سام سرلک گودرزی

درس هوش مصنوعی ، دانشکده مهندسی مکانیک، دانشگاه تهران

استاد: دکتر مسعود شریعت پناهی

موضوع: تمرین دوم – بخش اول

مقدمه

هدف این تمرین بررسی اثرات ویژگی‌های ماشین بر هزینه آنها است که با استفاده از الگوریتم‌های یادگیری ماشین انجام شده‌است

الف-۱) با استفاده از دستور info متوجه می‌شویم که هیچ کدام از داده‌های داخل جدول مقدار گم‌شده‌ای[1] ندارند. در ادامه این گزارش قابل مشاهده است که داده‌های حاضر از جنس عددی و دسته‌بندی تشکیل شده‌اند.

جدول ۱. خروجی دستور info، داده‌های int64، float64 داده عددی و object از جنس دسته‌بندی است

Column	Non-Null Count	Dtype
Car_ID	205 non-null	int64
Symboling	205 non-null	int64
CarName	205 non-null	object
fueltype	205 non-null	object
aspiration	205 non-null	object
doornumber	205 non-null	object
doornumber	205 non-null	object
carbody	205 non-null	object
drivewheel	205 non-null	object
enginelocation	205 non-null	object
wheelbase	205 non-null	float64
carlength	205 non-null	float64
carwidth	205 non-null	float64
carheight	205 non-null	float64
curbweight	205 non-null	int64
enginetype	205 non-null	object
cylindernumber	205 non-null	object
enginesize	205 non-null	int64
fuelsystem	205 non-null	object
boreratio	205 non-null	float64
stroke	205 non-null	float64
compressionratio	205 non-null	float64
horsepower	205 non-null	int64
peakrpm	205 non-null	int64
citympg	205 non-null	int64
highwaympg	205 non-null	int64
price	205 non-null	float64

[1] Missing value

الف-۲) با اجرا دستور describe() بر ستون price، مقادیر کمینه، بیشینه و انحراف معیار، در کنار مقدار میانگین و چارک‌های میان مقدار کمینه و بیشینه گزارش داده می‌شود.

جدول ۲. خروجی دستور describe()

count	205
mean	13276
std	7988
min	5118
25%	7788
50%	10295
75%	16503
max	45400

انحراف معیار ۷۹۶۰.۳ دلاری قیمت‌ها نشان می‌دهد که هزینه ماشین‌های حاضر در جدول، به صورت میانگین، به این مقدار از میانگین قیمت تمام خودروها، به مقدار ۱۳۲۷۶.۷۱ دلار، انحراف دارند. این عدد نشان می‌دهد که قیمت ها از میانگین فاصله زیادی دارند و تنوع قیمتی بالایی در این لیست وجود دارد.

الف-۳) برای محاسبه همبستگی بین هر یک از ویژگی‌ها با هزینه ابتدا از روش Label encoding داده‌های دسته‌بندی به داده‌های عددی تبدیل شدند. برای اینکار با استفاده از دستور select_dtypes داده‌هایی که ماهیت object داشتند از مجموعه داده انتخاب شدند و سپس با دستور LabelEncoder() به فرمت عددی تبدیل شدند. این مجموعه در ادامه به دادگان عددی متصل شده و با حذف ستون هزینه(price) همبستگی آنها با هزینه خودروها محاسبه شده.

جدول ۳.همبستگی هر ویژگی با هزینه نهایی خودرو

car_ID	-0.10909
symboling	-0.07998
wheelbase	0.577816
carlength	0.68292
carwidth	0.759325
carheight	0.119336
curbweight	0.835305
enginesize	0.874145
boreratio	0.553173
stroke	0.079443
compressionratio	0.067984
horsepower	0.808139
peakrpm	-0.08527

citympg	-0.68575
highwaympg	-0.6976
CarName	-0.23144
fueltype	-0.10568
aspiration	0.177926
doornumber	-0.03184
carbody	-0.08398
drivewheel	0.577992
enginelocation	0.324973
enginetype	0.049171
cylindernumber	-0.02763
fuelsystem	0.526823

