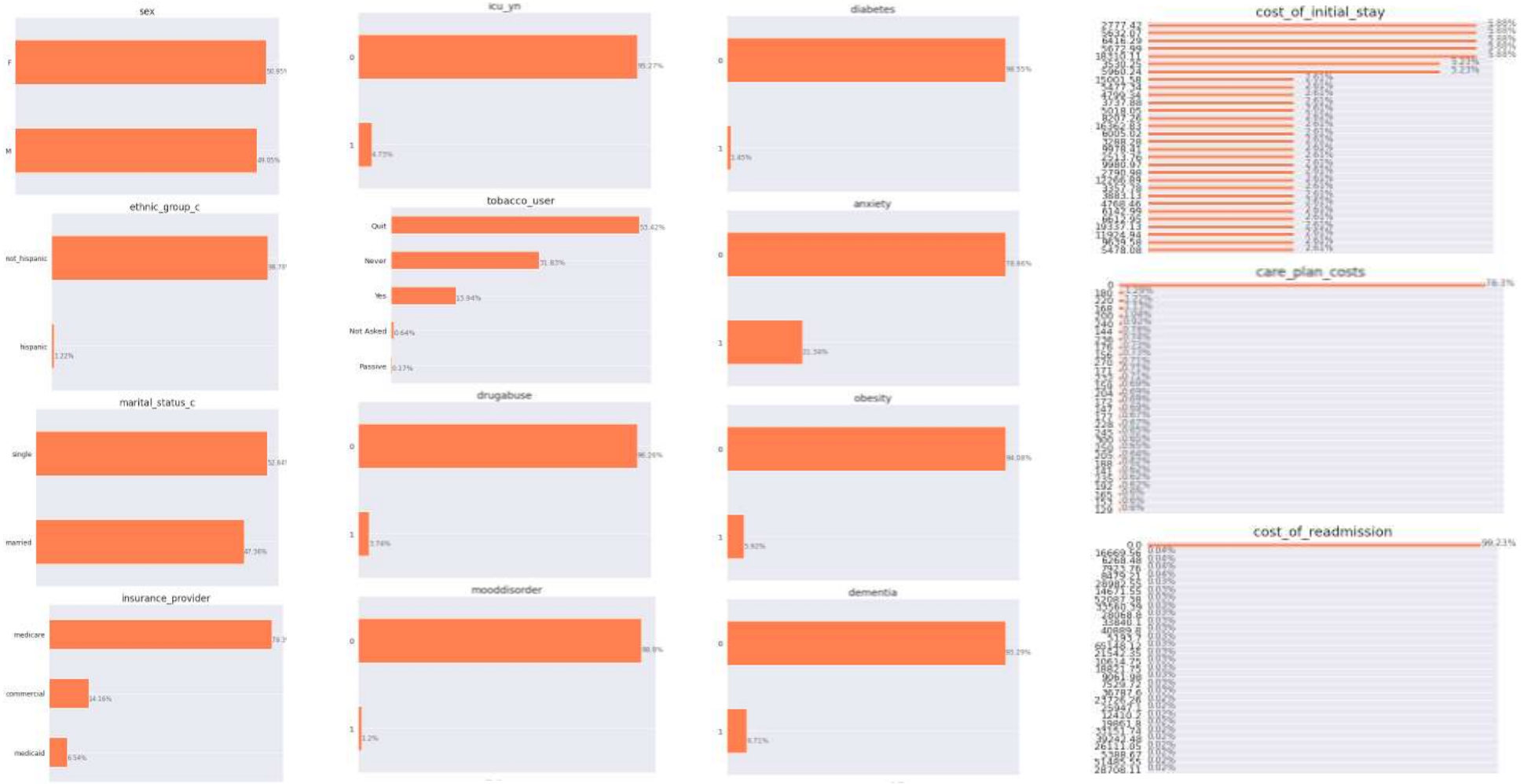
Histogram distribution of data



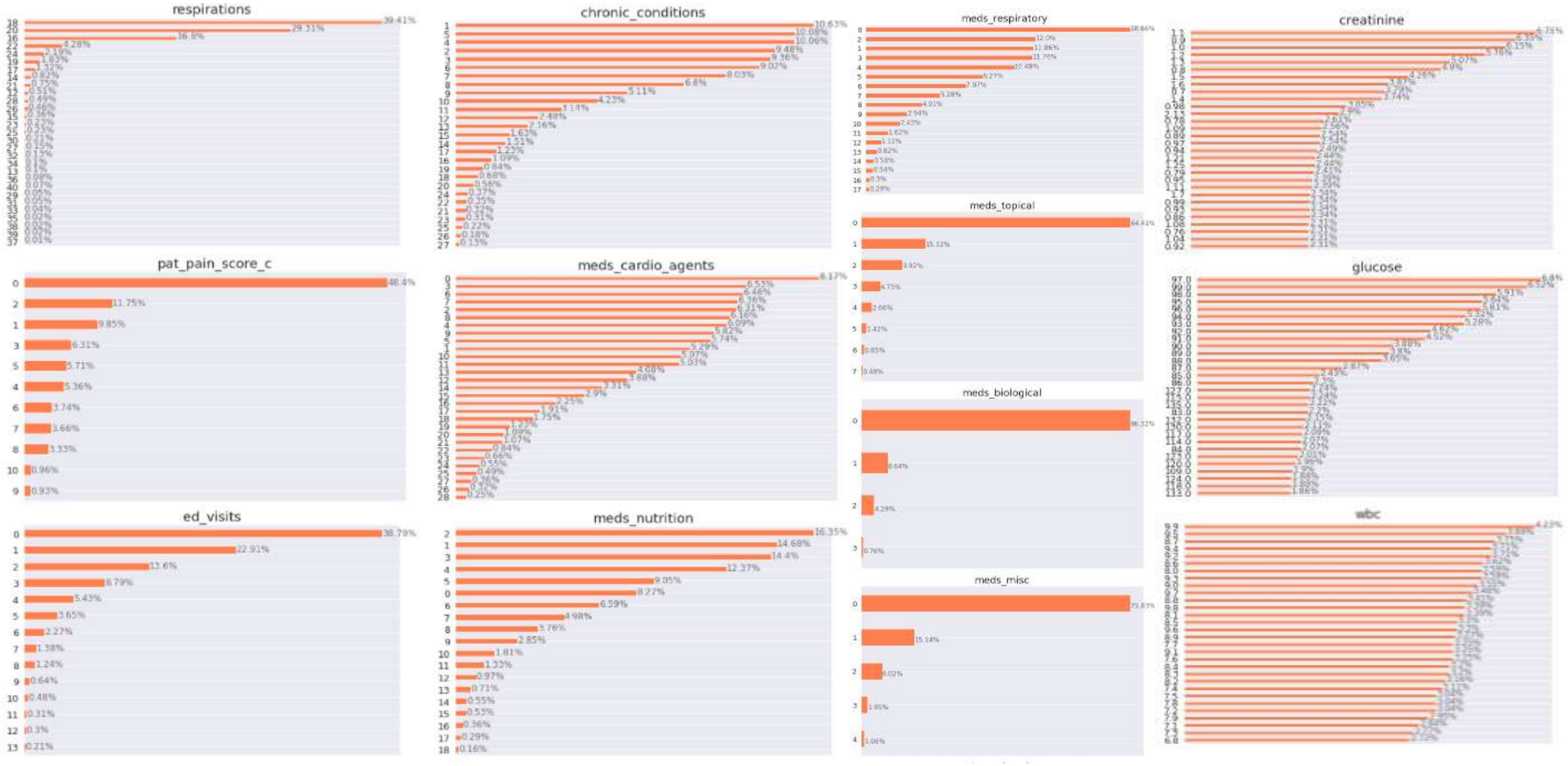
A histogram representation of the numerical distribution of features.

df.hist()function calls matplotlib.pyplot.hist(), on each series in the DataFrame, resulting in one histogram per column

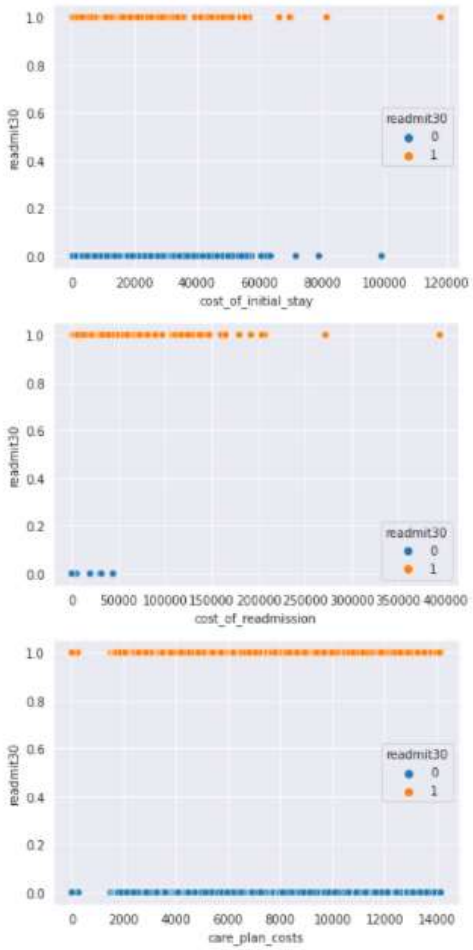
Histogram distribution of data

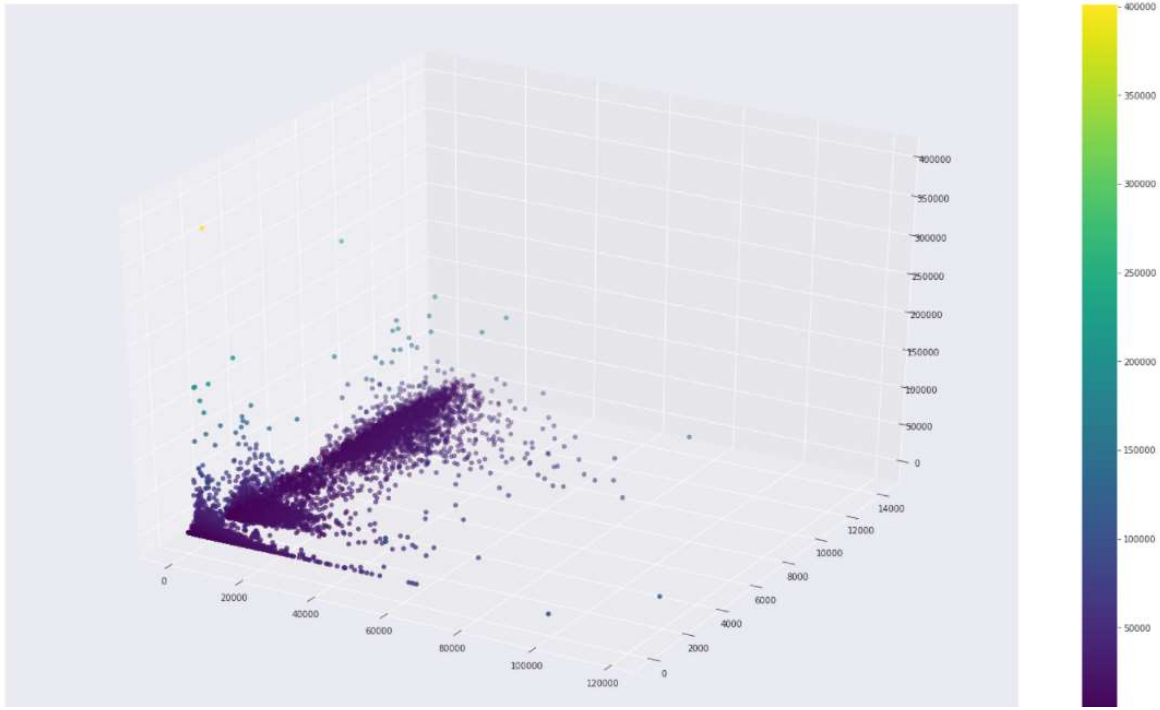


Histogram distribution of data



Scatter plot – numerical features



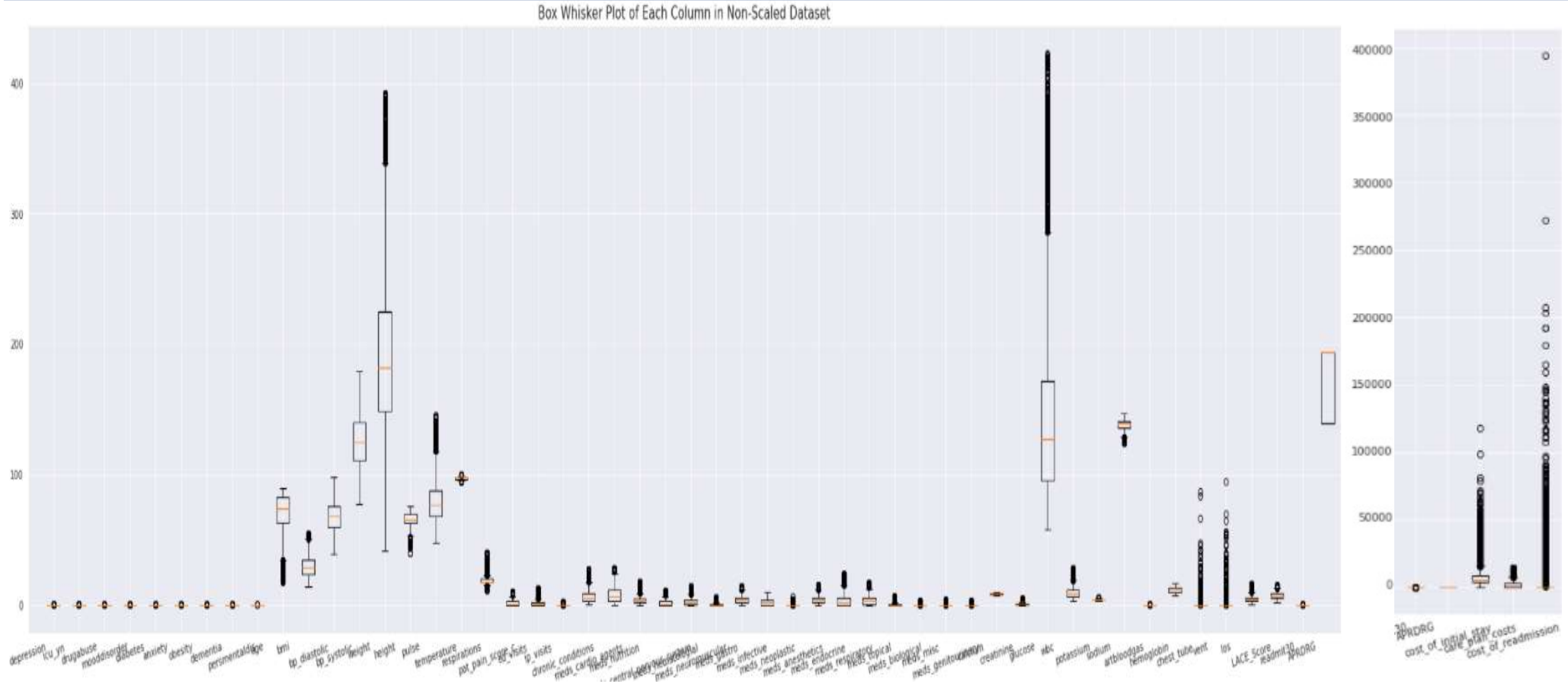


Cost of initial stay : 0 to 120K $

Cost of readmission: 0 to 400K $

Care plan costs: 0 to 14K $

Boxplot distribution of data



A boxplot is a graphical display of the distribution of data based on five key numbers: The “minimum”, 1st Quartile (25th percentile), median (2nd Quartile/ 50th Percentile), the 3rd Quartile (75th percentile), and the “maximum”. Any points that fall outside of these limits are referred to as outliers. Cost of readmission, Cost of initial stay have many outliers

