پروژه 3 : پیش بینی ابتلا به دیابت

ایمان حلاج نیا مقدم

بارگذاری و بررسی اولیه داده‌ها

در ابتدا مجموعه داده دیابت از فایل diabetes.csv خوانده و در متغیر diabetes_data ذخیره شد. سپس با استفاده از تابع ()head چند سطر ابتدایی داده‌ها نمایش داده شد تا ساختار کلی دیتاست و ویژگی‌های موجود بررسی شوند :

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

تابع ()info برای مشاهده اطلاعات کلی مجموعه داده به کار رفت. این تابع تعداد نمونه‌ها، نوع داده هر ویژگی و وجود یا عدم وجود مقادیر گمشده را نمایش می‌دهد :

```
<class 'pandas.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

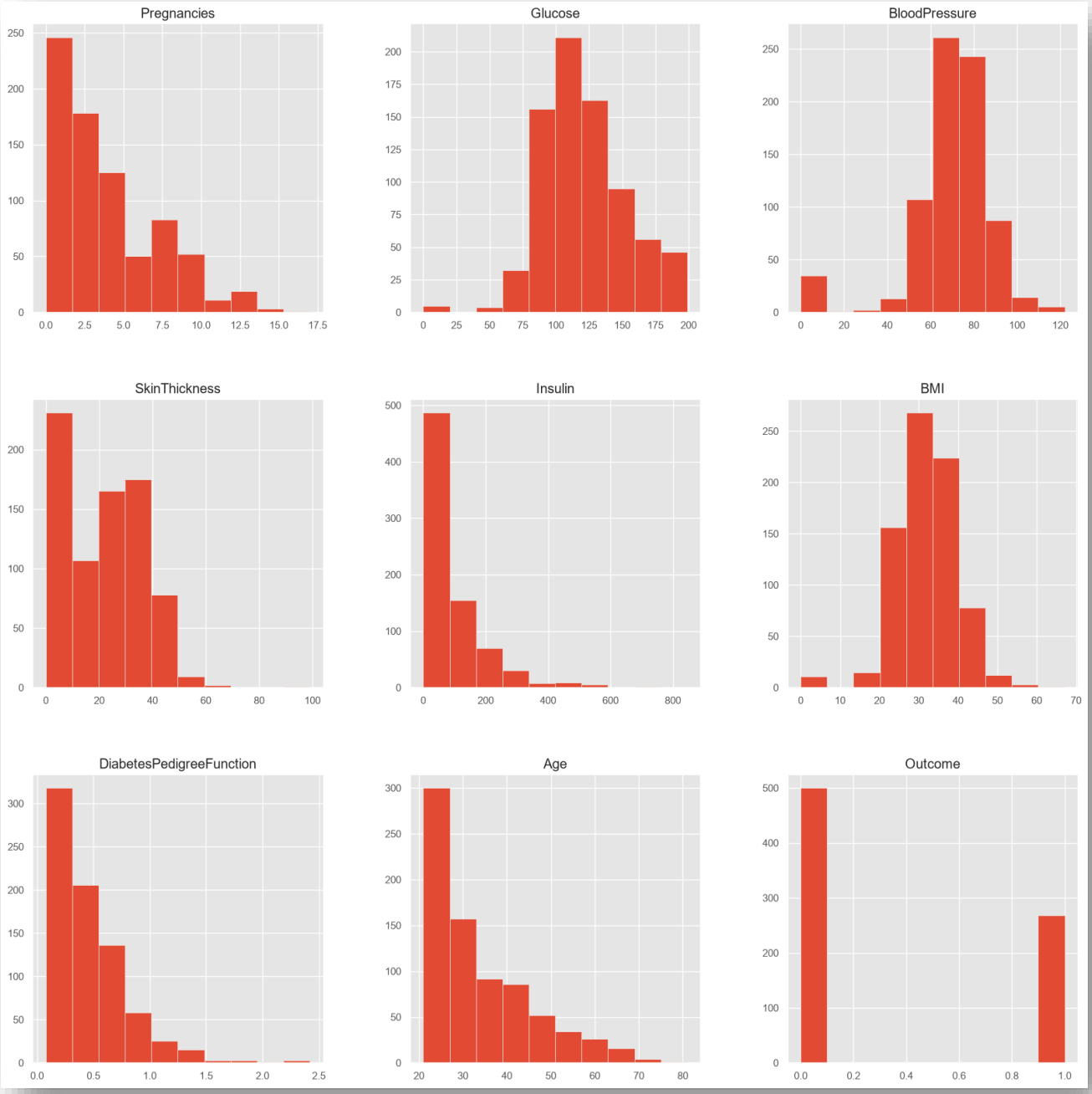
پس از آن، با استفاده از describe().T آمار توصیفی ویژگی‌ها محاسبه شد. این آمار شامل تعداد نمونه‌ها، میانگین، انحراف معیار، حداقل، حداکثر و چارک‌های مختلف برای هر ویژگی است که دید مناسبی از توزیع داده‌ها ارائه می‌دهد :

	count	mean	std	min	25%	50%	75%	max
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.00000	17.00
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.25000	199.00
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.00000	122.00
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.00000	99.00
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.25000	846.00
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.60000	67.10
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.62625	2.42
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.00000	81.00
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.00000	1.00

برای بررسی توزیع متغیر هدف(Outcome) ، درصد نمونه‌های هر کلاس محاسبه شد. این مرحله به منظور ارزیابی میزان تعادل یا عدم تعادل کلاس‌ها در مسئله طبقه‌بندی انجام گرفت :