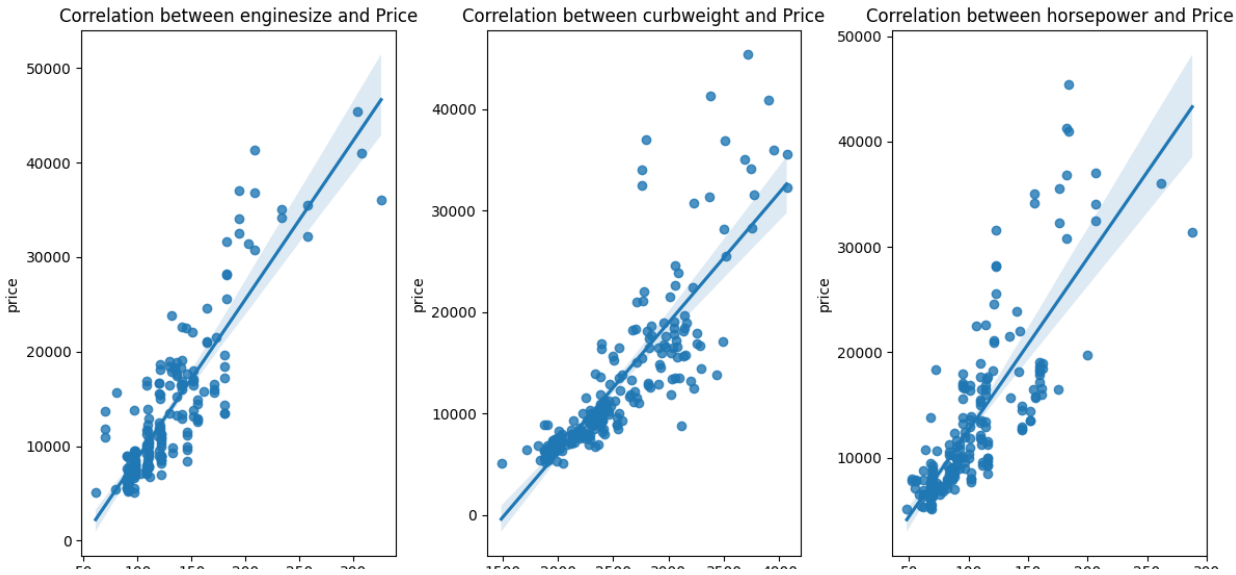
همبستگی ویژگی‌ها بر اساس اثری که بر هزینه دارند در بازه ۱- تا ۱ قرار می‌گیرند. هر چه قدرمطلق این مقدار بیشتر باشد، همبستگی آن ویژگی با هزینه بیشتر است و منفی یا مثبت بودن این مقدار به ترتیب اثر معکوس و موافق آن ویژگی را بر هزینه نهایی نشان می‌دهد.

ب) پیش پردازش دادگان

از آن جایی که در میان داده‌ها مقدار گم‌شده‌ای وجود ندارد و همچنین مقادیر دادگان در بازه منطقی قرار دارند(بر اساس گزارش describe) و از این نظر نیازی به پیش‌پردازش داده‌ها وجود ندارد. هر چند در ادامه با رسم کردن joinplot مشخص می‌شود که داده‌های حاضر پراکندگی و پرتی دارند که کنترل خواهند شد.