DataPrep EDA – Correlation Matrix Heatmap

```
#create a full report

from dataprep.eda import create_report
create_report(df1)
```

DataPrep Report | Overview | Variables | Interactions | Correlations | Missing Values

Overview

Dataset Statistics

Number of Variables	64
Number of Rows	12980
Missing Cells	0
Missing Cells (%)	0.0%
Duplicate Rows	0
Duplicate Rows (%)	0.0%
Total Size in Memory	12.7 MB
Average Row Size in Memory	1.0 KB
Variable Types	Numerical: 35 Categorical: 29

Dataset Insights

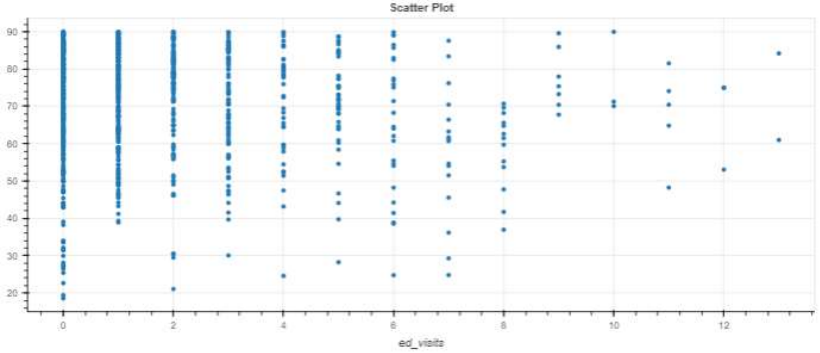
- patient_ is uniformly distributed — Uniform
- chest_t_ and vent have similar distributions — Similar Distribution
- respira_ is skewed — Skewed
- pat_pai_ is skewed — Skewed
- ed_visi_ is skewed — Skewed
- meds_nu_ is skewed — Skewed
- meds_ce_ is skewed — Skewed
- meds_he_ is skewed — Skewed
- meds_ga_ is skewed — Skewed
- meds_in_ is skewed — Skewed

1 2 3 4 5 6

Interactions

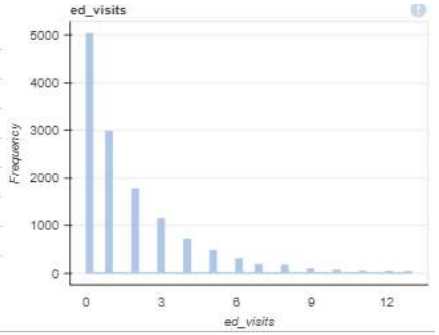
X-Axis: ed_visits Y-Axis: age

Scatter Plot



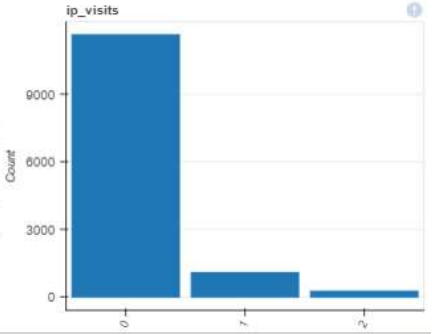
ed_visits
numerical
Show Details

Approximate Distinct Count	14	Mean	1.6991
Approximate Unique (%)	0.1%	Minimum	0
Missing	0	Maximum	13
Missing (%)	0.0%	Zeros	5035
Infinite	0	Zeros (%)	38.8%
Infinite (%)	0.0%	Negatives	0
Memory Size	202.8 KB	Negatives (%)	0.0%



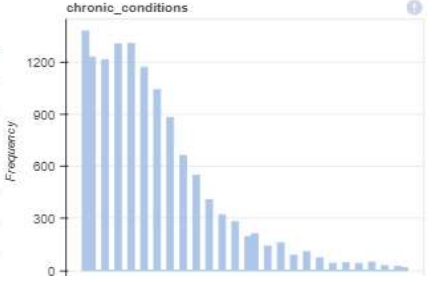
ip_visits
categorical
Show Details

Approximate Distinct Count	3
Approximate Unique (%)	0.0%
Missing	0
Missing (%)	0.0%
Memory Size	836.6 KB



chronic_conditi...
numerical
Show Details

Approximate Distinct Count	27	Mean	6.6124
Approximate Unique (%)	0.2%	Minimum	1
Missing	0	Maximum	27
Missing (%)	0.0%	Zeros	0
Infinite	0	Zeros (%)	0.0%
Infinite (%)	0.0%	Negatives	0
Memory Size	202.8 KB	Negatives (%)	0.0%



DataPrep EDA – Correlation Matrix Heatmap

Data Preparation

Data Preparation = Data Cleansing + Feature Engineering

Data preparation accounts for about 60-80% of the work of data scientists

Data Preprocessing

Drop irrelevant variables and noise variables, biased rows

- Drop the rows where readmit30 flag is 1, but has 0 cost_of_readmission and vice-versa
- Drop the rows corresponding to 'Expired' value in care_plan_following_discharge column
- Remove noise variables (**patient_key_code_number, weekday, month, meds_misc**)
- **APRDRG** is redundant variable of Condition
- **Height** and **Weight** are noise variables because of' bmi' variable
- **'persmentaldis'** has only 3 values of '1', whose 'readmit' values are '0'
- **'cost_of_readmission'** is a redundant variable and highly correlated to 'readmit30'

There are 12544 records in the dataset and 55 attributes after data cleaning

Feature Engineering

- Process of using domain knowledge to create, select and transform the most relevant variables from raw data. Encapsulates various data engineering techniques such as selecting relevant features, handling missing data, encoding the data, and normalizing it.
- F.E. is one of the most important tasks and plays a major role in determining the outcome of a model.
  The 2 types of data in this dataset: numerical features and categorical features.

Numerical Features

Numeric variables can be discrete or continuous. F.E. Techniques: **Binning, Scaling**

Categorical Features

Most Categorical variables are non-numeric data – nominal, ordinal, boolean. Since non-numeric data cannot interpreted by many machine learning and statistical functions, they must be converted to numerical variables, using the techniques **One-hot encoding, Ordinal encoding, Label encoding, Binning**, etc.

Encode the Categorical variables using Label Encoding

'sex', 'marital_status_c', 'ethnic_group_c', 'Condition', 'insurance_provider', 'care_plan_following_discharge'

Encode the Categorical variable 'tobacco user' using Dictionary - Find and Replace

'Quit': 0, 'Never': 1, 'Yes' : 2, 'Not Asked': 0, 'Passive': 3

Encode the numerical to categorical variables using Binning to reduce computation complexity

calcium, temperature, hemoglobin, chronic conditions, vent, meds_neoplastic, meds_biological, chest_tube, ip_visits, LACE_Score, age, bmi, Potassium, Sodium, Creatinine, Respiration, Patient pain score, Pulse, Glucose, bp_systolic, bp_diastolic, ed_visits, los, wbc, meds_cardio_agents, meds_nutrition, meds_central_nervous_system, meds_hematologial, meds_neuromuscular, meds_gastro, meds_infective, meds_anesthetics, meds_endocrine, meds_respiratory, meds_topical, meds_genitourinary

Feature Engineering

Blood calcium: 2 categories: ‘normal’, ‘hyperparathyroidism‘
Temperature: 2 categories: low(<99) or high (>=99)
Hemoglobin: 2 categories: anemic= <12, not anemic >=12
vent: 2 categories: no(=0) or yes (>=1), # Chronic conditions: 2 categories
chest_tube: 2 categories
ip_visits: 2 categories: no(=0) or yes (>=1)
LACE_Score: 2 categories: low risk(<=7), high (> 7)
age: 9 categories: bins = [0, 25, 35, 45, 55, 65, 75, 85, 95, 130]
bmi: 6 categories: 'underweight’, 'normal’, 'overweight’, 'obese I', 'obese II', 'obese III’
Potassium: 3 categories: severe hypokalemia < 2.5, mild hypokalemia (2.5-3.5), normal (3.5-5)
Sodium: 3 categories: normal (>130), low (<=120), very low (<100)
Creatinine rate: 4 categories: normal (<= 1.2), mild (1.2 - 1.5), I high risk (1.5-3), kidney failure (>3)
Respiration rate: 4 categories: low (<12), normal (12-20), high (20-25), very high (>25)
Patient pain score: 4 categories: low(<2), medium(>3 and <5), high (>5 and <8), very high (>8)
Pulse: 5 categories: Normal(<=80), high(>80 and <=100), Very high (>100)
Glucose: 5 categories: Low(<=70), normal(>70 and <=100), high1 (>100 and <= 125), # h2 (> 125 and <=200) h3 (> 200 and <=300), h4 (>=300)
bp_systolic: 5 categories: normal(<=90), high(>90 and <140), Very high (>=140)
bp_diastolic: 3 categories: normal(<=60), high(>60 and <90), Very high (>=90
ed_visits: 5 categories, # los: 8 categories
wbc can be split into 4 categories: low (<=4), normal (4 to 11), high (> 11 and <15), very high (>15)
meds_cardio_agents: 5 ranges, # meds_nutrition: 4 ranges, # meds_central_nervous_system: 5 ranges, # meds_hematologial: 5 ranges
meds_neuromuscular: 3 ranges,# meds_gastro: 7 ranges, # meds_infective: 6 ranges
meds_anesthetics: 6 ranges, # meds_endocrine: 6 ranges, # meds_respiratory: 6 ranges
meds_topical: 6 ranges, # meds_genitourinary: 6 ranges, # meds_neoplastic: 6 ranges, # meds_biological: 6 ranges

After feature engineering, all the data are numerical.

'care_plan_costs', 'cost_of_initial_stay' are numerical variables. All other features are categorical variables.

Log Transform - Skewed Data

Skewness/Kurtosis

Skewness is a measure of symmetry. Skewed data can cause problems with statistic models, as outliers, which often cause skew, can negatively impact a statistical model's performance.

Kurtosis is a measure of whether the data are heavy-tailed or light-tailed relative to a normal distribution. Features with high Skew and Kurtosis tend to have heavy tails, or outliers.

Negative skew: A data set is skewed to the left.

Positive skew: A data set is skewed to the right.

A bell curve with a normal distribution has a skew value of zero.

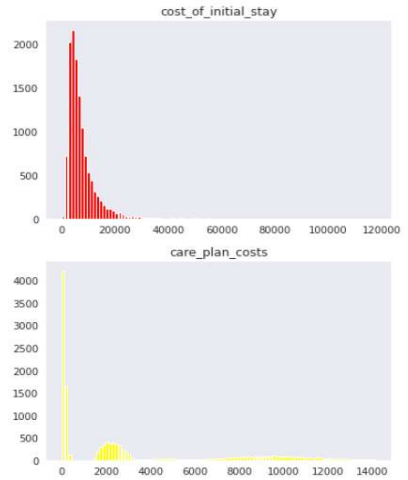
If skewness is less than -1 or greater than 1, the distribution is highly skewed.

If skewness is between -1 and -0.5 or between 0.5 and 1, the distribution is moderately skewed.

If skewness is between -0.5 and 0.5, the distribution is approximately symmetric.

Transformations to handle positively skewed data are

- Square root transformation
- Cube root transformation
- Log transformations
- Box-cox transformation




```
Skew
 care_plan_costs          1.208862
cost_of_initial_stay      4.171310
dtype: float64
Kurtosis
 care_plan_costs          0.053607
cost_of_initial_stay     33.743557
dtype: float64
```

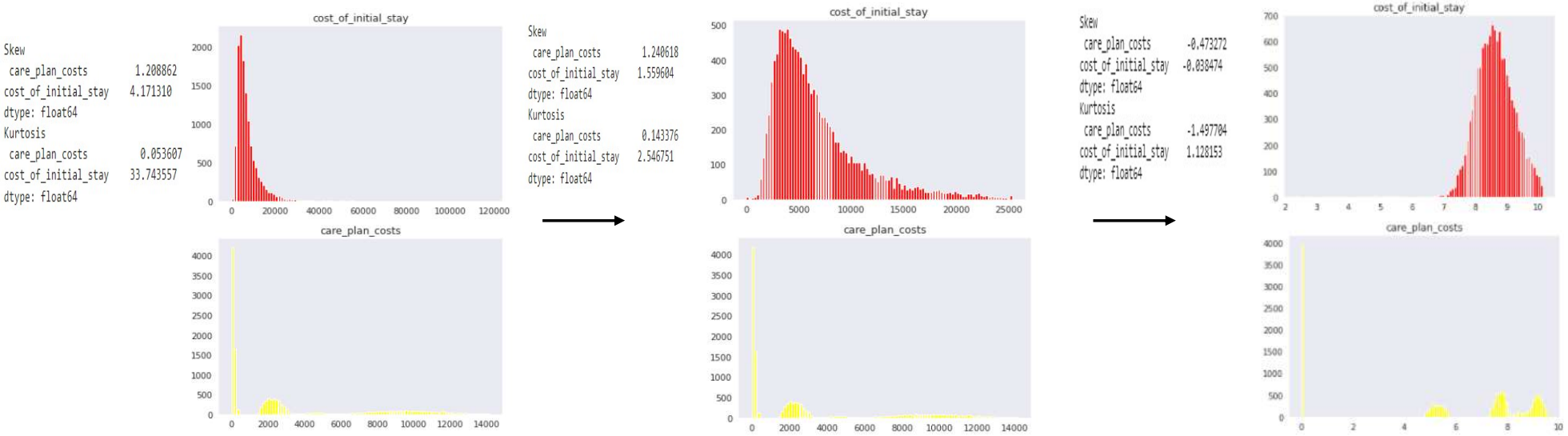
Both numerical variables are highly skewed, so remove outliers and apply log Transform to reduce skew.

Outlier detection

- Outliers are observations in a dataset that don't fit in some way. They are far from the rest of the observations or the center of mass of observations.
- They prohibit the model to estimate the true relationship between variables by introducing bias.
- Outliers can skew statistical measures and data distributions, providing a misleading representation of the underlying data and relationships.
- Removing outliers from training data prior to modeling result in a better fit of the data and better predictions.