```
Outcome
0    65.104167
1    34.895833
Name: count, dtype: float64
```

در نهایت، با استفاده از تابع hist() هیستوگرام تمامی ویژگی‌ها رسم شد. این نمودارها به تحلیل توزیع داده‌ها، شناسایی چولگی(Skewness) ، پراکندگی مقادیر و وجود داده‌های پرت احتمالی کمک می‌کنند :


Pregnancies
Glucose
BloodPressure
SkinThickness
Insulin
BMI
DiabetesPedigreeFunction
Age
Outcome

پیش‌پردازش داده‌های گمشده

در این مرحله مقادیر نامعتبر موجود در مجموعه داده شناسایی و اصلاح شدند. در برخی از ویژگی‌ها مانند Glucose، BloodPressure، SkinThickness، Insulin و BMI مقدار صفر از نظر پزشکی منطقی نیست و نشان‌دهنده داده گمشده (Missing Value) است. بنابراین ابتدا یک نسخه کپی از دیتاست ایجاد شد تا داده‌های اصلی بدون تغییر باقی بمانند. سپس مقادیر صفر در ستون‌های مذکور با مقدار NaN جایگزین شدند تا به‌عنوان داده‌های گمشده شناخته شوند. پس از این تبدیل، تعداد مقادیر گمشده در هر ستون با استفاده از تابع isnull().sum() محاسبه و بررسی شد :

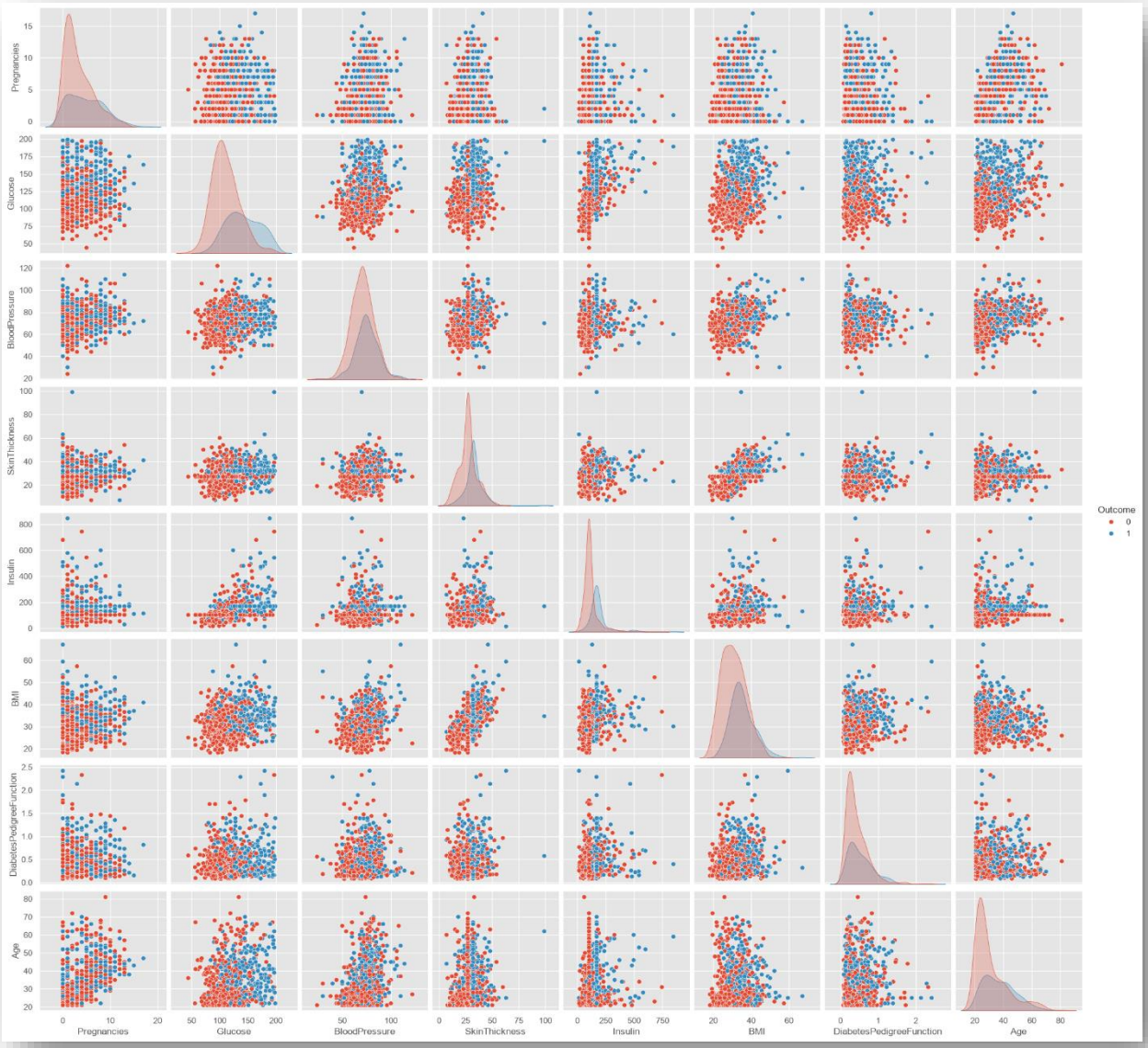
```
Pregnancies                 0
Glucose                     5
BloodPressure              35
SkinThickness             227
Insulin                   374
BMI                        11
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

برای جایگزینی داده‌های گمشده، تابعی به نام median_target() تعریف شد که میانه هر ویژگی را به تفکیک کلاس متغیر هدف (Outcome) محاسبه می‌کند. به عبارت دیگر، برای افراد مبتلا به دیابت (Outcome=1) و افراد غیرمبتلا (Outcome=0) میانه جداگانه‌ای محاسبه می‌شود. در ادامه، تمامی مقادیر گمشده هر ویژگی با میانه همان ویژگی در کلاس مربوطه جایگزین شدند. این روش نسبت به استفاده از یک میانه کلی، اطلاعات بیشتری از توزیع داده‌ها را حفظ می‌کند و باعث می‌شود الگوهای مرتبط با متغیر هدف کمتر دچار تغییر شوند.

در نهایت، مجدداً تعداد مقادیر گمشده بررسی شد تا اطمینان حاصل شود که تمامی داده‌های گمشده با موفقیت جایگزین شده‌اند و مجموعه داده برای مراحل بعدی تحلیل و مدل‌سازی آماده است :

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

پس از تکمیل فرآیند پاک‌سازی داده‌ها و جایگزینی مقادیر گمشده، از نمودار Pair Plot برای تحلیل بصری روابط بین ویژگی‌های مختلف استفاده شد :



در این نمودار:

قطر اصلی (Diagonal) شامل توزیع هر ویژگی به‌صورت هیستوگرام یا نمودار چگالی است و خانه‌های خارج از قطر نمودارهای پراکندگی (Scatter Plot) بین هر دو ویژگی را نمایش می‌دهند.