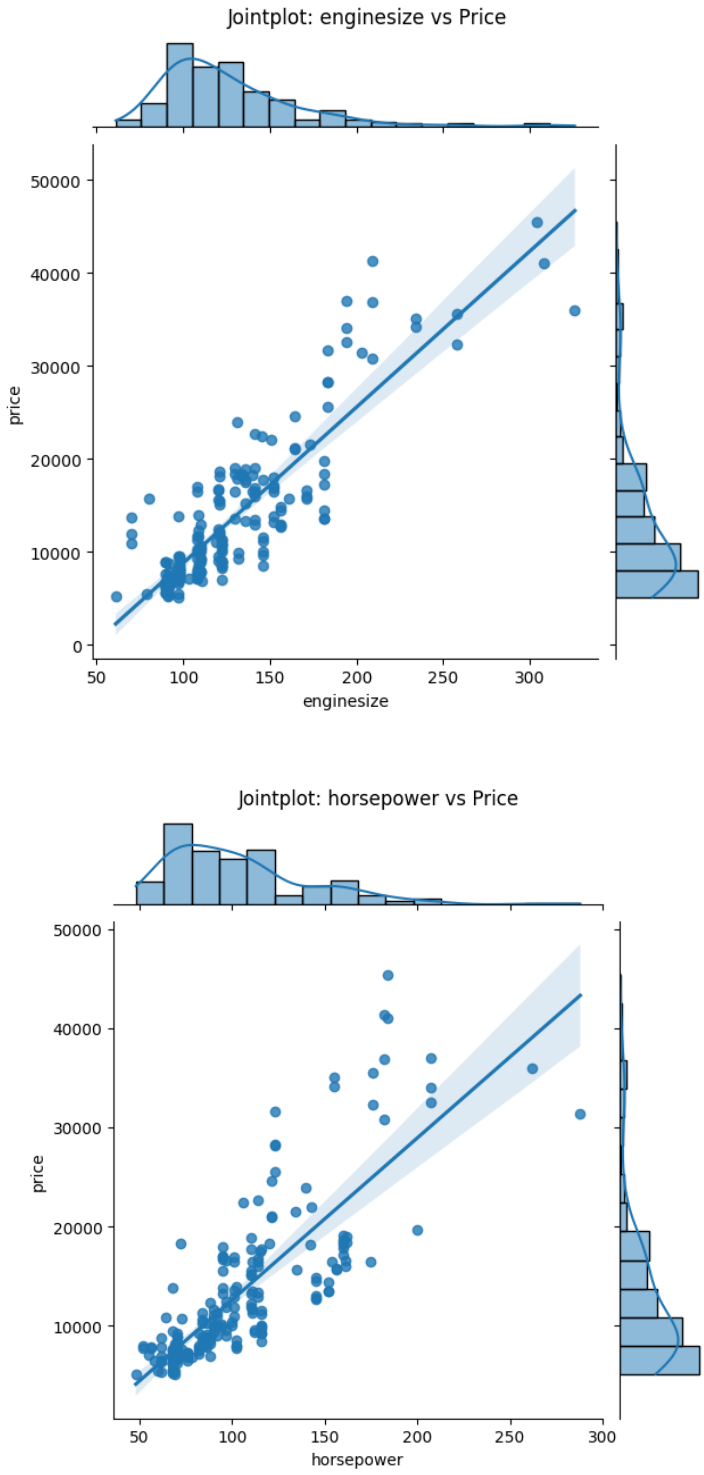
نتایجی که قدرمطلق آنها کمتر از ۲ بوده در این مرحله از مجموعه‌دادگان حذف شده و به عنوان داده ذخیره شده‌است.

از بین ویژگی‌ها سایز موتور، وزن خودرو، و قدرت خودرو به ترتیب بیشترین همبستگی را با هزینه دارند.



شکل ۱.همبستگی ویژگی‌های سایز موتور، وزن خودرو، و قدرت خودرو (به ترتیب از چپ به راست)

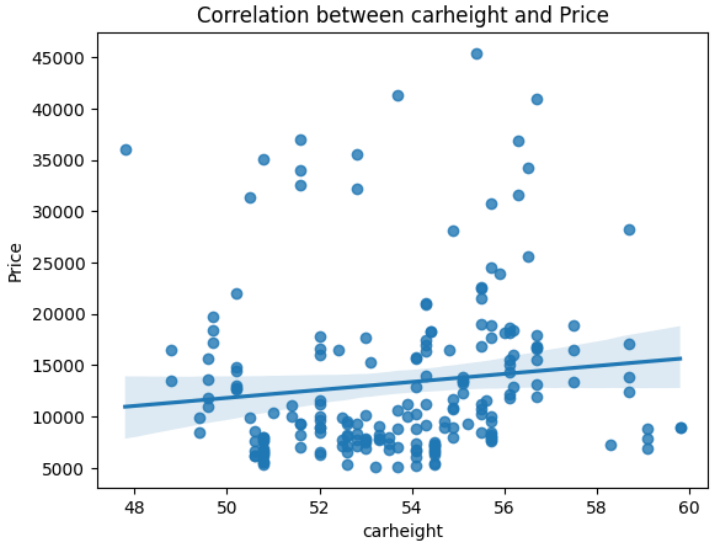
ج-۲) اجرای دستور joinplot نمودارهای شکل۲ را تولید می‌شود.



شکل ۲. نمودار joinplot دو ویژگی قدرت خودرو و سایز موتور

بررسی این دو نمودار این نتیجه را حاصل می‌کند که دو ویژگی قدرت موتور و سایز موتور هر دو برای آموزش مدل رگرسیون مناسب‌اند زیرا داده‌های رسم شده به مقدار قابل قبولی روند خطی را با شیب نزدیک ۰.۸ دنبال می‌کنند. که در بخش قبلی نیز ثابت شده‌بود. برداشت دیگری که از این دو نمودار می‌توان داشت، وجود داده‌های پرت در این دو ویژگی و همچنین هزینه خودروها وجود دارد.