Since both numeric variables 'care_plan_costs','cost_of_initial_stay' are highly skewed, there may be possible outliers outside 3 Standard Deviation.

Remove outliers that are outside 3 standard deviation from either side of the mean (0.3%) using z-score.



There are 12332 records in the dataset and 55 attributes after removing outliers.

Correlation

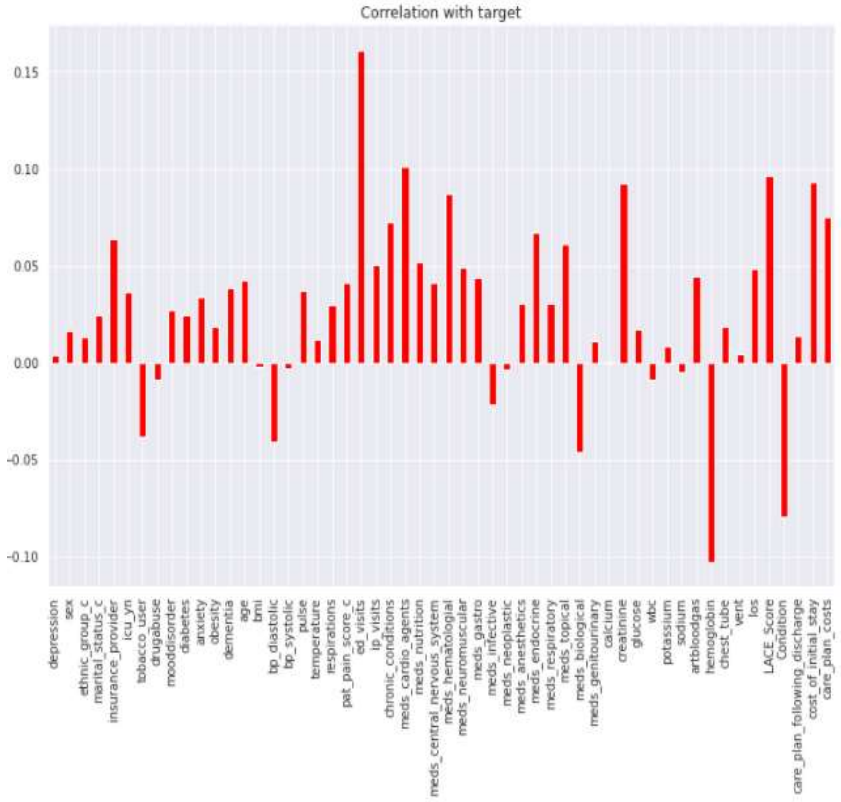
The correlation coefficient **r** measures the strength and direction of a linear relationship between two variables. r is always between +1 (Strong positive) and –1 (Strong negative).

Strong correlation: r > 0.7, Moderate correlation: 0.5 to 0.4 , Weak correlation: r < 0.4

Top 5 features that have strong/moderate linear relationship with **Readmit30**:

ed_visits, hemoglobin, LACE_Score, cost_of_initial_stay, meds_cardio_agents.

All correlations are weak.



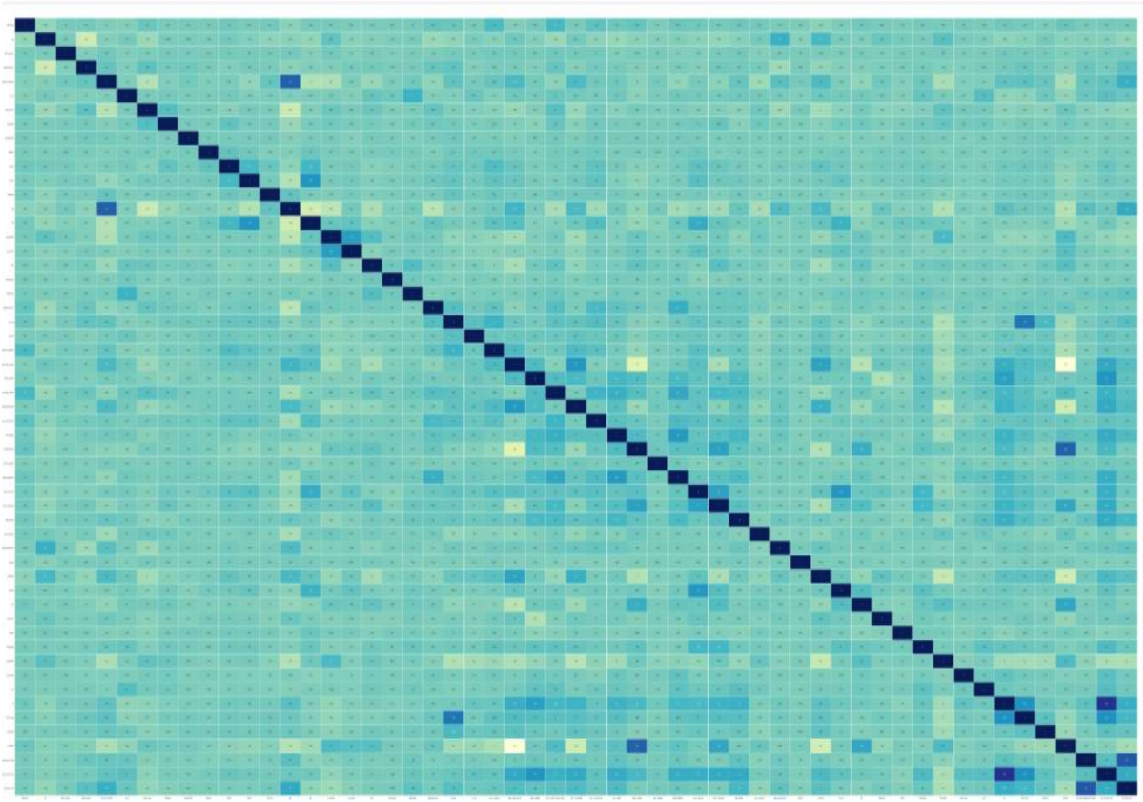
Multicollinearity

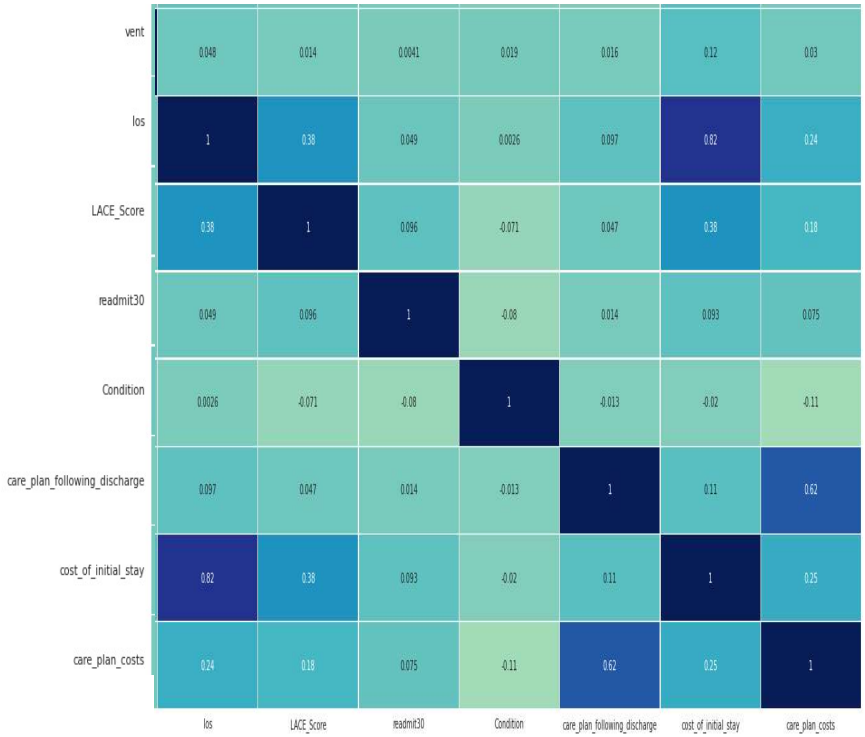
- High multicollinearity (> 95%) results from a linear relationship between 2 independent variables with a high degree of correlation.
- Multicollinearity makes it hard to interpret your coefficients, and it reduces the power of your model to identify independent variables that are statistically significant.

Correlation between Cost of post-discharge care plans and Readmission is 0.075.

Cost of post-discharge care plans do not have a high correlation with Readmission, hence it does not affect Readmission much.

Correlation heat map





Correlation between variables

cost_of_initial_stay	los	0.83
los	cost_of_initial_stay	0.83
care_plan_costs	care_plan_following_discharge	0.62
care_plan_following_discharge	care_plan_costs	0.62
age	insurance_provider	0.59
insurance_provider	age	0.59
meds_infective	Condition	0.58
Condition	meds_infective	0.58
LACE_Score	ed_visits	0.49
ed_visits	LACE_Score	0.49
cost_of_initial_stay	LACE_Score	0.38
LACE_Score	cost_of_initial_stay	0.38
los	LACE_Score	0.38
LACE_Score	los	0.38
meds_gastro	meds_anesthetics	0.34
meds_anesthetics	meds_gastro	0.34
meds_respiratory	meds_endocrine	0.31
meds_endocrine	meds_respiratory	0.31
meds_respiratory	meds_infective	0.30
meds_infective	meds_respiratory	0.30
meds_respiratory	Condition	0.27
Condition	meds_respiratory	0.27
meds_central_nervous_system	meds_anesthetics	0.27
meds_anesthetics	meds_central_nervous_system	0.27
meds_respiratory	cost_of_initial_stay	0.26
cost_of_initial_stay	meds_respiratory	0.26
	meds_topical	0.26
meds_topical	cost_of_initial_stay	0.26
meds_hematologial	cost_of_initial_stay	0.26
cost_of_initial_stay	meds_hematologial	0.26
care_plan_costs	age	0.26
age	care_plan_costs	0.26
Condition	meds_hematologial	0.25
meds_hematologial	Condition	0.25
meds_gastro	cost_of_initial_stay	0.25
cost_of_initial_stay	meds_gastro	0.25
	care_plan_costs	0.25
care_plan_costs	cost_of_initial_stay	0.25
cost_of_initial_stay	meds_anesthetics	0.24
meds_anesthetics	cost_of_initial_stay	0.24
marital_status_c	sex	0.24
sex	marital_status_c	0.24
los	care_plan_costs	0.24
care_plan_costs	los	0.24
meds_endocrine	cost_of_initial_stay	0.23
cost_of_initial_stay	meds_endocrine	0.23
tobacco_user	age	0.23
age	tobacco_user	0.23
insurance_provider	care_plan_costs	0.23
care_plan_costs	insurance_provider	0.23
meds_topical	los	0.23
los	meds_topical	0.23
meds_gastro	meds_central_nervous_system	0.23
meds_central_nervous_system	meds_gastro	0.23
meds_genitourinary	sex	0.23
sex	meds_genitourinary	0.23
creatinine	Condition	0.22
Condition	creatinine	0.22
cost_of_initial_stay	meds_central_nervous_system	0.21
meds_central_nervous_system	cost_of_initial_stay	0.21
meds_hematologial	los	0.21
los	meds_hematologial	0.21
	meds_respiratory	0.21
meds_respiratory	los	0.21
los	meds_gastro	0.21
meds_gastro	los	0.21
chronic_conditions	ed_visits	0.20
ed_visits	chronic_conditions	0.20
meds_central_nervous_system	meds_neuromuscular	0.20
meds_neuromuscular	meds_central_nervous_system	0.20
meds_endocrine	artbloodgas	0.18
artbloodgas	meds_endocrine	0.18
meds_respiratory	artbloodgas	0.18
artbloodgas	meds_respiratory	0.18
age	meds_cardio_agents	0.18
meds_cardio_agents	age	0.18
los	meds_anesthetics	0.18
meds_anesthetics	los	0.18
meds_gastro	meds_topical	0.18
meds_topical	meds_gastro	0.18
care_plan_costs	LACE_Score	0.18
LACE_Score	care_plan_costs	0.18
meds_central_nervous_system	los	0.18
los	meds_central_nervous_system	0.18
meds_central_nervous_system	depression	0.17
depression	meds_central_nervous_system	0.17
anxiety	obesity	0.17
obesity	anxiety	0.17
meds_respiratory	meds_central_nervous_system	0.17
meds_central_nervous_system	meds_respiratory	0.17
care_plan_costs	meds_topical	0.17
meds_topical	care_plan_costs	0.17
meds_infective	cost_of_initial_stay	0.17
cost_of_initial_stay	meds_infective	0.17
meds_endocrine	los	0.17
los	meds_endocrine	0.17
hemoglobin	meds_hematologial	0.16
meds_hematologial	hemoglobin	0.16
ed_visits	readmit30	0.16
readmit30	ed_visits	0.16
meds_infective	los	0.16
los	meds_infective	0.16
meds_hematologial	age	0.16
age	meds_hematologial	0.16
obesity	age	0.16
age	obesity	0.16
Condition	age	0.16
age	Condition	0.16
meds_cardio_agents	insurance_provider	0.16
insurance_provider	meds_cardio_agents	0.16
creatinine	meds_hematologial	0.16
meds_hematologial	creatinine	0.16
tobacco_user	insurance_provider	0.16
insurance_provider	tobacco_user	0.16
depression	chronic_conditions	0.16
chronic_conditions	depression	0.16
age	creatinine	0.16
creatinine	age	0.16
meds_topical	meds_anesthetics	0.15
meds_anesthetics	meds_topical	0.15

Features selection

Feature selection methods are intended to reduce the number of predictor variables to those that are believed to be most useful to a model in order to predict the target variable.

- helps in avoiding the curse of dimensionality.
- simplifies the model, reduces the computational time and complexity of training and testing a classifier
- improve both accuracy and efficiency
- It eliminates irrelevant and noisy features by keeping the ones with minimum redundancy and maximum relevance to the target variable, facilitates an enhanced understanding for the learning model or data, so it results in more cost-effective models.
- It reduces overfitting hence enhance the generalization.
- Feature selection offers a simple yet effective

Feature Selection Methods

1. Univariate Selection – SelectKBest, 2. Feature Importance - Tree Based Classifiers, 3. Correlation Matrix with Heatmap

Feature importance was evaluated based on their abilities to separate two classes using different classifiers.