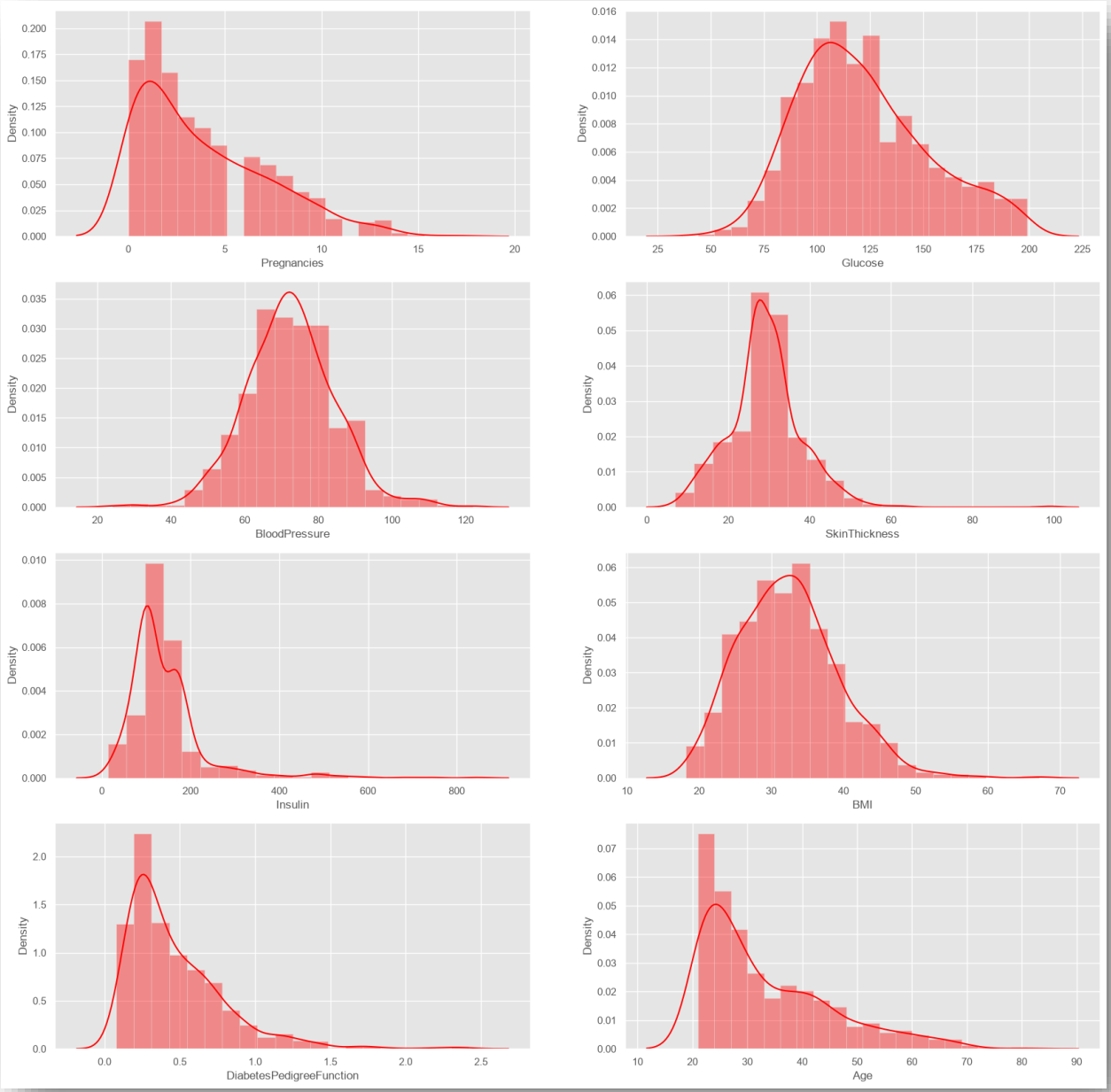
این نمودار امکان بررسی موارد زیر را فراهم می‌کند:

- شناسایی روابط احتمالی بین ویژگی‌ها
- بررسی همبستگی بین متغیرها
- ارزیابی میزان تفکیک‌پذیری کلاس‌های دیابتی و غیردیابتی
- شناسایی الگوها، خوشه‌ها و داده‌های پرت احتمالی

به طور کلی، Pair Plot یک ابزار اکتشافی قدرتمند است که قبل از مرحله مدل‌سازی، دید مناسبی از ساختار داده‌ها و ارتباط میان ویژگی‌ها در اختیار پژوهشگر قرار می‌دهد.

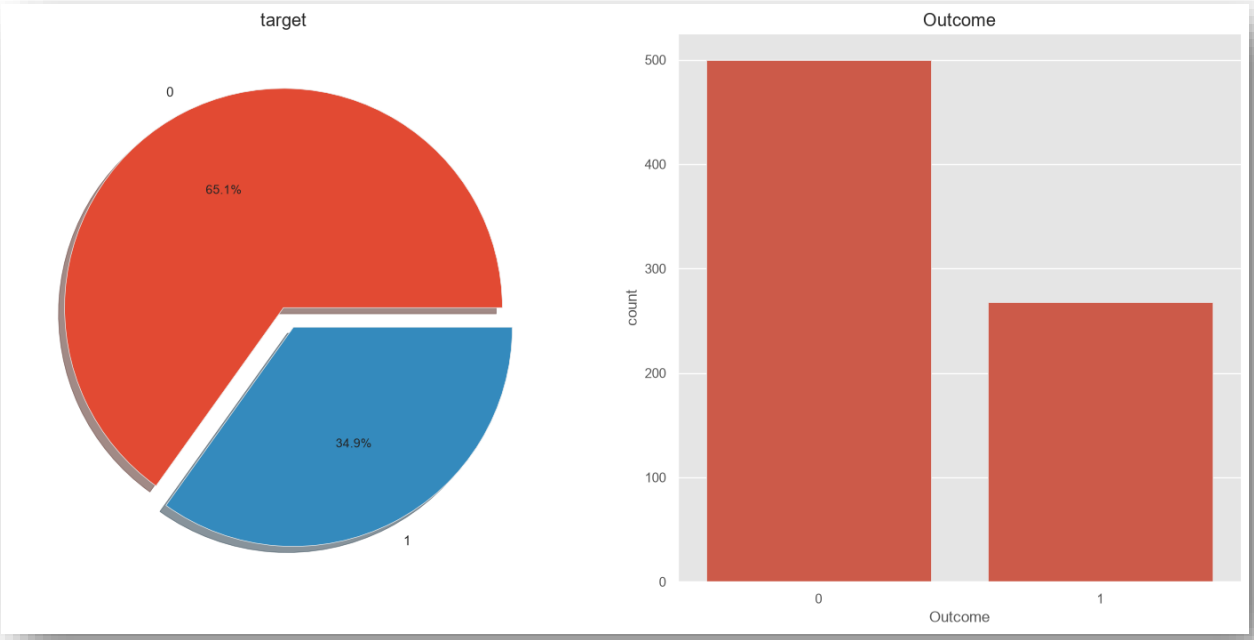
تحلیل اکتشافی داده‌ها(EDA)

پس از پاک‌سازی داده‌ها، تحلیل اکتشافی داده‌ها (Exploratory Data Analysis یاEDA) به منظور بررسی توزیع ویژگی‌ها، توزیع متغیر هدف و روابط بین متغیرها انجام شد :