بر خلاف دو ویژگی ذکر شده، با بررسی نمودار شکل ۳ می‌توان مشاهده کرد که ویژگی "ارتفاع ماشین" ارتباط ضعیفی را با هزینه خودرو نشان می‌دهد و داده‌ها به شکل نامتقارنی اطراف رگرسیون خطی میان این ویژگی و هزینه پراکنده شده‌اند.



شکل ۳. نمودار نشان دهنده همبستگی میان ارتفاع خودرو و هزینه

ج-۳) با استفاده از دستور SelectKBest و انتخاب عدد ۱۰ برای متغیر K، ۱۰ تا ویژگی برتر (مؤثر بر هزینه) انتخاب شدند و از این ۱۰ ویژگی در ادامه برای تربیت مدل استفاده شده.

جدول ۴. ۱۰ ویژگی برتر حاضر در مجموعه دادگان که توسط دستور SelectKBest استخراج شده‌اند

Feature	Value
wheelbase	0.577816
carlength	0.68292
carwidth	0.759325
curbweight	0.835305
enginesize	0.874145
boreratio	0.553173
horsepower	0.808139
citympg	-0.68575
highwaympg	-0.6976
drivewheel	0.577992
fuelsystem	0.526823

ج-۱) اجرای دستور train_test_split در این مرحله انجام شده و دادگان به دو بخش train (در برگیرنده ۷۰٪ مجموعه داده) و بخش test (در برگیرنده ۳۰٪ مجموعه داده) تقسیم شده‌اند.

ج- ۴، ۵ و ۶) در مرحله اجرا مدل‌های یادگیری ماشین از داده‌ها تولید شده در بخش ج-۱ استفاده شده و برای هر کدام نیز داده مربوط به میانگین مجموع مربعات و ضریب تعیین R^2 ، به کمک دستورهای mean_squared_error و r2_score حساب شده‌است.

	RMSE	R2
linear regression	3918	0.778
Lasso	3918	0.778
Ridge	3919	0.778
SVR	3937	0.776

سام سرلک گودرزی

درس هوش مصنوعی ، دانشکده مهندسی مکانیک، دانشگاه تهران

استاد: دکتر مسعود شریعت پناهی

موضوع: تمرین دوم

مقدمه

هدف این تمرین طراحی و آموزش مدلی بر پایه ۸ ویژگی افراد مؤنث که ابتلا یا عدم ابتلا آنها را به بیماری دیابت بررسی می‌کند. تمرین نمونه‌ای از دسته وضایف دسته‌بندی دوگانه است.

ویژگی‌های ذکر شده در جدول تعداد دفعات بارداری، سطح گلوکز در خون، فشار خون، ضخامت پوست، سطح انسولین در خون، شاخص توده‌ی بدنی، ریسک دیابت نوع ۲، و سن شخص است.

در ادامه پاسخ به سؤال‌های مطرح شده به ترتیب پاسخ داده‌شده و روند توضیحات داخل کد نیز همین مسیر را پیروی می‌کند.

الف) دادگان خام

الف-۱) ابتدا داده خام بارگذاری شده و با استفاده از دستورات Describe() و Info() مشخصات اولیه و کلی مجموعه داده بدست آمده و در جدول ۱ و ۲ قابل مشاهده است.

جدول ۱. مشخصات استخراج شده از دستور info()

Feature	Non-Null Count	Dtype
Pregnancies	635	float64
Glucose	654	float64
BloodPressure	680	float64
SkinThickness	624	float64
Insulin	680	float64
BMI	684	float64
DPF[2]	590	float64
Age	655	float64
Outcome	768	int64

[2] DiabetesPedigreeFunction

جدول ۲. مشخصات استخراج شده از دستور describe()

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DPF	Age	Outcome
count	635	654	680	624	680	684	590	655	768
mean	3.700787	113.422	68.786765	20.38622	80.1235	32.08	0.466676	33.16	0.349
std	3.518126	202.817	19.724841	15.98705	115.681	7.801	0.322408	13.83	0.477
min	-22	-5000	-2	0	0	0	0.078	-150	0
25%	1	99	62	0	0	27.4	0.24325	24	0
50%	3	117	72	23	34	32.3	0.368	29	0
75%	6	140.75	80	32	129.25	36.6	0.6115	41	1
max	17	199	122	99	846	67.1	2.329	81	1