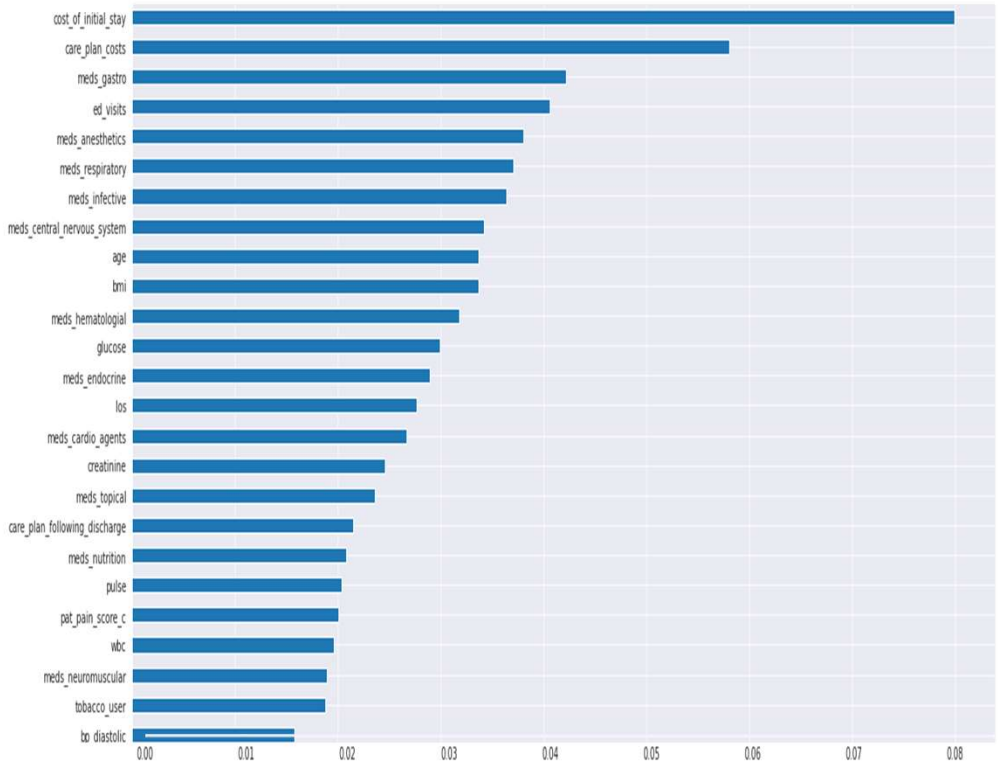
Feature importance gives you a score for each feature, the higher the score more important is the feature towards your output variable. Feature importance is an inbuilt class that comes with Tree Based Classifiers.

17 features have feature importance score < 0.01, remove all the low important features and

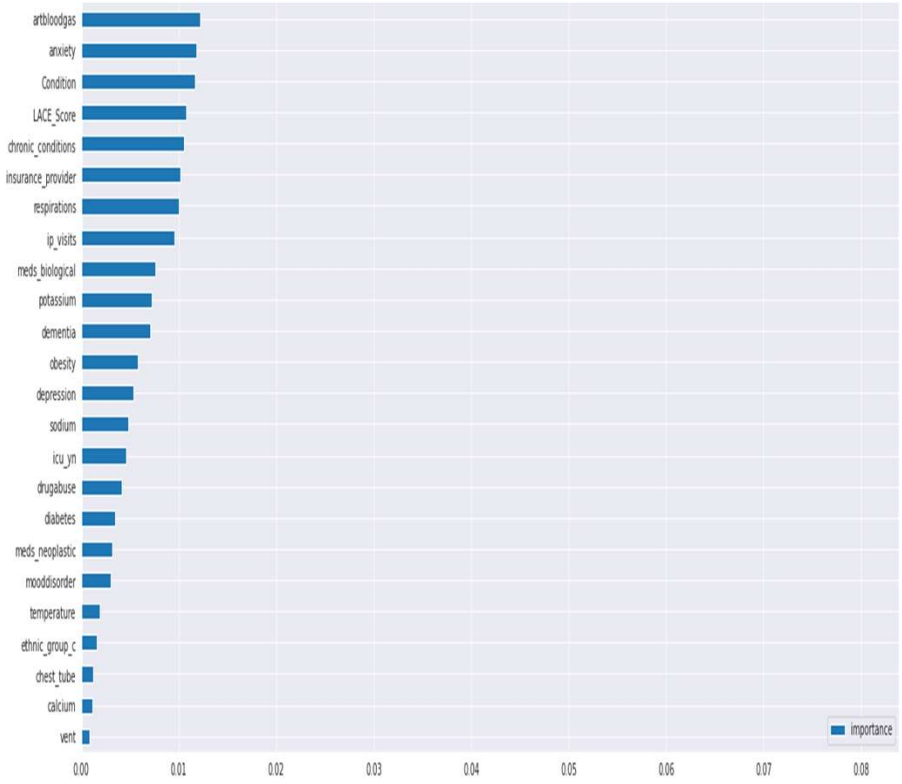
select only top 38 important features, drop all the other features

Features selection

Most important features



Least important features

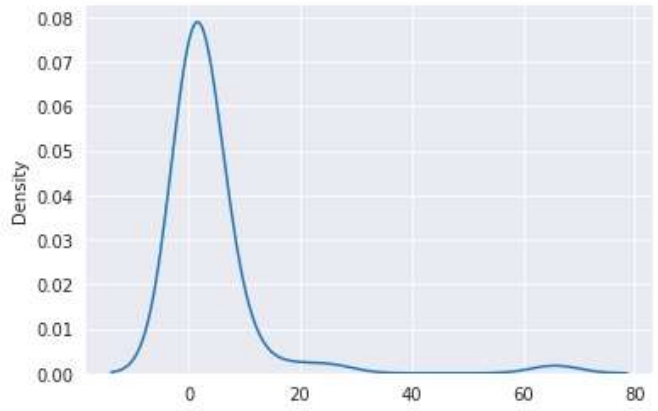


Dataset after Data Preprocessing

Dataset after Data cleaning, Feature engineering and Feature selection has almost Standard normal distribution with mean zero and standard deviation of 1.

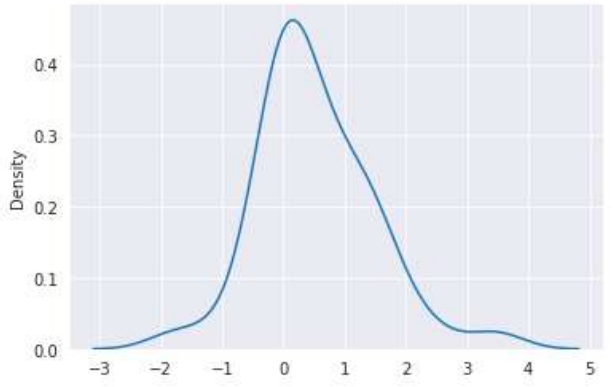
Raw dataset

```
1 sns.distplot(df.skew(),hist=False)
2 plt.show()
```



Dataset after Data preparation

```
1 sns.distplot(df8.skew(),hist=False)
2 plt.show()
```



Standardization and Normalization

Numeric features have different scales. Scaling helps to compare independent features with different ranges/units after converting them to comparable values. There are two major scaling methods:

Normalization is the ideal choice when we know that the distribution of data does not follow a Gaussian distribution or for algorithms that do not assume any data distribution like K-Nearest Neighbors and Neural Networks. (Sklearn - MinMaxScaler)

Standardization can be used when data follows a Gaussian distribution. Z-score is used for Standardization. (Sklearn- StandardScaler)

Once the standardization is done, all the features will have a mean of zero, a standard deviation of one, and thus, the same scale.

These are not strict rules and ideally, we can try both and select the option which gives the best validation results.

Standardization

- Features are scaled so that they have properties of standard normal distribution (mean=0 and std. dev. = 1)
- $z = \frac{(x-\mu)}{\sigma}$ $z = \frac{value - mean}{standard\ deviation}$
- Disadvantage: Data is not bound to a specific range of values

Normalization

- Features are scaled to a fixed range – 0 to 1
- $X_{new} = \frac{X - Xmin}{X_{max} - Xmin}$
- Disadvantage: If there are outliers in the data, they will be scaled to a very small interval

Prediction model

Prediction Model = Data Preparation + Predictive Algorithm + Model training + Model evaluation + Hyperparameters

Predictive Algorithms

- Machine learning Algorithm
- Deep learning

Most used Predictive Analytics Models: Decision trees, Regression (linear and logistic), Neural networks

Others: Ensemble, Classification, Clustering , Forecast, Outliers, Time- series

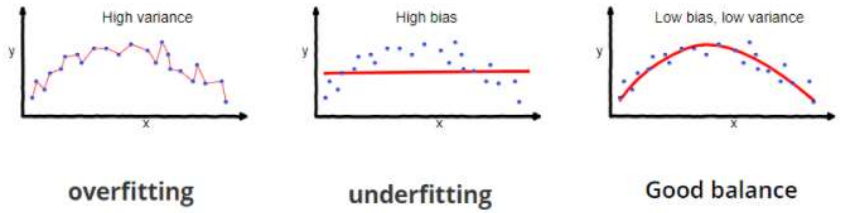
Terminology

Bias - error rate of the training data

Variance - error rate of the testing data

Underfitting – high Bias

Overfitting – low bias and high Variance,



Bias-Variance trade off:

If the model's complexity is too low, (simple model) the model's performance on the training data nor the testing data is not good, therefore it's underfit.

As the complexity of the model increases, the model performs well on the training data but it doesn't perform well on the testing data and it's overfit.

The best prediction model should have low error rate on the training data (**low bias**) and the testing data (**low variance**).

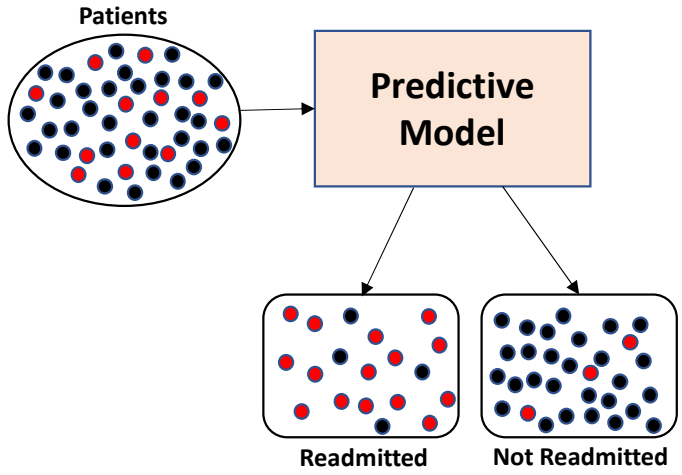
Classification model

Binary classification model predicts if each patient will be readmitted (1) or not readmitted (0). A good prediction model should be able to predict most patients that were readmitted to the hospital within 30 days or not readmitted.

The readmitted class distribution is the target variable. . Binary classification model predicts if each patient will be readmitted (1) or not readmitted (0).

Fit the dataset to various classification models and find out the best fit model among these.

- Linear Algorithms
  - Logistic Regression
  - Linear Discriminant Analysis
  - Naive Bayes
- Nonlinear Algorithms
  - Decision Tree
  - k-Nearest Neighbors
  - Artificial Neural Networks
  - Support Vector Machine
- Ensemble Algorithms (Bagging, Boosting)
  - Random Forest
  - Extra Trees
  - Bootstrap aggregating (Bagging)
  - Stochastic Gradient Boosting
  - AdaBoost Adaptive Boosting (AdaBoost)
  - XGBoost



Performance metrics - Classification model

- **Accuracy** measures the proportion of correctly classified results from the total number of cases.
- **Precision** is the proportion of true results over all positive results.
- **Recall** is the ability of a model to detect all positive results.
- **F-score** is the weighted average of precision and recall, where the ideal F-score value is 1.
- **AUC** (Area Under Curve) measures the area under the curve plotted with true positives on y axis and false positives on x axis.
- **Confusion Matrix:** summary of prediction results. The number of correct and incorrect predictions are summarized with count values and broken down by each class.
- **True Positives (TP)** are the instances in which the model *correctly* predicted a *positive* results.
- **True Negatives (TN)** are the instances in which the model *correctly* predicted a *negative* result.
- **False Positives (FP)** are the instances in which the model *incorrectly* predicted a *positive* result

  FP: Type 1 error
- **False Negatives (FN)** are the instances in which the model *incorrectly* predicted a *negative* result

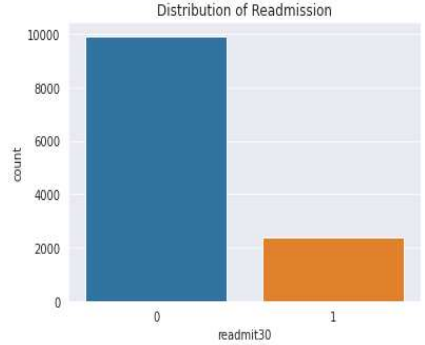
  FN: Type 2 error

Imbalanced dataset

A classification data set with skewed class proportions is called imbalanced. Classes that make up a large proportion of the data set are called majority classes and smaller proportion are minority classes. In this classification model, the target variable Readmit30 is moderately imbalanced. 80.5% were not readmitted, 19.5% were readmitted within 30 days.

Machine Learning Algorithms are usually designed to improve accuracy by reducing the error. They do not consider the class distribution balance of classes. To solve class imbalance problems, sampling techniques are used to create a balanced class distribution.

```
Relative Frequencies of Readmissions
0    0.805
1    0.195
Name: readmit30, dtype: float64
```



Degree of imbalance	Proportion of Minority Class
Mild	20-40% of the data set
Moderate	1-20% of the data set
Extreme	<1% of the data set

Cost-Sensitive Algorithms

Cost-sensitive algorithms are modified versions of machine learning algorithms designed to take the differing costs of misclassification into account when fitting the model on the training dataset. Cost (weight) of misclassification is configured to be inversely proportional to the distribution of classes in the training dataset. Some of the algorithms that can be configured using cost-sensitive training:

Logistic Regression, Decision Trees, Support Vector Machines, Artificial Neural Networks, Bagged Decision Trees, Random Forest, Stochastic Gradient Boosting.

Applying class weights (Keras model): TensorFlow and Keras take class imbalance into account.