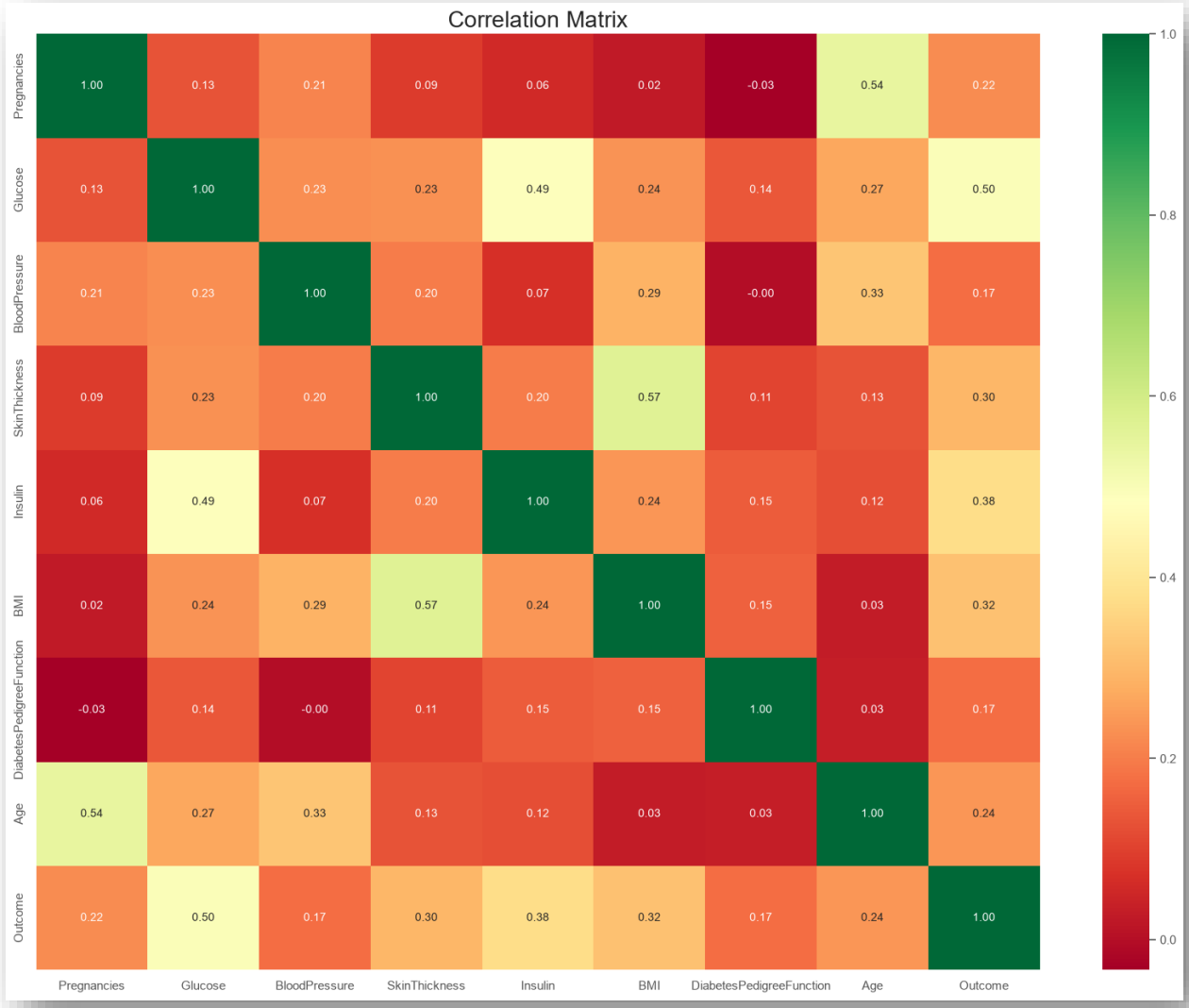
این مرحله کمک می‌کند تا قبل از آموزش مدل، شناخت بهتری از رفتار آماری هر ویژگی به دست آید.

در ادامه، توزیع کلاس‌های متغیر هدف با استفاده از دو نمودار مختلف (دایره ای و ستونی) نمایش داده شد:



بررسی توزیع متغیر هدف اهمیت زیادی دارد؛ زیرا عدم تعادل شدید بین کلاس‌ها می‌تواند بر عملکرد مدل‌های یادگیری ماشین تأثیر منفی بگذارد.

پس از بررسی توزیع داده‌ها، ماتریس همبستگی بین تمامی ویژگی‌ها با استفاده از تابع corr() محاسبه شد :



ضریب همبستگی مقدار عددی بین 1+ و 1- است:

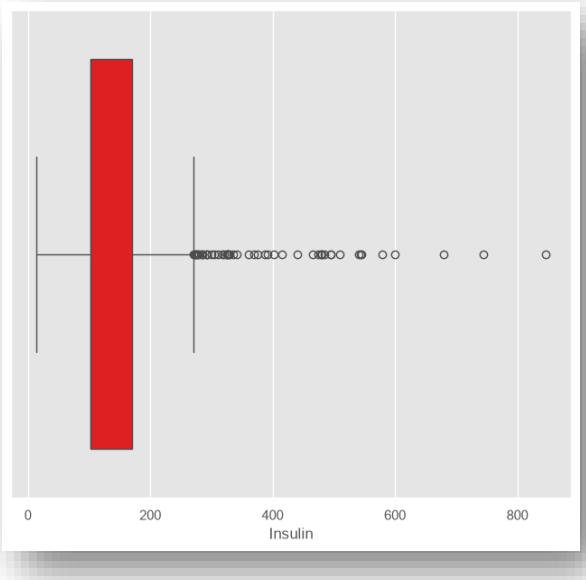
- **1+** : همبستگی مثبت کامل
- **0** : عدم وجود رابطه خطی
- **1-** : همبستگی منفی کامل

این ماتریس نشان می‌دهد که تغییرات هر ویژگی تا چه اندازه با تغییرات سایر ویژگی‌ها مرتبط است.

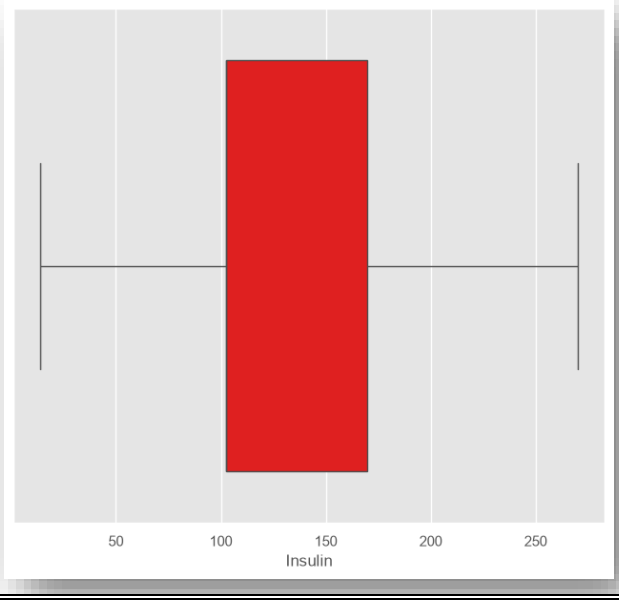
شناسایی و حذف داده‌های پرت

در این مرحله، داده‌های پرت (Outliers) موجود در مجموعه داده شناسایی و مدیریت شدند. ابتدا از روش IQR (Interquartile Range) استفاده شد. در این روش، بازه مجاز مقادیر هر ویژگی بر اساس چارک اول (Q1) و چارک سوم (Q3) محاسبه شده و مقادیری که خارج از این محدوده قرار داشتند به عنوان داده پرت در نظر گرفته شدند. پس از بررسی ویژگی‌ها، مقادیر پرت موجود در ویژگی Insulin شناسایی و مقدار آن‌ها به نزدیک‌ترین حد مجاز جایگزین شد تا تأثیر آن‌ها بر فرآیند یادگیری کاهش یابد .

