**Methodology:**

1. **Summarized data/Data Visualization**
2. **PairPlot**
3. Below pair plot is giving pairwise relationships across an entire data frame.
4. Also, it separates diabetic and nondiabetic data with different

colors

1. diabetic =1 is represented by blue
2. nondiabetic =0 is represented by orange

**A picture containing outdoor, open, overhead

Description automatically generated**

**B. Bar Graph**

Below plot gives distribution of outcome as diabetic and nondiabetic, we can infer that about 67% of people are nondiabetic and about 33% are diabetic

**A picture containing logo

Description automatically generated**

**C. Prepare Data**

1. **Data Cleansing**

Missing values:

Data has some missing values so filling them with either the mean or median of the column

Using **fillna()** function

1. **Feature selection:**

**Correlation Matrix**

Below matrix is giving correlation among all the features, we can infer that glucose, BMI, Age, DiabetesPedigreeFunction, and pregnancies are highly correlated to output whereas insulin, skin thickness, and Blood Pressure are less correlated to outcome.

**Since insulin, skin thickness, and Blood Pressure are less correlated to the outcome we can drop these features while doing feature selection**

A screenshot of a computer

Description automatically generated with medium confidence

1. **Feature extension:**
2. **BMI:**

BMI can be categorized into Underweight, Healthy weight, overweight, and Obesity based on BMI values

* Underweight: BMI is less than 18.5
* Healthy weight: BMI is between 18.5 and 24.9
* Overweight: BMI is between 25 and 29.9
* Obesity: BMI is 30 or more

After the division of BMI data based on BMI value, I added a new column to the data frame which is **BMI\_cat**

Also, we can visualize the categorical division of BMI feature by group by for diabetic and nondiabetic people,

**A picture containing table

Description automatically generated**

1. **Blood Pressure**

Similarly, we can subcategorize blood pressure as normal, high blood pressure - stg1, high blood pressure – stg2, and hypertensive crisis based on BP values.

* Normal-> BP<80
* high blood pressure - stg1 -> BP >= 80 and BP<90
* high blood pressure – stg2 -> BP >= 90 and BP<120
* hypertensive crisis -> BP>=120

After the division of Blood Pressure data based on values, I added a new column to the data frame which is **BloodPressure\_cat**

Also, we can visualize the blood pressure categories impact on outcome using groupby function

A picture containing table

Description automatically generated

1. **Glucose:**

We can divide Glucose levels into three categories normal, pre-diabetic, and diabetic.

* Glucose< 140 is normal
* Glucose >=140 and Glucose < 200 is prediabetic
* Glucose >=200 is diabetic

After division, I added a new column as Glucose\_cat

Also, we can visualize the impacts of glucose categories on outcome using groupby

Graphical user interface, text, application

Description automatically generated

1. **Age**

We can divide age into 3 categories as ‘young adults’, 'middle-aged adults', and old adults'

* Young adults: age <31
* middle-aged adults: age >=31 and age <45
* old adults: age >=45

based on categorical division I added a new column to the dataset as Age\_cat.

Also, we can visualize the impact of age categories on outcome using group by,

Graphical user interface, application

Description automatically generated with medium confidence

1. **Pregnancies**

Similarly, we can categorize pregnancies based on whether a particular person had a pregnancy or not.

I created a new column ‘Had pregnancy’ in the dataset and if a woman has pregnancies more than 0 then the value is True otherwise it’s false.

We can visualize the impact of the pregnancy category on outcome using group by function.

A picture containing graphical user interface

Description automatically generated

**Below are the number of categories in each newly created column**

* **BMI\_cat - 4**
* **BloodPressure\_cat - 4**
* **Glucose\_cat - 2**
* **Age\_cat - 3**
* **Had pregnancy – 2**
* Since newly created columns are categorical and not numerical, **we can face NaN(Not a Number)** exceptions for future data processing.
* In order to avoid this, I created a new feature for each category of new columns. For example , for BMI\_cat column I created 4 new features because it has 4 sub-categories. The new features are ‘BMI\_cat\_healthy weight’, ‘BMI\_cat\_obesity’, ‘BMI\_cat\_overweight’ and ‘BMI\_cat\_underweight’. Again, the value of each feature is either 0 or 1, based on whether that category matches to that row or not e.g., ‘BMI\_cat\_obesity’ value will be 1 if that person is categorized as obsessed otherwise it will be 0.
* Again, **after adding sub-categorical numerical features I dropped categorical features** e.g., after adding ‘BMI\_cat\_healthy weight’, ‘BMI\_cat\_obesity’, ‘BMI\_cat\_overweight’ and ‘BMI\_cat\_underweight’ features I dropped BMI\_cat feature because its categorical and sub-categorical features are providing enough information to me.
* Now after feature engineering I get total **20 features**, which are
  + 1. **5 original features :**
       1. Pregnancies, Glucose, BMI, DiabetesPedigreeFunction,Age ( *I dropped 3 original features i.e., insulin, Blood Pressure and Skin Thickness as a part of feature selection)*
    2. **15 New Features:**