Imbalanced dataset

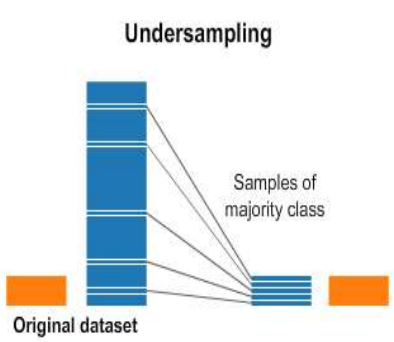
Resampling Techniques

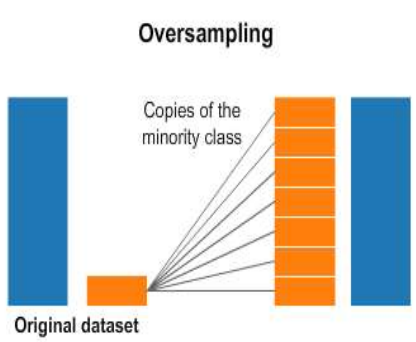
- Random Under-Sampling

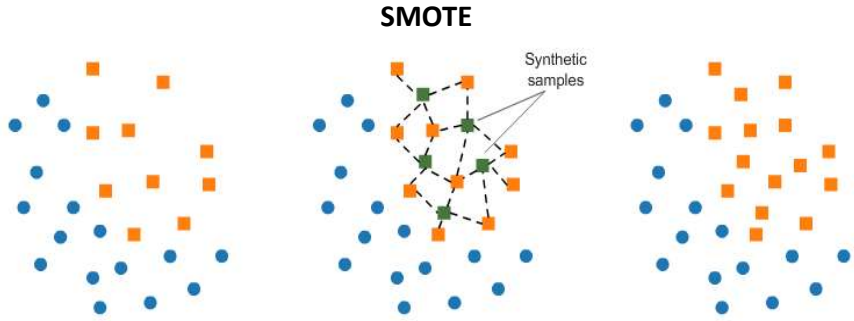
  Condensed Nearest Neighbor, Tomek Links, Edited Nearest Neighbors, Neighborhood Cleaning Rule, One-Sided Selection
- Random Over-Sampling

  Synthetic Minority Over-sampling Technique for imbalanced data (SMOTE), Borderline SMOTE, SVM Smote, K-Means SMOTE, ADASYN
- Combined Oversampling and Undersampling

  SMOTE and Random Undersampling, SMOTE and Tomek Links, SMOTE and Edited Nearest Neighbors
- Cluster-Based Over Sampling
- Adaptive synthetic minority oversampling technique (ADASYN) for imbalanced data







Imbalanced dataset

Aim of this classification model is to reduce Type 2 error (Patients were readmitted (1) but predicted as not reamitted (0)

Scikit-learn package in Python is used for predictive modeling. Scikit-learn contains many built-in algorithms and functions for analyzing the performance of models.

Performance metrics functions from sklearn.metrics:

- confusion_matrix, accuracy_score, recall_score, precision_score, f1_score, roc_curve, roc_auc_score

For imbalanced (target) classification problems, accuracy cannot be used as the performance metric. F1, Recall, Precision, Area Under the Precision-Recall Curve (AUPRC) are more meaningful performance metrics.

		Predicted		
		Readmitted within 30 days (True) - 1	Not Readmitted within 30 days (False) - 0	Recall (Sensitivity) $\frac{TP}{(TP + FN)}$
Actual	Readmitted within 30 days (True) - 1	True Positive	False Positive (Type I error)	Precision (Specificity) $\frac{TN}{(TN + FP)}$
	Not Readmitted within 30 days (False) - 0	False Negative (Type II error)	True Negative	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$

Model evaluation

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses

Train/test split validation

- Train the model on the training set
- Validate the model on the test set

k-fold Cross Validation (k = 10)

1. Split training dataset into *k* folds.
2. The first *k-1* folds are used to train a model, and the holdout *k*th fold is used as the test set.
3. Repeat until all folds is given an opportunity to be used as the holdout test set.
4. A total of *k* models are fit and evaluated, and the performance of the model is calculated as the mean of these runs.

Both can result in misleading results and potentially fail when used on classification problems with a severe class imbalance. Stratify the sampling by the class label, using **stratified train-test split** or **stratified k-fold cross-validation**. **stratified k-fold cross-validation** will enforce the class distribution in each split of the data to match the distribution in the complete training dataset.

The **wider the gap** between the training score and the cross validation score, the more likely that the model is **overfitting (high variance)**.
If the score is low in both training and cross-validation sets this is an indication that the model is **underfitting (high bias)**

Hyperparameter tuning

- Machine learning algorithms have hyperparameters that allows the user to tailor the behavior of the algorithm to a specific dataset.
  Parameters are the configuration variables that is internal to a model and found by the learning algorithm, using the data.
- **Hyperparameter**: Adjustable parameters that must be tuned the developer to obtain a model with optimal performance.
- **Random search**, **Grid search**, and **Bayesian optimization** are some of the strategies for finding hyperparameter values.
- The more hyperparameters of an algorithm that you need to tune, the slower the tuning process. Therefore, it is desirable to select a minimum subset of model hyperparameters to search or tune.
- Not all model hyperparameters are equally important. Some hyperparameters have an outsized effect on the behavior, and in turn, the performance of a machine learning algorithm.

Hyperparameters that are most important for some of the classification machine learning algorithms:

Logistic Regression:
solver in ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga'], **penalty** in ['none', 'l1', 'l2', 'elasticnet'], **C** [100, 10, 1.0, 0.1, 0.01]

Bagged Decision Trees (Bagging):
n_estimators [10, 100, 1000]

Stochastic Gradient Boosting:
learning_rate [0.001, 0.01, 0.1], **n_estimators** [10, 100, 1000]

Random Forest:
max_features [1 to 20], **max_features** in ['sqrt', 'log2'], **n_estimators** in [10, 100, 1000]

Oversampling - Overfitting

Resampling has two drawbacks, when the target class is as highly imbalanced

1. Oversampling the minority class might lead to overfitting, i.e. the model learns patterns that only exist in the particular sample that has been oversampled.
2. Undersampling the majority class might lead to underfitting, i.e. the model fails to capture the general pattern in the data

- Better results could be obtained with more sophisticated resampling, like a combination of under- and oversampling.
- Better results could also be obtained with supervised algorithms other than the decision tree.
- Some machine learning supervised algorithms, such as logistic regression, are less sensitive to class imbalance than decision tree.
- Ensemble models, are more robust to overfitting.
- Better results could be obtained with the decision tree, for example by applying pruning techniques to avoid the overfitting effect or controlling the tree growth.
- Sometimes the data is just not sufficient to describe the minority class.

When upsampling before cross validation, oversampling allows data to leak from the validation folds into the training folds.

The correct way for cross validation when oversampling:

1. First split into training and validation folds. Then, on each fold:
2. Oversample the minority class
3. Train the classifier on the training folds
4. Validate the classifier on the remaining fold

Hyperparameter tuning - Threshold

- Class membership is mapped to a crisp class label using a threshold, such as 0.5, where all values equal or greater than the threshold are mapped to positive class and all other values are mapped to negative class.

Prediction < 0.5 = Class 0

Prediction >= 0.5 = Class 1

- For severe class imbalance problems, the default threshold of 0.5 can result in poor performance. To improve the performance of a classifier that predicts probabilities on an imbalanced classification problem is to tune the threshold used to map probabilities to class labels.
- Threshold for the classifier can be calculated directly using ROC Curves and Precision-Recall Curves or grid search to tune the threshold and locate the optimal value. The default threshold for interpreting probabilities to class labels is 0.5 and tuning this hyperparameter is called threshold moving.

Probabilities to Class Labels

Predict the probability of class membership. Some classification tasks require a crisp class label prediction. This means that even though a probability or scoring of class membership is predicted, it must be converted into a crisp class label.

The decision for converting a predicted probability or scoring into a class label is governed by a parameter referred to as the "*decision threshold*,"

Threshold = Sensitivity + Specificity – 1

Threshold that results in the best balance of precision and recall is the same as optimizing the F-measure that summarizes the harmonic mean of both measures.

F-Measure = (2 * Precision * Recall) / (Precision + Recall)

Find F-measure for each threshold, then locate the score and threshold with the largest value.

Logistic Regression, KNN

After hyperparameter tuning, all the model's performance metrics, ROC AUC and Precison-recall AUC are obtained and compared to select the best model.

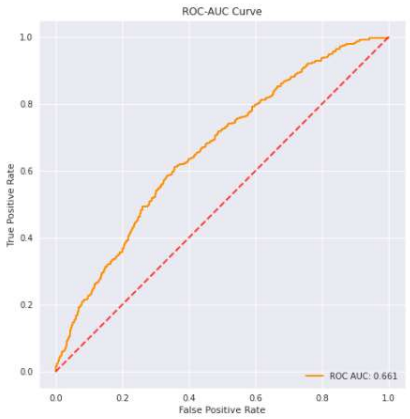
Logistic Regression

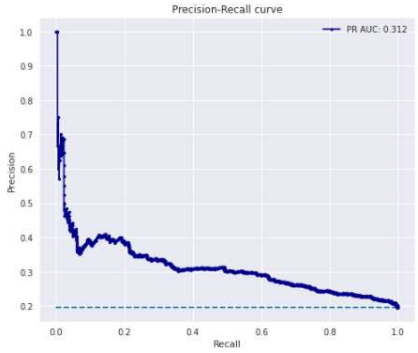
```
Mean ROC AUC: 0.662
Precision-Recall AUC: 0.312
Accuracy: 0.609
Precision: 0.278
Recall: 0.628
F1-score: 0.385
best threshold value: 0.52
```

Classification Report:

	precision	recall	f1-score	support
Not Readmitted	0.87	0.60	0.71	1986
Readmitted	0.28	0.63	0.39	481
accuracy			0.61	2467
macro avg	0.57	0.62	0.55	2467
weighted avg	0.75	0.61	0.65	2467

Classification metrics - Logistic Regression





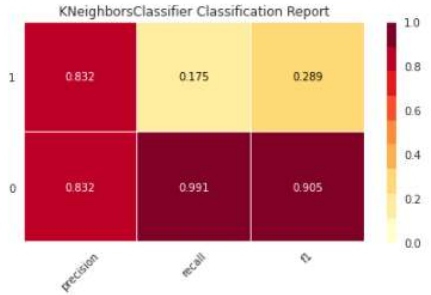
KNN

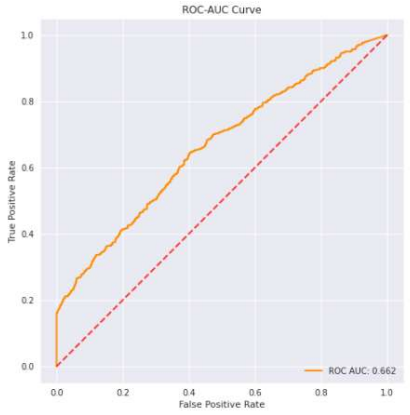
```
Mean ROC AUC: 0.651
Precision-Recall AUC: 0.437
Accuracy: 0.832
Precision: 0.832
Recall: 0.175
F1-score: 0.289
best threshold value: 0.19
```

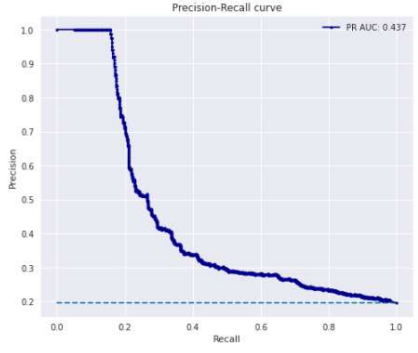
Classification Report:

	precision	recall	f1-score	support
Not Readmitted	0.83	0.99	0.90	1986
Readmitted	0.83	0.17	0.29	481
accuracy			0.83	2467
macro avg	0.83	0.58	0.60	2467
weighted avg	0.83	0.83	0.78	2467

Classification metrics - KNN



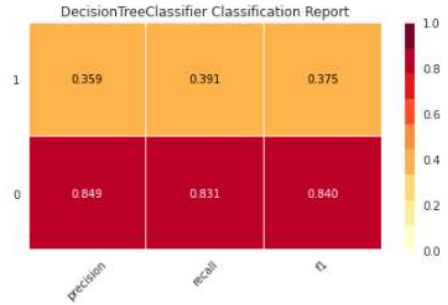


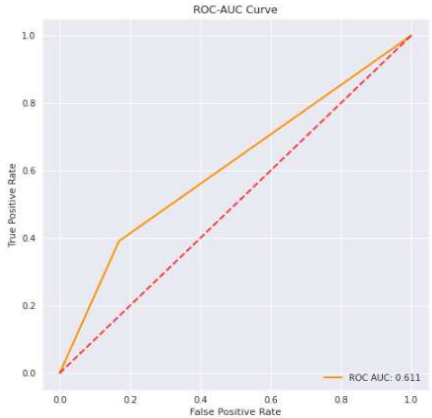


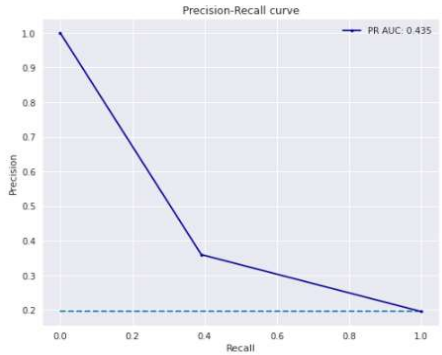
Decision Tree, Random Forest

Decision Tree Classifier

```
Mean ROC AUC: 0.579
Precision-Recall AUC: 0.435
Accuracy: 0.745
Precision: 0.359
Recall: 0.391
F1-score: 0.375
best threshold value: 1.0

----------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   suppor

Not Readmitted       0.85      0.83      0.84      198
    Readmitted       0.36      0.39      0.37       48

      accuracy                           0.75      246
     macro avg       0.60      0.61      0.61      246
  weighted avg       0.75      0.75      0.75      246

----------------------------------------------------------
Classification metrics - Decision Tree Classifier
```

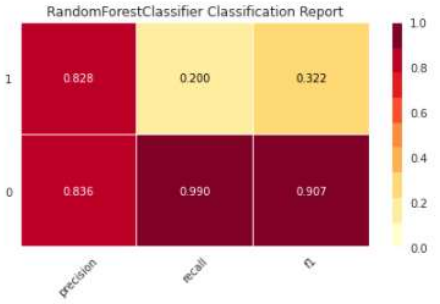


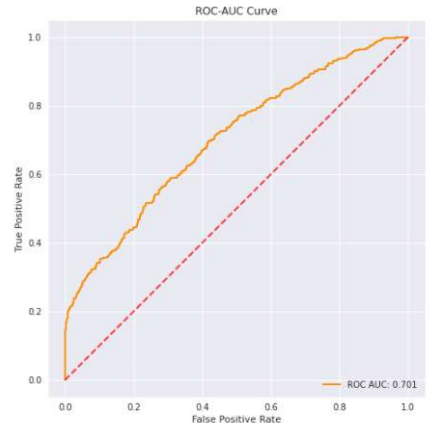


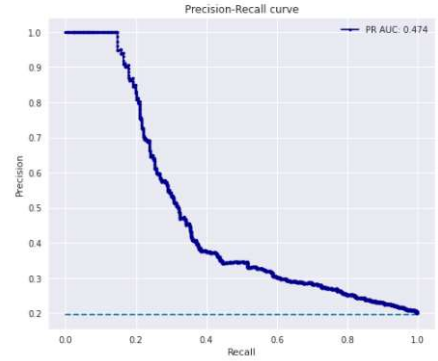


Random Forest Classifier

```
Mean ROC AUC: 0.696
Precision-Recall AUC: 0.474
Accuracy: 0.836
Precision: 0.828
Recall: 0.200
F1-score: 0.322
best threshold value: 0.31

----------------------------------------------------------

 Classification Report:
              precision    recall  f1-score   support

Not Readmitted       0.84      0.99      0.91      1986
    Readmitted       0.83      0.20      0.32       481

      accuracy                           0.84      2467
     macro avg       0.83      0.59      0.61      2467
  weighted avg       0.83      0.84      0.79      2467

----------------------------------------------------------
Classification metrics - Random Forest Classifier
```