قبل از حدف داده های پرت :



بعد از حذف داده های پرت :



در ادامه، برای شناسایی دقیق‌تر داده‌های پرت از الگوریتم Local Outlier Factor (LOF) استفاده شد. این الگوریتم با بررسی تراکم محلی داده‌ها، نمونه‌هایی را که رفتار متفاوتی نسبت به همسایگان خود دارند شناسایی می‌کند. پس از محاسبه امتیاز LOF برای تمامی نمونه‌ها، نمونه‌هایی که به عنوان داده پرت تشخیص داده شدند از مجموعه داده حذف گردیدند. این کار باعث افزایش کیفیت داده‌ها و بهبود عملکرد مدل‌های یادگیری ماشین در مراحل بعدی شد.

مهندسی ویژگی(Feature Engineering)

در این مرحله، به منظور افزایش کیفیت داده‌ها و بهبود عملکرد مدل، ویژگی‌های جدیدی از روی ویژگی‌های موجود ایجاد شدند.

ابتدا ویژگی BMI بر اساس دسته‌بندی استاندارد پزشکی به گروه‌های مختلف شامل کم‌وزن، وزن طبیعی، اضافه‌وزن و درجات مختلف چاقی تقسیم شد و یک ویژگی جدید با نام NewBMI ایجاد گردید :

```
# define BMI classification categories as a categorical data type
NewBMI = pd.Series(["Underweight","Normal", "Overweight","Obesity 1", "Obesity 2", "Obesity 3"], dtype = "category")
NewBMI
```

```
0    Underweight
1         Normal
2     Overweight
3      Obesity 1
4      Obesity 2
5      Obesity 3
dtype: category
Categories (6, str): ['Normal', 'Obesity 1', 'Obesity 2', 'Obesity 3', 'Overweight', 'Underweight']
```

سپس وضعیت انسولین به دو گروه طبیعی و غیرطبیعی طبقه‌بندی شد و ویژگی جدیدی با نام NewInsulinScore به مجموعه داده اضافه شد :

```
# if insulin>=16 & insuline<=166->normal
def set_insuline(row):
    if row["Insulin"]>=16 and row["Insulin"]<=166:
        return "Normal"
    else:
        return "Abnormal"
```

همچنین مقادیر قند خون (Glucose) بر اساس بازه‌های استاندارد به چند سطح مختلف دسته‌بندی شدند و ویژگی NewGlucose ایجاد شد :

```
# Some intervals were determined according to the glucose variable and these were assigned categorical variables.
NewGlucose = pd.Series(["Low", "Normal", "Overweight", "Secret", "High"], dtype = "category")
diabetes_data_copy["NewGlucose"] = NewGlucose
diabetes_data_copy.loc[diabetes_data_copy["Glucose"] <= 70, "NewGlucose"] = NewGlucose[0]
diabetes_data_copy.loc[(diabetes_data_copy["Glucose"] > 70) & (diabetes_data_copy["Glucose"] <= 99), "NewGlucose"] = NewGlucose[1]
diabetes_data_copy.loc[(diabetes_data_copy["Glucose"] > 99) & (diabetes_data_copy["Glucose"] <= 126), "NewGlucose"] = NewGlucose[2]
diabetes_data_copy.loc[diabetes_data_copy["Glucose"] > 126 ,"NewGlucose"] = NewGlucose[3]
```

در نهایت، از روش One-Hot Encoding برای تبدیل ویژگی‌های طبقه‌ای ایجادشده به ویژگی‌های عددی استفاده شد تا داده‌ها برای استفاده در الگوریتم‌های یادگیری ماشین آماده شوند. این فرآیند باعث می‌شود مدل بتواند اطلاعات موجود در ویژگی‌های طبقه‌ای را به شکل مؤثرتری مورد استفاده قرار دهد.

آماده‌سازی داده‌ها برای آموزش مدل

در این مرحله، داده‌ها برای آموزش مدل یادگیری ماشین آماده شدند. ابتدا با استفاده از RobustScaler مقادیر ویژگی‌ها نرمال‌سازی شدند تا تأثیر داده‌های پرت کاهش یابد و مقیاس ویژگی‌ها یکنواخت شود.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age
0	0.6	0.775	0.000	1.000000	1.000000	0.177778	0.669707	1.235294
1	-0.4	-0.800	-0.375	0.142857	0.000000	-0.600000	-0.049511	0.117647
2	1.0	1.650	-0.500	0.571429	1.000000	-0.966667	0.786971	0.176471
3	-0.4	-0.700	-0.375	-0.714286	-0.126866	-0.433333	-0.528990	-0.470588
4	-0.6	0.500	-2.000	1.000000	0.977612	1.233333	4.998046	0.235294

سپس مجموعه داده به دو بخش آموزشی (80 درصد) و آزمون (20 درصد) تقسیم شد تا عملکرد مدل روی داده‌های دیده‌نشده ارزیابی شود. در نهایت، با استفاده از StandardScaler ویژگی‌های مجموعه آموزش و آزمون استانداردسازی شدند. این مرحله باعث می‌شود ویژگی‌ها دارای میانگین صفر و انحراف معیار یک باشند که می‌تواند موجب بهبود سرعت همگرایی و عملکرد بسیاری از الگوریتم‌های یادگیری ماشین شود.

آموزش و ارزیابی مدل های مختلف

در این مرحله، مدل **رگرسیون لجستیک (Logistic Regression)** بر روی داده‌های آموزشی آموزش داده شد. پس از آموزش، عملکرد مدل با استفاده از داده‌های آموزش و آزمون ارزیابی شد و دقت (Accuracy) هر دو مجموعه محاسبه گردید.

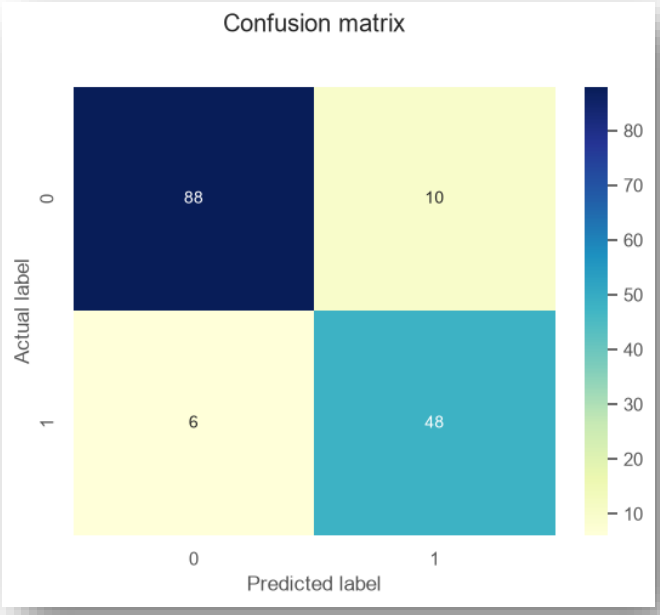
دقت این مدل بر روی داده های آموزش :

```
0.8486842105263158
```

دقت این مدل بر روی داده های تست :

```
0.8947368421052632
```

برای بررسی دقیق‌تر عملکرد مدل، ماتریس درهم‌ریختگی (Confusion Matrix) رسم شد تا تعداد نمونه‌های طبقه‌بندی‌شده به‌صورت صحیح و نادرست در هر کلاس مشخص شود.