I added below 15 new features

1. **4 BMI features:**
2. BMI\_cat\_healthy weight
3. BMI\_cat\_obesity
4. BMI\_cat\_overweight
5. BMI\_cat\_underweight
6. **4 Blood Pressure Feaures:**
7. 'BloodPressure\_cat\_high blood pressure - stg1',
8. 'BloodPressure\_cat\_high blood pressure - stg2',
9. 'BloodPressure\_cat\_hipertensive crisis',
10. 'BloodPressure\_cat\_normal',
11. **2 - glucose features**
12. 'Glucose\_cat\_normal'
13. 'Glucose\_cat\_prediabetes',
14. 3- Age features
15. Age\_cat\_middle aged adults'
16. 'Age\_cat\_old adults'
17. 'Age\_cat\_young adults'
18. 2-Pregnacy features
19. 'Had pregnancy\_False'
20. 'Had pregnancy\_True'

* Correlation matrix relation with all the features

A picture containing text

Description automatically generated

Using above correlation matrix I dropped below less correlated features

1. 'BMI\_cat\_healthy weight'
2. 'BMI\_cat\_overweight',
3. 'BMI\_cat\_underweight'
4. 'BloodPressure\_cat\_high blood pressure - stg1',
5. 'BloodPressure\_cat\_high blood pressure - stg2',
6. 'BloodPressure\_cat\_hipertensive crisis'
7. 'BloodPressure\_cat\_normal'
8. 'Glucose\_cat\_normal',
9. 'Age\_cat\_young adults',
10. 'Had pregnancy\_False',
11. 'Had pregnancy\_True'

**Hence finalized features are,**

1. **'Pregnancies'**
2. **'Glucose'**
3. **'BMI'**
4. **'DiabetesPedigreeFunction'**
5. **'Age',**
6. **'BMI\_cat\_obesity'**
7. **'Glucose\_cat\_prediabetes',**
8. **'Age\_cat\_middle aged adults'**
9. **'Age\_cat\_old adults'**
10. **'Outcome'**
11. **Evaluate Algorithm**

After Feature selection and feature extension, I validated below 11 algorithms with K fold validation.

Below is the accuracy for each algorithm,

LR: 0.765680 (0.044985)

LDA: 0.765732 (0.048088)

KNN: 0.746113 (0.047899)

CART: 0.663221 (0.063092)

NB: 0.739662 (0.048063)

SVM: 0.643601 (0.052315)

AdaBT: 0.742861 (0.047061)

DT: 0.666473 (0.062465)

SVM\_L: 0.762480 (0.046319)

SVM\_S: 0.759201 (0.042785)

XGB: 0.741407 (0.057544)

Chart, box and whisker chart

Description automatically generated

1. **After applying standard scalar,**

S\_LR: 0.772263 (0.053670)

S\_LDA: 0.765732 (0.048088)

S\_KNN: 0.723533 (0.068792)

S\_CART: 0.677869 (0.066381)

S\_NB: 0.739662 (0.048063)

S\_SVM: 0.760894 (0.041846)

S\_AdaBT: 0.741248 (0.048427)

S\_DT: 0.664833 (0.061644)

S\_SVM\_L: 0.764067 (0.044260)

S\_SVM\_S: 0.760894 (0.041846)

S\_XGB\_S: 0.741407 (0.057544)

1. **MinMax Scalar:**

MinMaxLR: 0.770677 (0.047473)

MinMaxLDA: 0.765732 (0.048088)

MinMaxKNN: 0.723453 (0.065694)

MinMaxCART: 0.668086 (0.057173)

MinMaxNB: 0.739662 (0.048063)

MinMaxSVM: 0.736383 (0.032880)

MinMaxAdaBoost: 0.741248 (0.048427)

MinMaxDT: 0.668112 (0.061071)

MinMaxSVM\_linear: 0.751163 (0.050248)

MinMaxSVM\_scale: 0.759281 (0.042213)

MinMaxXGBoost\_scale: 0.741433 (0.059245)

1. **Robust Scalar:**

RobustLR: 0.769064 (0.057590)

RobustLDA: 0.765732 (0.048088)

RobustKNN: 0.738075 (0.055090)

RobustCART: 0.666526 (0.074965)

RobustNB: 0.739662 (0.048063)

RobustSVM: 0.767398 (0.040698)

RobustAdaBoost: 0.741248 (0.048427)

RobustDT: 0.663194 (0.062512)

RobustSVM\_linear: 0.764120 (0.045755)

RobustSVM\_scale: 0.757589 (0.035563)

RobustXGB\_scale: 0.741407 (0.057544)

1. **Improve Accuracy/Tuning**
2. **Knn tuning**

with K parameters and standard scalar , The best k value is 29 between 1-40 odd numbers

Output:

Best: 0.760735 using {'n\_neighbors': 29}

1. **SVM tuning**

* with soft margin and gamma = auto and kernel\_values = ['linear', 'poly', 'rbf', 'sigmoid']

Output:

Best: 0.767398 using {'C': 0.5, 'kernel': 'rbf'}

* SVM tuning with hard margin and gamma = scale and kernel\_values = ['linear', 'poly', 'rbf', 'sigmoid']

Output:

Best: 0.765706 using {'C': 10, 'kernel': 'linear'}

* Tune MinMaxScalar SVM with soft margin and gamma = scale

Output:

Best: 0.769038 using {'C': 1.3, 'kernel': 'linear'}

1. **Finalized Model**

I saved all the both pickle and joblib and tested them with test data. I got below accuracies

Test data results

LR : 0.7792207792207793

LDA : 0.7922077922077922

KNN : 0.7727272727272727

CART : 0.7467532467532467

NB : 0.7987012987012987

SVM : 0.7727272727272727

AdaBT : 0.7857142857142857

DT : 0.7532467532467533

SVM\_L : 0.7792207792207793

SVM\_S : 0.7662337662337663

XGB : 0.7597402597402597