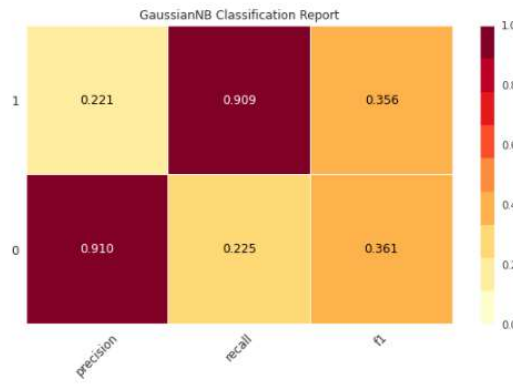
Naive Bayes, Adaboost

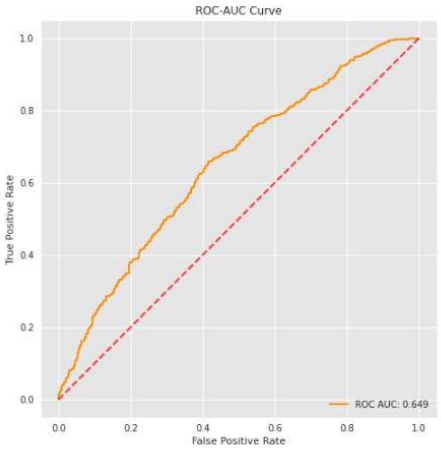
Naive Bayes Classifier

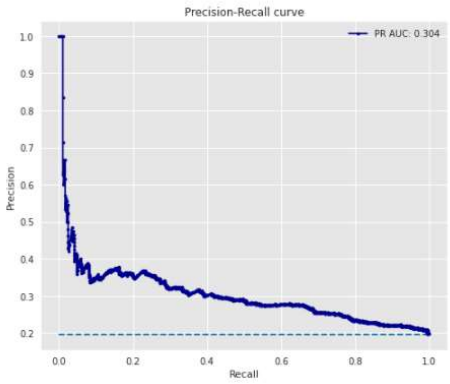
```
Mean ROC AUC: 0.645
Precision-Recall AUC: 0.304
Accuracy: 0.358
Precision: 0.221
Recall: 0.909
F1-score: 0.356
best threshold value: 0.88
```

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

Not Readmitted       0.91      0.23      0.36      1986
    Readmitted       0.22      0.91      0.36       481

      accuracy                           0.36      2467
     macro avg       0.57      0.57      0.36      2467
  weighted avg       0.78      0.36      0.36      2467
```

Classification metrics - Naive Bayes Classifier





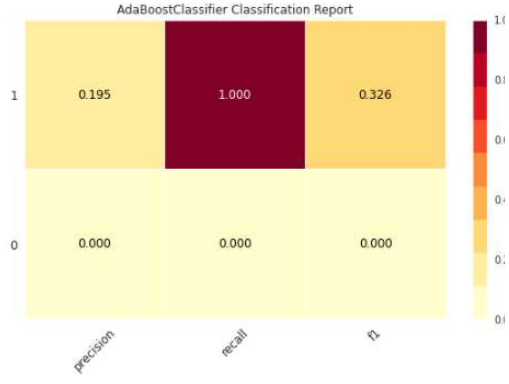


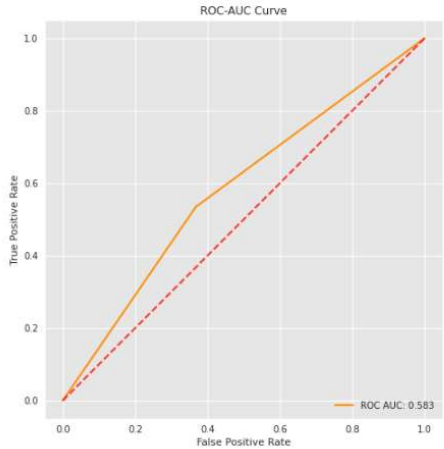
Adaboost Classifier

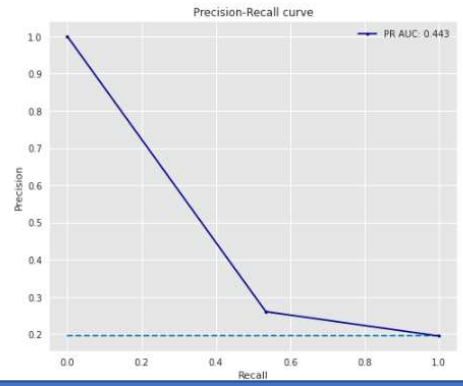
```
Mean ROC AUC: 0.582
Precision-Recall AUC: 0.443
Accuracy: 0.195
Precision: 0.195
Recall: 1.000
F1-score: 0.326
best threshold value: 1.0
```

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

Not Readmitted       0.00      0.00      0.00      1986
    Readmitted       0.19      1.00      0.33       481

      accuracy                           0.19      2467
     macro avg       0.10      0.50      0.16      2467
  weighted avg       0.04      0.19      0.06      2467
```

Classification metrics - AdaBoost Classifier







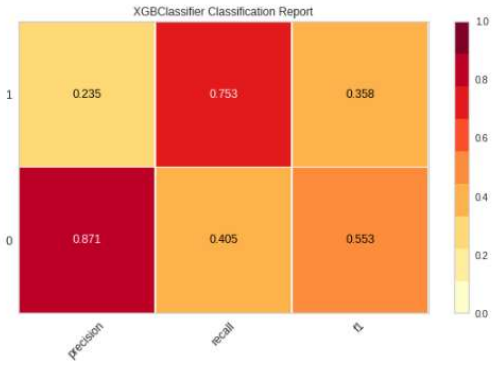
XGB, Gradient Boosting

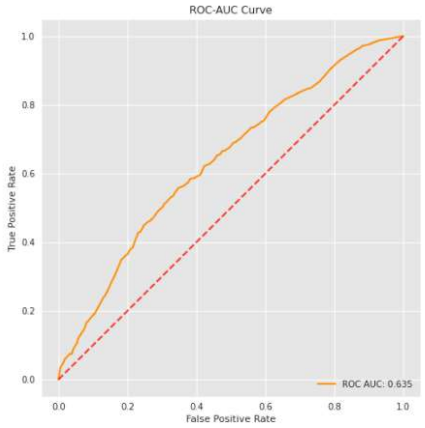
XGB Classifier

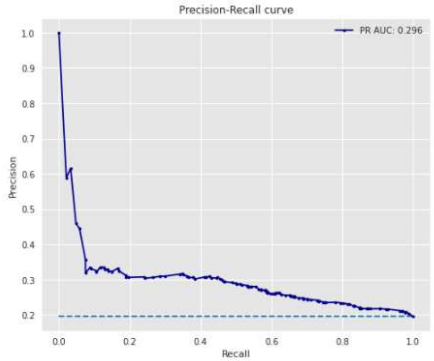
```
Mean ROC AUC: 0.630
Precision-Recall AUC: 0.296
Accuracy: 0.473
Precision: 0.235
Recall: 0.753
F1-score: 0.358
best threshold value: 0.51
```

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

Not Readmitted       0.87      0.41      0.55      1986
    Readmitted       0.23      0.75      0.36       481

      accuracy                           0.47      2467
     macro avg       0.55      0.58      0.46      2467
  weighted avg       0.75      0.47      0.52      2467
```

Classification metrics - XGB





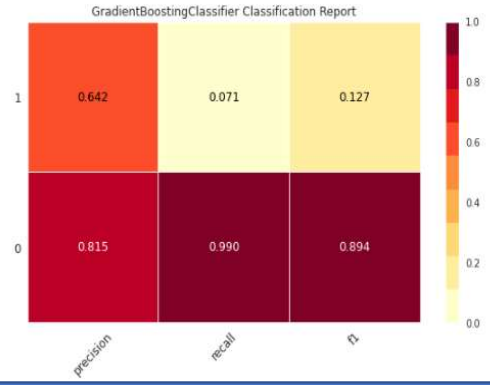


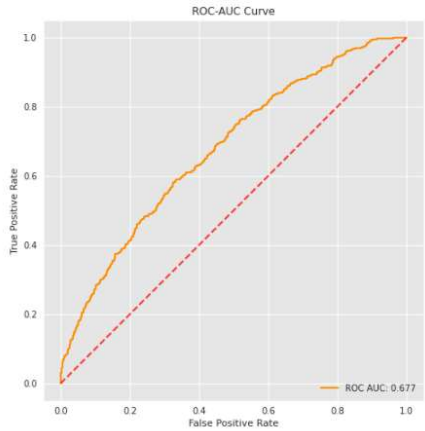
Gradient Boosting Classifier

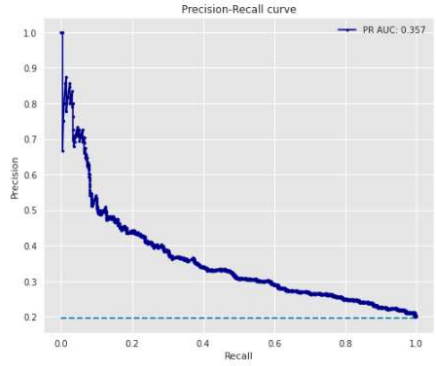
```
Mean ROC AUC: 0.680
Precision-Recall AUC: 0.357
Accuracy: 0.811
Precision: 0.642
Recall: 0.071
F1-score: 0.127
best threshold value: 0.21
```

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

Not Readmitted       0.81      0.99      0.89      1986
    Readmitted       0.64      0.07      0.13       481

      accuracy                           0.81      2467
     macro avg       0.73      0.53      0.51      2467
  weighted avg       0.78      0.81      0.74      2467
```

Classification metrics - Gradient Boosting Classifier



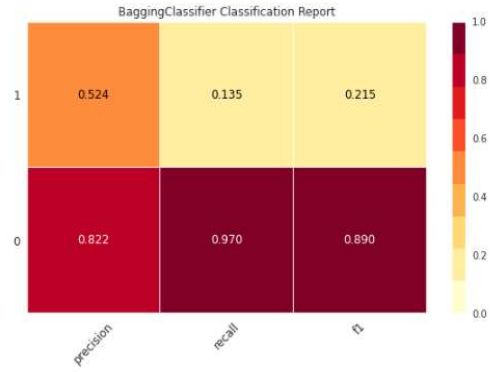


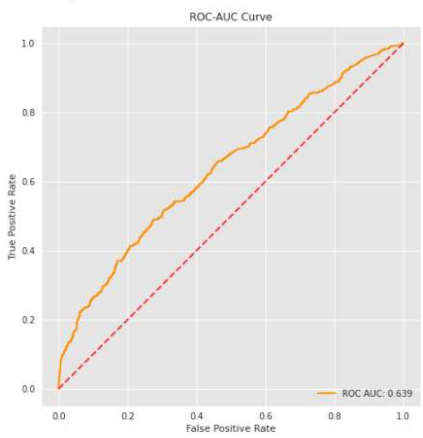


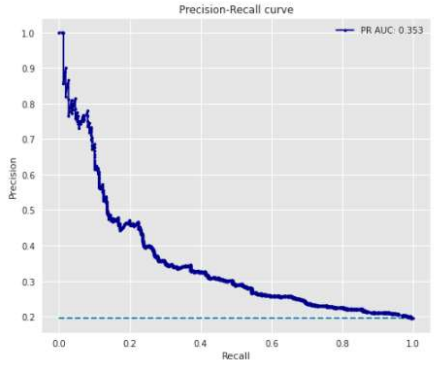
Bagging, Balanced Random Forest

Bagging Classsifer

```
Mean ROC AUC: 0.641
Precision-Recall AUC: 0.353
Accuracy: 0.807
Precision: 0.524
Recall: 0.135
F1-score: 0.215
best threshold value: 0.22

-----------------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   support

Not Readmitted       0.82      0.97      0.89      1986
    Readmitted       0.52      0.14      0.21       481

      accuracy                           0.81      2467
     macro avg       0.67      0.55      0.55      2467
  weighted avg       0.76      0.81      0.76      2467

-----------------------------------------------------------------
Classification metrics - Bagging Classifier
```

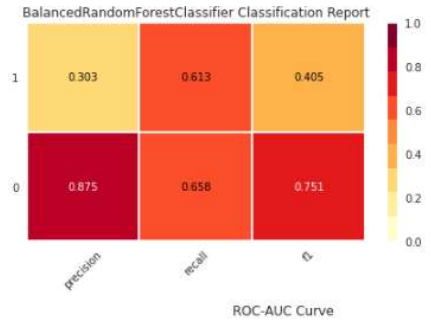


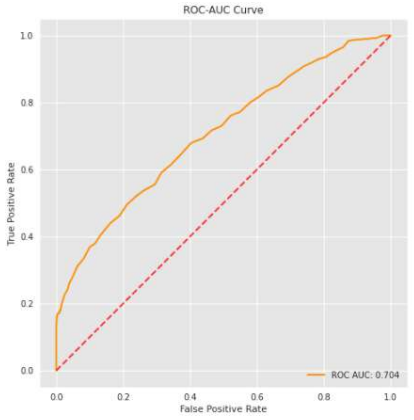


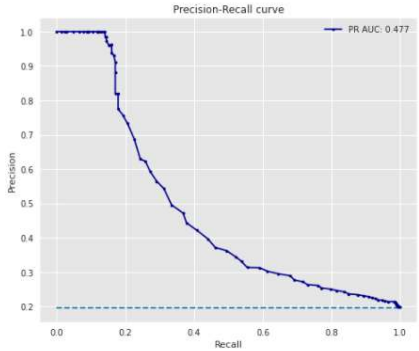


Balanced Random Forest Classifier

```
Mean ROC AUC: 0.694
Precision-Recall AUC: 0.477
Accuracy: 0.649
Precision: 0.303
Recall: 0.613
F1-score: 0.405
best threshold value: 0.49

-----------------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   support

Not Readmitted       0.88      0.66      0.75      1986
    Readmitted       0.30      0.61      0.41       481

      accuracy                           0.65      2467
     macro avg       0.59      0.64      0.58      2467
  weighted avg       0.76      0.65      0.68      2467

-----------------------------------------------------------------
Classification metrics - Balanced Random Forest Classifier
```





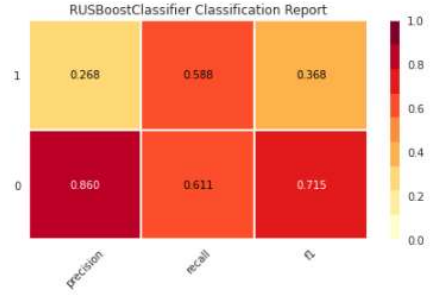


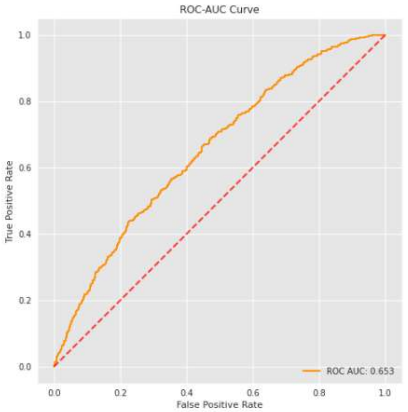
ROC-AUC Curve

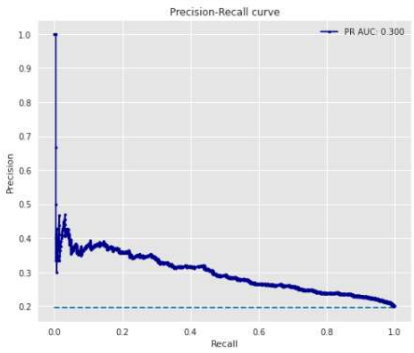
RUS Boost Classifier, Easy Ensemble Classifier