همچنین، با استفاده از گزارش طبقه‌بندی (Classification Report) معیارهای ارزیابی شامل Accuracy، Precision، Recall و F1-Score محاسبه شدند. این معیارها دید جامعی از عملکرد مدل در تشخیص افراد مبتلا و غیرمبتلا به دیابت ارائه می‌دهند.

	precision	recall	f1-score	support
0	0.94	0.90	0.92	98
1	0.83	0.89	0.86	54
accuracy			0.89	152
macro avg	0.88	0.89	0.89	152
weighted avg	0.90	0.89	0.90	152

در مرحله ی بعد، مدل **K-Nearest Neighbors (KNN)** با استفاده از داده‌های آموزشی آموزش داده شد. پس از آموزش، عملکرد مدل بر روی داده‌های آموزش و آزمون ارزیابی و مقدار دقت (Accuracy) برای هر دو مجموعه محاسبه شد.

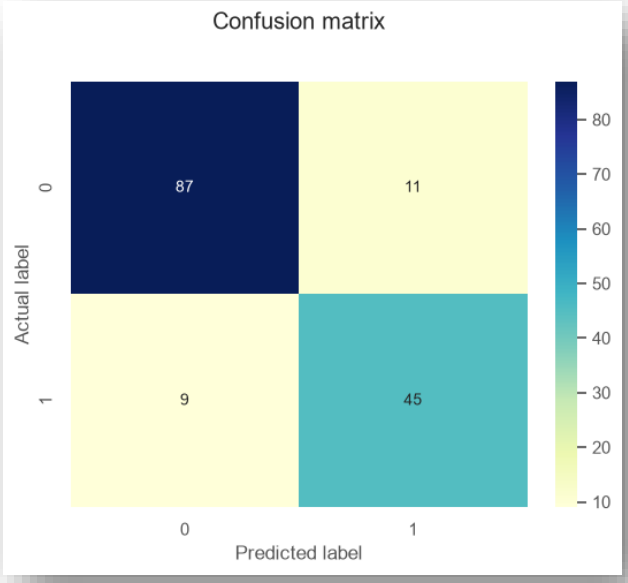
دقت این مدل بر روی داده های آموزش :

```
0.8651315789473685
```

دقت این مدل بر روی داده های تست :

```
0.868421052631579
```

ماتریس درهم ریختگی :



گزارش طبقه بندی :

	precision	recall	f1-score	support
0	0.91	0.89	0.90	98
1	0.80	0.83	0.82	54
accuracy			0.87	152
macro avg	0.85	0.86	0.86	152
weighted avg	0.87	0.87	0.87	152

در این مرحله، مدل **ماشین بردار پشتیبان (Support Vector Machine یا SVM)** ایجاد شد. سپس به منظور دستیابی به بهترین عملکرد، از روش Grid Search برای تنظیم ابرپارامترهای مدل استفاده گردید. در این فرآیند، مقادیر مختلف پارامترهای C و Gamma مورد بررسی قرار گرفتند و با استفاده از جستجوی شبکه‌ای، بهترین ترکیب این پارامترها بر اساس داده‌های آموزشی انتخاب شد. این کار باعث می‌شود مدل با تنظیمات بهینه آموزش دیده و عملکرد بهتری در طبقه‌بندی داده‌ها داشته باشد.

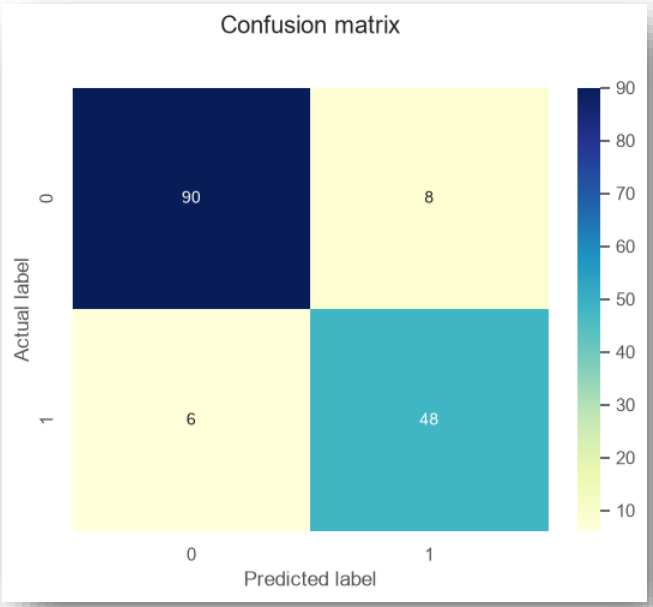
دقت این مدل بر روی داده های آموزش :

```
0.8618421052631579
```

دقت این مدل بر روی داده های تست :

```
0.9078947368421053
```

ماتریس درهم ریختگی :



گزارش طبقه بندی :

	precision	recall	f1-score	support
0	0.94	0.92	0.93	98
1	0.86	0.89	0.87	54
accuracy			0.91	152
macro avg	0.90	0.90	0.90	152
weighted avg	0.91	0.91	0.91	152

مدل بعدی، مدل **درخت تصمیم (Decision Tree)** با استفاده از داده‌های آموزشی آموزش داده شد. پس از پایان آموزش، عملکرد مدل بر روی داده‌های آموزش و آزمون ارزیابی و مقدار دقت (Accuracy) برای هر دو مجموعه محاسبه گردید.