RUS Boost Classifier

```
Mean ROC AUC: 0.648
Precision-Recall AUC: 0.300
Accuracy: 0.607
Precision: 0.268
Recall: 0.588
F1-score: 0.368
best threshold value: 0.5

---------------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   support

Not Readmitted       0.86      0.61      0.71      1986
    Readmitted       0.27      0.59      0.37       481

      accuracy                           0.61      2467
     macro avg       0.56      0.60      0.54      2467
  weighted avg       0.74      0.61      0.65      2467

---------------------------------------------------------------
Classification metrics - Balanced Random Forest Classifier
```

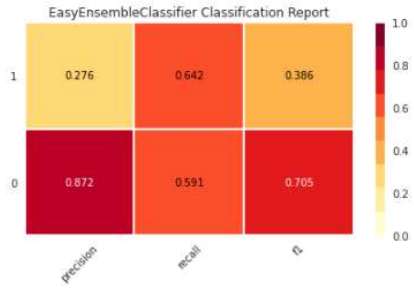


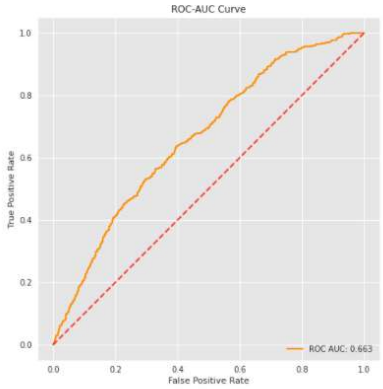


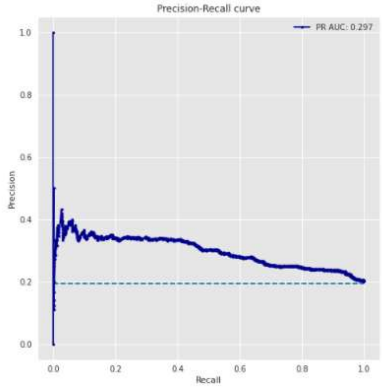


Easy Ensemble Classifier

```
Mean ROC AUC: 0.659
Precision-Recall AUC: 0.297
Accuracy: 0.601
Precision: 0.276
Recall: 0.642
F1-score: 0.386
best threshold value: 0.5

---------------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   support

Not Readmitted       0.87      0.59      0.70      1986
    Readmitted       0.28      0.64      0.39       481

      accuracy                           0.60      2467
     macro avg       0.57      0.62      0.55      2467
  weighted avg       0.76      0.60      0.64      2467

---------------------------------------------------------------
Classification metrics - Easy Ensemble Classifierr
```







Balance Cascade Classifier, Under Bagging Classifier

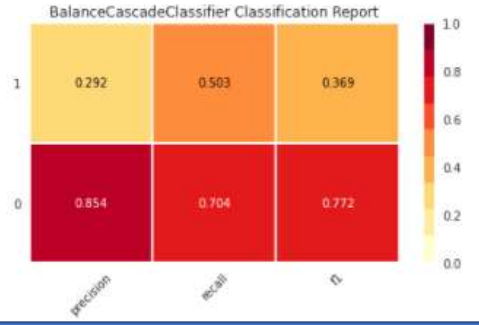
Balanced Cascade Classifier

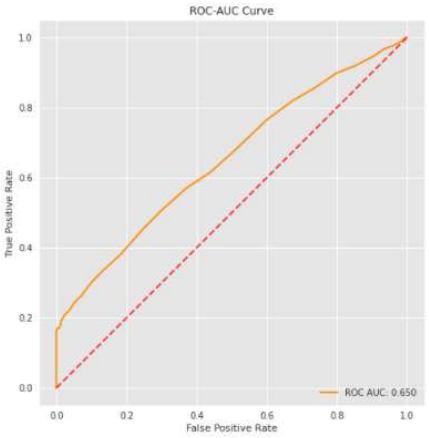
```
Mean ROC AUC: 0.655
Precision-Recall AUC: 0.433
Accuracy: 0.665
Precision: 0.292
Recall: 0.503
F1-score: 0.369
best threshold value: 0.5
```

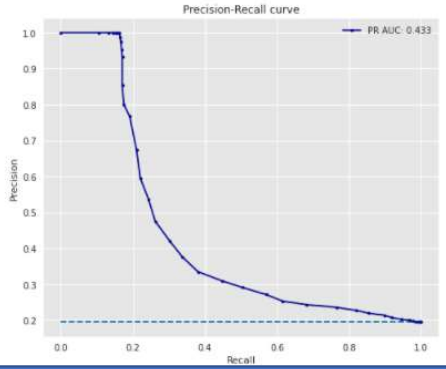
Classification Report:

	precision	recall	f1-score	support
Not Readmitted	0.85	0.70	0.77	1986
Readmitted	0.29	0.50	0.37	481
accuracy			0.66	2467
macro avg	0.57	0.60	0.57	2467
weighted avg	0.74	0.66	0.69	2467

Classification metrics - Balanced Cascade Classifier







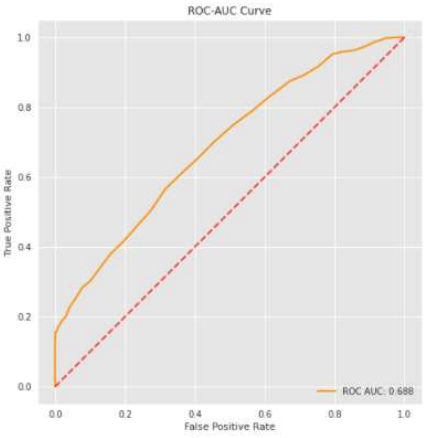
Under Bagging Classifier

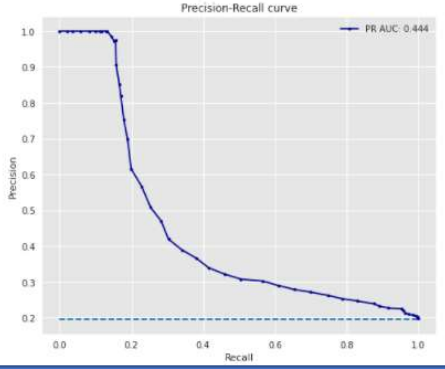
```
Mean ROC AUC: 0.684
Precision-Recall AUC: 0.444
Accuracy: 0.728
Precision: 0.340
Recall: 0.416
F1-score: 0.374
best threshold value: 0.44
```

Classification Report:

	precision	recall	f1-score	support
Not Readmitted	0.85	0.80	0.83	1986
Readmitted	0.34	0.42	0.37	481
accuracy			0.73	2467
macro avg	0.59	0.61	0.60	2467
weighted avg	0.75	0.73	0.74	2467

Classification metrics - Balanced Random Forest Classifie

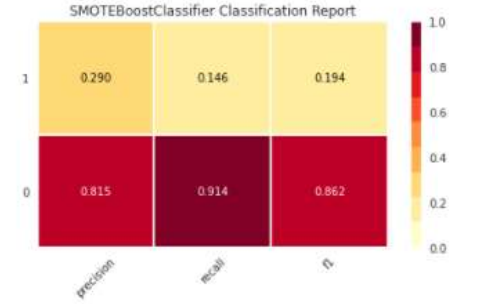


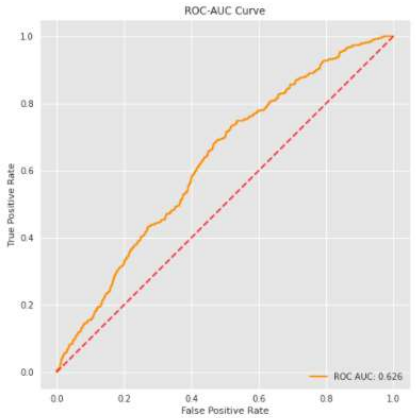


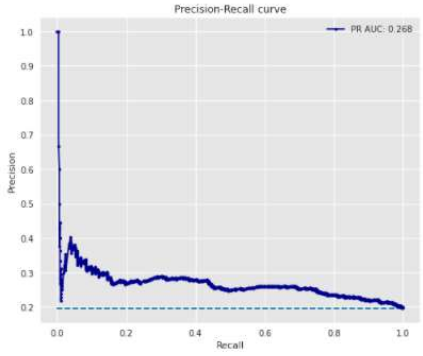
SMOTE Boost Classifier, SMOTE Bagging Classifier

SMOTE Boost Classifier

```
Mean ROC AUC: 0.643
Precision-Recall AUC: 0.268
Accuracy: 0.764
Precision: 0.290
Recall: 0.146
F1-score: 0.194
best threshold value: 0.5

---------------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   support

Not Readmitted       0.82      0.91      0.86      1986
    Readmitted       0.29      0.15      0.19       481

      accuracy                           0.76      2467
     macro avg       0.55      0.53      0.53      2467
  weighted avg       0.71      0.76      0.73      2467

---------------------------------------------------------------
Classification metrics - Balanced Random Forest Classifie
```

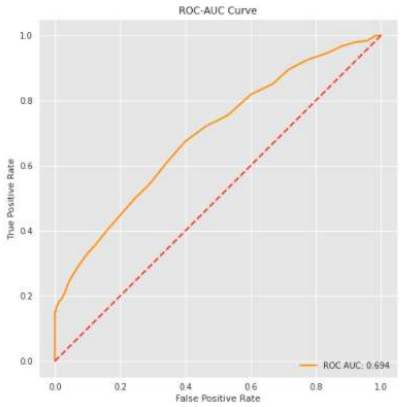


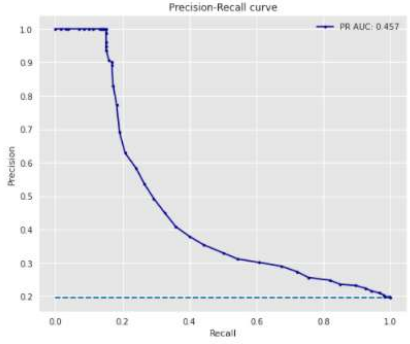




SMOTE Bagging Classifier

```
Mean ROC AUC: 0.678
Precision-Recall AUC: 0.457
Accuracy: 0.832
Precision: 0.830
Recall: 0.173
F1-score: 0.286
best threshold value: 0.24

---------------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   support

Not Readmitted       0.83      0.99      0.90      1986
    Readmitted       0.83      0.17      0.29       481

      accuracy                           0.83      2467
     macro avg       0.83      0.58      0.60      2467
  weighted avg       0.83      0.83      0.78      2467

---------------------------------------------------------------
Classification metrics - Balanced Random Forest Classifier
```

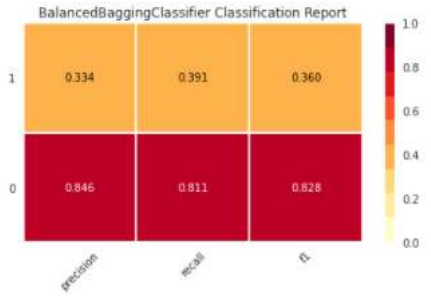


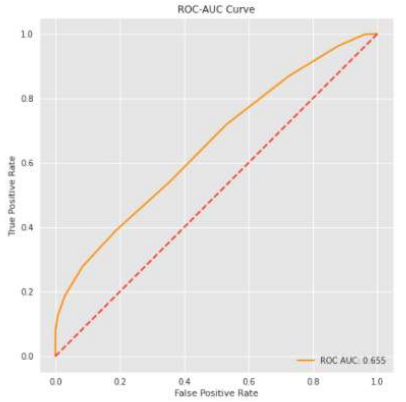


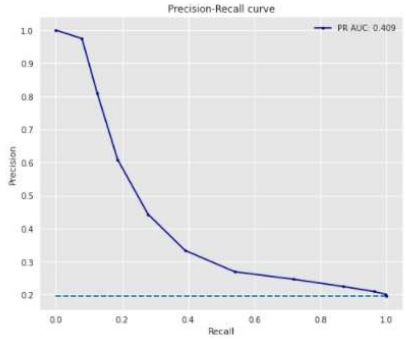
Balanced Bagging Classifier

Balanced Bagging Classifier

```
Mean ROC AUC: 0.650
Precision-Recall AUC: 0.409
Accuracy: 0.729
Precision: 0.334
Recall: 0.391
F1-score: 0.360
best threshold value: 0.5

---------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   support

Not Readmitted       0.85      0.81      0.83      1986
    Readmitted       0.33      0.39      0.36       481

      accuracy                           0.73      2467
     macro avg       0.59      0.60      0.59      2467
  weighted avg       0.75      0.73      0.74      2467

---------------------------------------------------------
Classification metrics - BalancedBaggingClassifier
```







Imbalanced-learn (imblearn) is an open source, MIT-licensed library relying on scikit-learn and provides tools when dealing with classification with imbalanced classes.

Classifiers from imblearn: Balanced Random Forest, RUS Boost, Easy Ensemble, Balanced Bagging, SMOTE Bagging, SMOTE Boost, Balance Cascade, Under Bagging

Balanced Random Forest Classifier has the highest F1 score, with high recall and ROC-AUC and Precision-Recall AUC values.