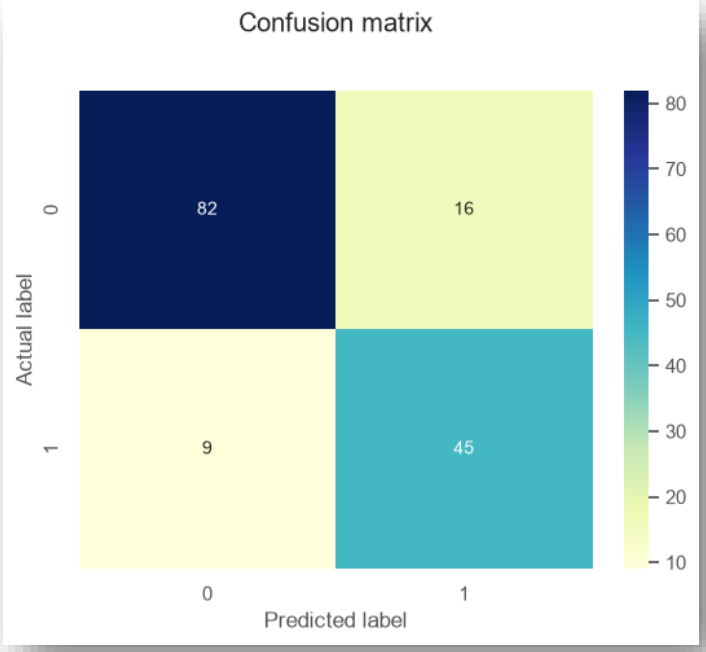
دقت این مدل بر روی داده های آموزش :

```
1.0
```

دقت این مدل بر روی داده های تست :

```
0.8355263157894737
```

ماتریس درهم ریختگی :



گزارش طبقه بندی :

	precision	recall	f1-score	support
0	0.90	0.84	0.87	98
1	0.74	0.83	0.78	54
accuracy			0.84	152
macro avg	0.82	0.84	0.83	152
weighted avg	0.84	0.84	0.84	152

در این مرحله، مدل **جنگل تصادفی (Random Forest)** با استفاده از داده‌های آموزشی آموزش داده شد. برای این مدل، مقادیر مناسبی برای ابرپارامترهایی مانند تعداد درخت‌ها(n_estimators) ، حداکثر عمق درخت (max_depth)، حداقل تعداد نمونه در گره‌ها و تعداد ویژگی‌های مورد استفاده در هر تقسیم تعیین شد تا عملکرد مدل بهبود یابد.

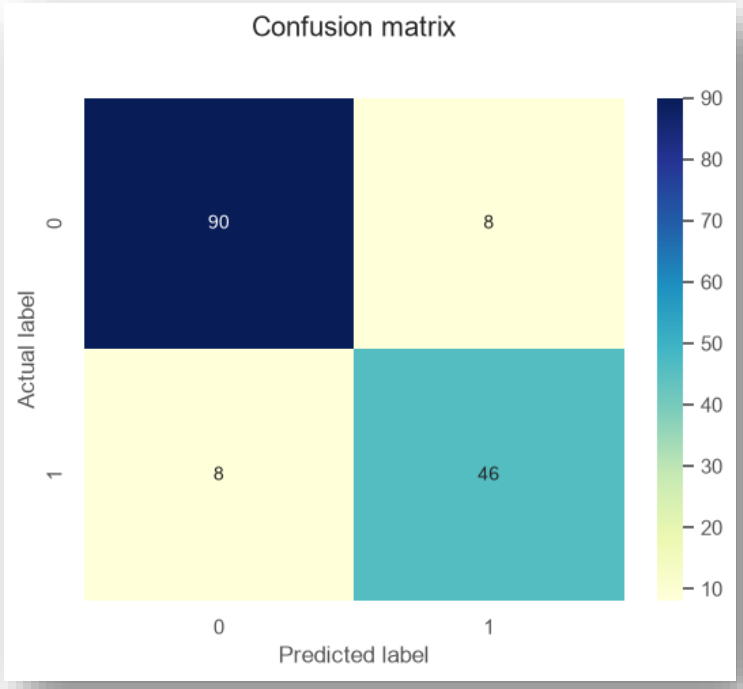
دقت این مدل بر روی داده های آموزش :

```
0.9967105263157895
```

دقت این مدل بر روی داده های تست :

```
0.8947368421052632
```

ماتریس درهم ریختگی :



گزارش طبقه بندی :

	precision	recall	f1-score	support
0	0.92	0.92	0.92	98
1	0.85	0.85	0.85	54
accuracy			0.89	152
macro avg	0.89	0.89	0.89	152
weighted avg	0.89	0.89	0.89	152

مدل بعدی، مدل **گرادیان بوستینگ (Gradient Boosting)** ایجاد شد و به منظور دستیابی به بهترین عملکرد، از روش Grid Search همراه با اعتبارسنجی متقابل ۱۰ بخشی (10-Fold Cross Validation) برای تنظیم ابرپارامترهای مدل استفاده گردید. در این فرآیند، ترکیب‌های مختلفی از پارامترهای تابع خطا (Loss Function)، نرخ یادگیری (Learning Rate) و تعداد درخت‌ها (Number of Estimators) مورد بررسی قرار گرفتند. در نهایت، بهترین ترکیب پارامترها بر اساس نتایج اعتبارسنجی انتخاب شد تا مدل با تنظیمات بهینه آموزش دیده و عملکرد آن با سایر الگوریتم‌های یادگیری ماشین مقایسه شود .

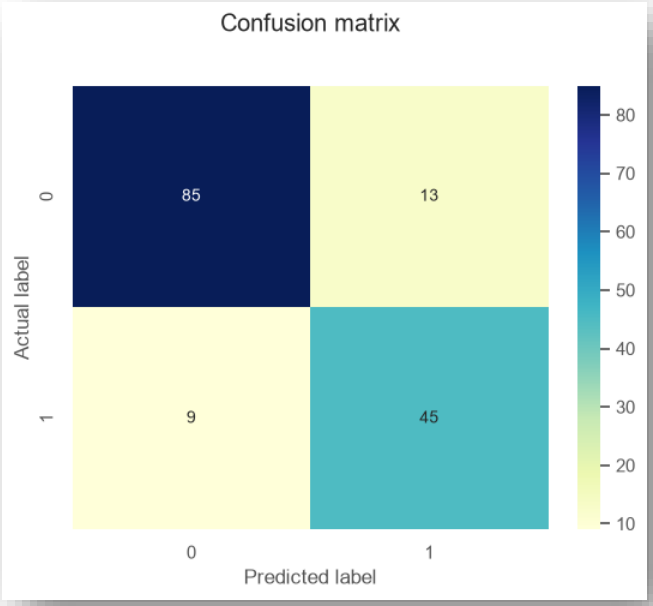
دقت این مدل بر روی داده های آموزش :

```
1.0
```

دقت این مدل بر روی داده های تست :

```
0.8552631578947368
```

ماتریس درهم ریختگی :



گزارش طبقه بندی :

	precision	recall	f1-score	support
0	0.90	0.87	0.89	98
1	0.78	0.83	0.80	54
accuracy			0.86	152
macro avg	0.84	0.85	0.84	152
weighted avg	0.86	0.86	0.86	152

در این مرحله، مدل **XGBoost (Extreme Gradient Boosting)** برای مسئله طبقه‌بندی دودویی ایجاد و بر روی داده‌های آموزشی آموزش داده شد. برای این مدل، ابرپارامترهایی مانند نرخ یادگیری (Learning Rate)، حداکثر عمق درخت‌ها (Max Depth) و تعداد درخت‌ها (Number of Estimators) تنظیم شدند تا عملکرد مدل بهبود یابد. پس از آموزش، عملکرد مدل با استفاده از معیارهای ارزیابی مورد بررسی قرار گرفت و نتایج آن با سایر الگوریتم‌های یادگیری ماشین مقایسه شد.

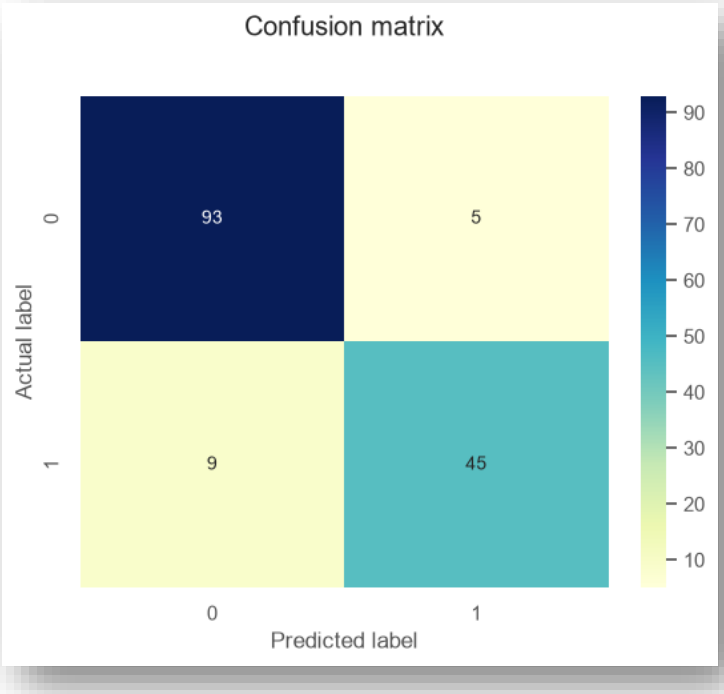
دقت این مدل بر روی داده های آموزش :

```
0.975328947368421
```

دقت این مدل بر روی داده های تست :

```
0.9078947368421053
```

ماتریس درهم ریختگی :



گزارش طبقه بندی :

	precision	recall	f1-score	support
0	0.91	0.95	0.93	98
1	0.90	0.83	0.87	54
accuracy			0.91	152
macro avg	0.91	0.89	0.90	152
weighted avg	0.91	0.91	0.91	152

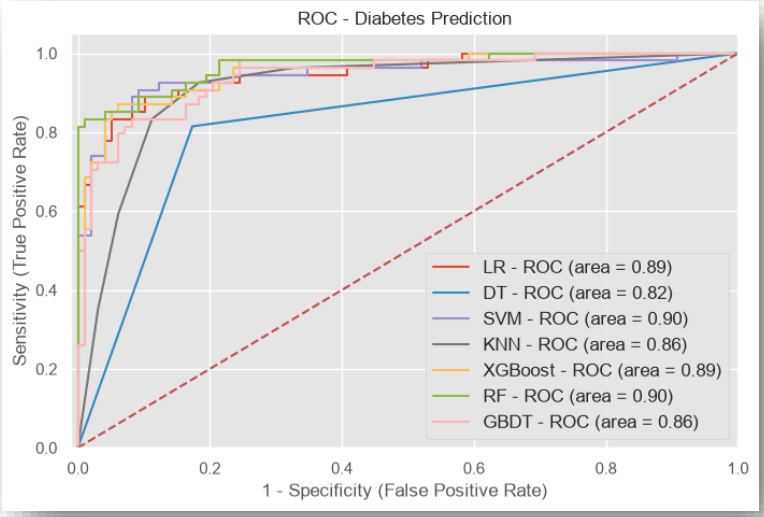
مقایسه و ارزیابی نهایی مدل‌ها

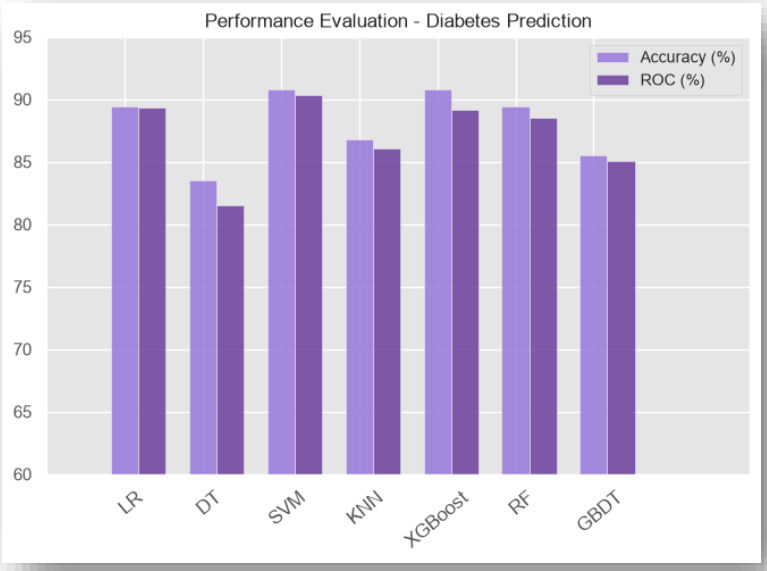
پس از آموزش تمامی الگوریتم‌ها، عملکرد آن‌ها با استفاده از معیارهای مختلف ارزیابی و با یکدیگر مقایسه شد.

ابتدا دقت (Accuracy) هر مدل محاسبه و در یک جدول تجمیع شد. سپس مدل‌ها بر اساس میزان دقت به صورت نزولی مرتب شدند تا بهترین مدل از نظر عملکرد مشخص شود :

	Model	Score
2	SVM	90.79
6	XgBoost	90.79
0	Logistic Regression	89.47
4	Random Forest Classifier	89.47
1	KNN	86.84
5	Gradient Boosting Classifier	85.53
3	Decision Tree Classifier	83.55

در ادامه، برای ارزیابی دقیق‌تر توانایی مدل‌ها در تفکیک افراد مبتلا و غیرمبتلا به دیابت، منحنی ROC (Receiver Operating Characteristic) برای تمامی مدل‌ها رسم شد. این منحنی رابطه بین نرخ تشخیص صحیح (True Positive Rate) و نرخ خطای مثبت (False Positive Rate) را نمایش می‌دهد :



همچنین مقدار AUC (Area Under Curve) برای هر مدل محاسبه شد. این معیار نشان‌دهنده قدرت مدل در تفکیک کلاس‌ها بوده و هرچه مقدار آن به ۱ نزدیک‌تر باشد، عملکرد مدل بهتر است.

در نهایت، نتایج دو معیار Accuracy و ROC-AUC به صورت نمودار میله‌ای نمایش داده شدند تا مقایسه بصری بین مدل‌ها انجام شود. این مقایسه امکان شناسایی مناسب‌ترین الگوریتم برای پیش‌بینی ابتلا به دیابت را فراهم کرده و مبنایی برای انتخاب مدل نهایی ارائه می‌دهد :