Models with high F1 score:

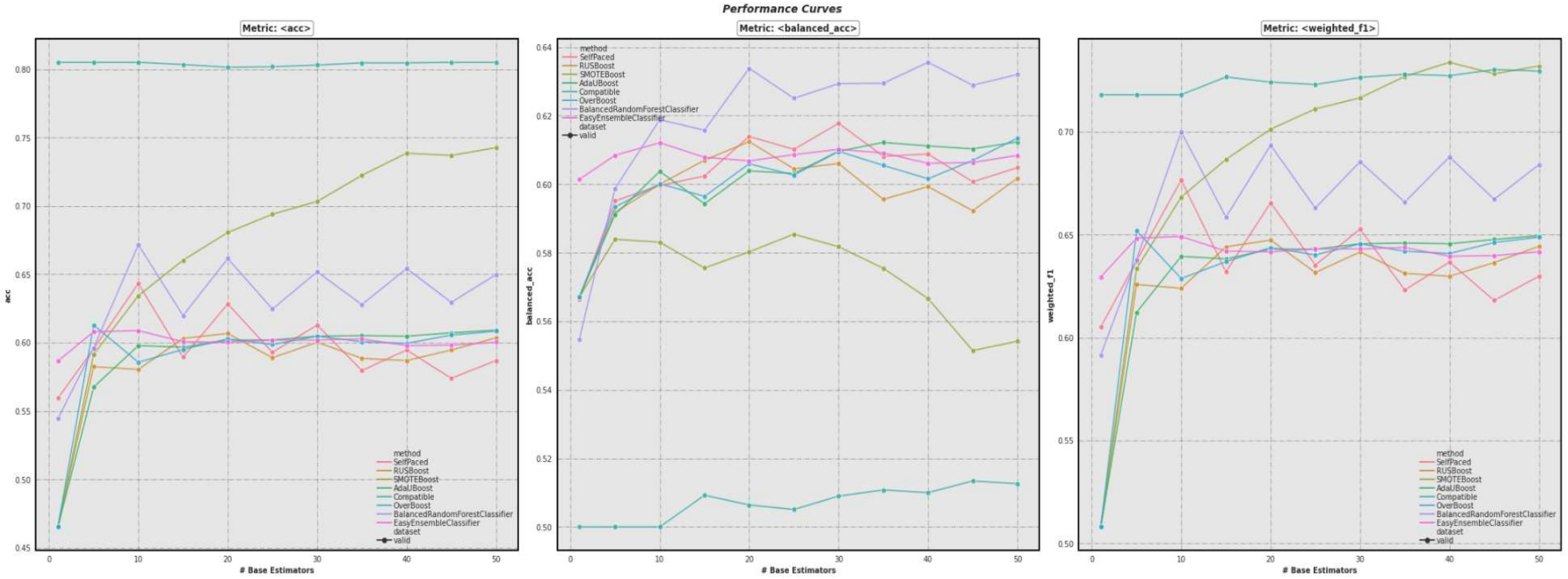
- Logistic Regression
- Decision Tree
- Naive Bayes
- XGB
- Balanced Random Forest
- RUS Boost
- Easy Ensemble
- Balance Cascade
- Under bagging
- Balanced Bagging

Balanced Random Forest

```
Mean ROC AUC: 0.694
Precision-Recall AUC: 0.477
Accuracy: 0.649
Precision: 0.303
Recall: 0.613
F1-score: 0.405
best threshold value: 0.49

---------------------------------------------------------------

 Classification Report:
                precision    recall  f1-score   support

Not Readmitted       0.88      0.66      0.75      1986
    Readmitted       0.30      0.61      0.41       481

      accuracy                           0.65      2467
     macro avg       0.59      0.64      0.58      2467
  weighted avg       0.76      0.65      0.68      2467
```

Performance comparisons

ImbalancedEnsembleVisualizer in **imbalanced-ensemble** (IMBENS) is a Python toolbox for quick implementation, modification, evaluation, and visualization of ensemble learning algorithms for class-imbalanced data. It was built on the basis of scikit-learn and imbalanced-learn. IMBENS includes more than 15 ensemble imbalanced learning (EIL) algorithms, from resampling-based methods to cost-sensitive ensemble learning.



Pycaret, H2O, AutoSklearn

Pycaret

- PyCaret is an open-source, low-code machine learning library in Python that automates machine learning workflows. It is an end-to-end machine learning and model management tool that makes the experiment cycle exponentially fast and efficient.
- PyCaret is essentially a Python wrapper around several machine learning libraries and frameworks such as scikitlearn, XGBoost, LightGBM, CatBoost, spaCy, Optuna, Hyperopt, Ray, and many more.

H2O AutoML

- AutoML is a function in **H2O** that automates the process of building a large number of models, with the goal of finding the "best" model without any prior knowledge or effort by the Data Scientist.
- AutoML (in H2O) trains and cross-validates a default Random Forest, an Extremely-Randomized Forest, a random grid of Gradient Boosting Machines (GBMs), a random grid of Deep Neural Nets, a fixed grid of GLMs, and then trains two Stacked Ensemble models at the end. One ensemble contains all the models (optimized for model performance), and the second ensemble contains just the best performing model from each algorithm class/family (optimized for production use).

Auto-sklearn

- **Auto-sklearn** provides out-of-the-box supervised machine learning.
- Built around the scikit-learn machine learning library, auto-sklearn automatically searches for the right learning algorithm for a new machine learning dataset and optimizes its hyperparameters.

Pycaret (Classification model)

```
1 # Code snippet 27
2 # Compare and train all Classification models to evaluate performance
3 compare_models(budget_time=2) # time limit of 1 min
```

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.8376	0.6982	0.1711	0.9341	0.2885	0.2435	0.3586	2.661
dt	Decision Tree Classifier	0.7223	0.5871	0.3668	0.3130	0.3375	0.1634	0.1643	0.600
svm	SVM - Linear Kernel	0.6340	0.0000	0.4910	0.2630	0.3167	0.1275	0.1471	1.001
lr	Logistic Regression	0.6192	0.6479	0.5840	0.2729	[illegible]	0.1477	0.1697	2.797
ridge	Ridge Classifier	0.6185	0.0000	0.5834	0.2724	0.3713	0.1467	0.1686	0.392
nb	Naive Bayes	0.4930	0.5955	0.6584	0.2246	0.3344	0.0655	0.0890	0.389
knn	K Neighbors Classifier	0.3311	0.6078	0.8968	0.2106	0.3411	0.0414	0.0952	3.952
qda	Quadratic Discriminant Analysis	0.1928	0.4994	0.9988	0.1928	0.3232	-0.0005	-0.0098	0.560

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=100,
                       n_jobs=-1, oob_score=False, random_state=548, verbose=0,
                       warm_start=False)
```

After hyperparameters tuning

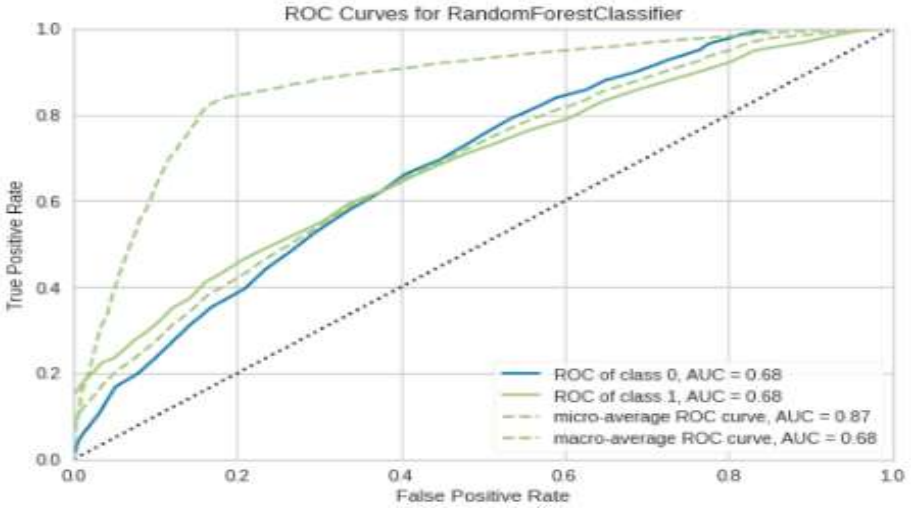
```
1 tuned_model = tune_model(LogisticRegression(),n_iter=50, choose_better=True, optimize='F1')
```

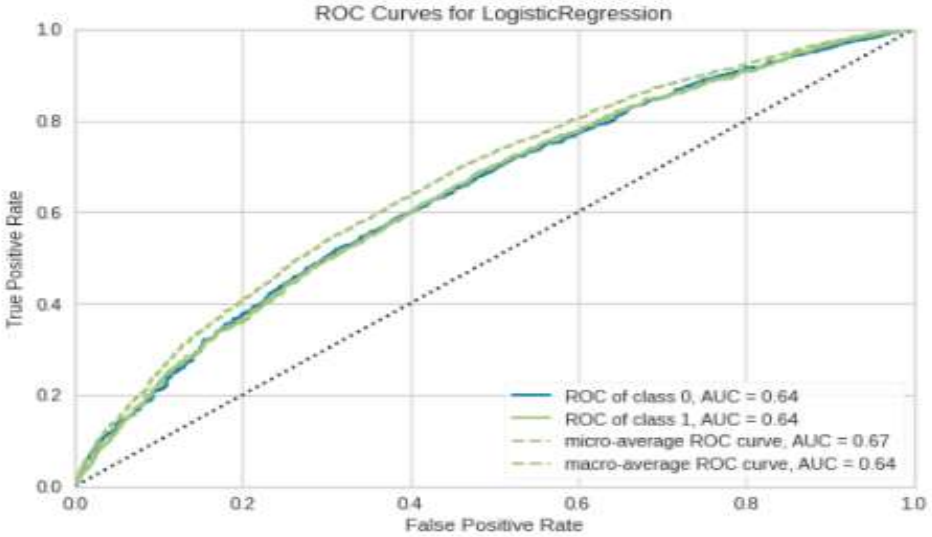
	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC
0	0.6331	0.6501	0.5749	0.2807	0.3772	0.1587	0.1792
1	0.6204	0.6407	0.5988	0.2770	0.3788	0.1556	0.1796
2	0.6211	0.6400	0.5301	0.2611	0.3499	0.1242	0.1397
3	0.6176	0.6422	0.5843	0.2709	0.3702	0.1457	0.1679
4	0.6060	0.6300	0.5422	0.2542	0.3462	0.1142	0.1309
5	0.6199	0.6602	0.5843	0.2725	0.3716	0.1482	0.1703
6	0.6338	0.6830	0.6867	0.3016	0.4191	0.2072	0.2447
7	0.6107	0.6395	0.5868	0.2685	0.3684	0.1401	0.1625
8	0.5933	0.6389	0.5808	0.2566	0.3560	0.1197	0.1410
9	0.6211	0.6630	0.6228	0.2826	0.3888	0.1670	0.1945
Mean	0.6177	0.6488	0.5892	0.2726	0.3726	0.1480	0.1710
SD	0.0115	0.0149	0.0409	0.0133	0.0200	0.0257	0.0311

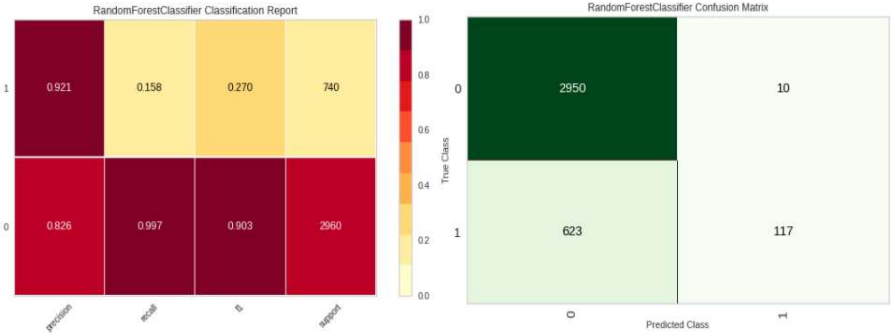
Random Forest Classifier has the highest AUC, but Logistic Regression, Ridge, Decision tree, KNN, Naive Bayes have high f1 value for the minority class.

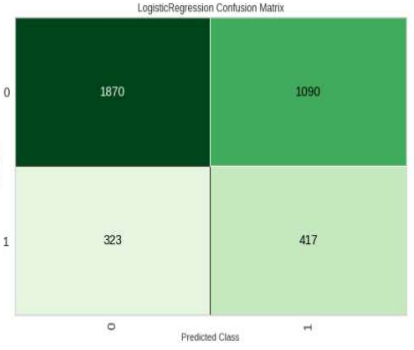
After hyperparameters tuning, Logistic Regression performance metrics show very little improvement.

Pycaret (Classification model)









H2O - Classification model

The current version of AutoML in H2O, trains and cross-validates a default Random Forest, an Extremely-Randomized Forest, a random grid of Gradient Boosting Machines (GBMs), a random grid of Deep Neural Nets, a fixed grid of Generalized Linear Model (GLMs), and then trains two Stacked Ensemble models at the end.

The H2O AutoML interface is designed to have as few parameters as possible so that all the user needs to do is point to their dataset, identify the response column and optionally specify a time constraint or limit on the number of total models trained.

```
ModelMetricsBinomialGLM: stackedensemble
** Reported on validation data. **

MSE: 0.13642734378719534
RMSE: 0.3693607231246919
LogLoss: 0.4375231709739638
Null degrees of freedom: 983
Residual degrees of freedom: 979
Null deviance: 977.0344684467017
Residual deviance: 861.0456004767609
AIC: 871.0456004767609
AUC: 0.7293352023738561
AUCPR: 0.46969909473482924
Gini: 0.45867040474771215

Confusion Matrix (Act/Pred) for max f1 @ threshold = 0.31194269319097
```

	0	1	Error	Rate
0	0	679.0	112.0	0.1416	(112.0/791.0)
1	1	97.0	96.0	0.5026	(97.0/193.0)
2	Total	776.0	208.0	0.2124	(209.0/984.0)

Maximum Metrics: Maximum metrics at their respective thresholds

	metric	threshold	value	idx
0	max f1	0.311943	0.478803	131.0
1	max f2	0.155875	0.586331	278.0
2	max f0point5	0.311943	0.468293	131.0
3	max accuracy	0.499739	0.826220	27.0
4	max precision	0.888551	1.000000	0.0
5	max recall	0.038809	1.000000	395.0
6	max specificity	0.888551	1.000000	0.0
7	max absolute_mcc	0.311943	0.346043	131.0
8	max min_per_class_accuracy	0.227808	0.678756	200.0
9	max mean_per_class_accuracy	0.232678	0.683817	196.0
10	max tns	0.888551	791.000000	0.0
11	max fns	0.888551	191.000000	0.0
12	max fps	0.027855	791.000000	399.0
13	max tps	0.038809	193.000000	395.0
14	max tnr	0.888551	1.000000	0.0
15	max fnr	0.888551	0.989637	0.0
16	max fpr	0.027855	1.000000	399.0
17	max tpr	0.038809	1.000000	395.0

```
3 aml.leaderboard.head(20)
```

model_id	auc	logloss	aucpr
StackedEnsemble_BestOfFamily_2_AutoML_1_20220315_221111	0.729335	0.437523	0.469699
GBM_3_AutoML_1_20220315_221111	0.718426	0.43544	0.475857
StackedEnsemble_BestOfFamily_1_AutoML_1_20220315_221111	0.713984	0.449035	0.405571
GBM_4_AutoML_1_20220315_221111	0.710706	0.437795	0.474564
GBM_1_AutoML_1_20220315_221111	0.708168	0.446541	0.42694
GBM_2_AutoML_1_20220315_221111	0.696875	0.44692	0.434471
DRF_1_AutoML_1_20220315_221111	0.691602	0.823099	0.477722
GLM_1_AutoML_1_20220315_221111	0.686751	0.461593	0.340396
XGBoost_1_AutoML_1_20220315_221111	0.67799	0.470899	0.378785
XGBoost_2_AutoML_1_20220315_221111	0.677718	0.470002	0.417275

Questions?