الف-۲) از خروجی دستور info() تعداد داده‌های غیر گم‌شده مجموعه مشخص شده و همچنین با کمک دستور isnull() به صورت جدا این داده‌ها که به فرمت NaN هستند شناسایی شده و در ادامه نسبت داده‌های گم‌شده در هر ستون(ویژگی) نسبت به کل داده‌های آن ستون چاپ شده که در جدول ۳ قرار دارد.

جدول ۳. تعداد داده‌های گم‌شده در هر ستون و نسبت آن به کل داده‌ها

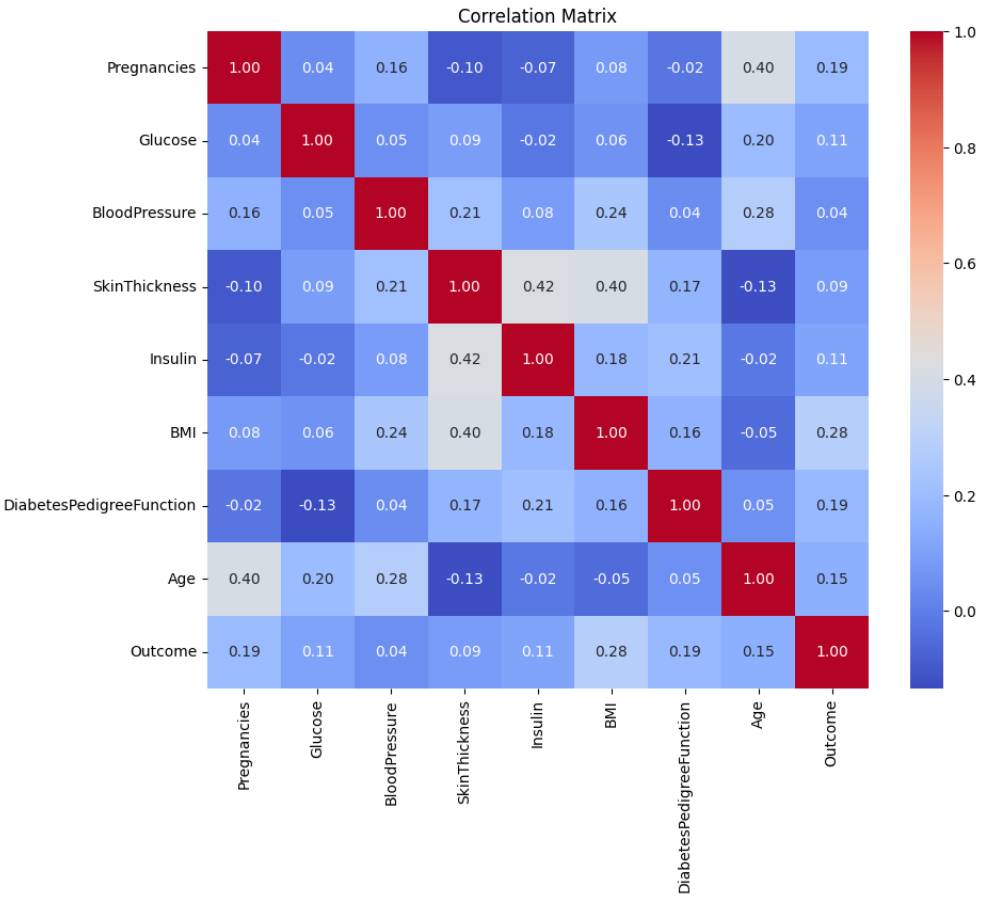
Column	NaN Count	NaN Percentage
Pregnancies	133	17.317708
Glucose	114	14.84375
BloodPressure	88	11.458333
SkinThickness	144	18.75
Insulin	88	11.458333
BMI	84	10.9375
DPF	178	23.177083
Age	113	14.713542

الف-۳) در این قسمت هم‌بستگی میان ویژگی‌ها و نتیجه آزمایشات حساب شده و نقشه حرارتی این هم‌بستگی به کمک کتابخانه seaborn رسم شده(شکل ۱).

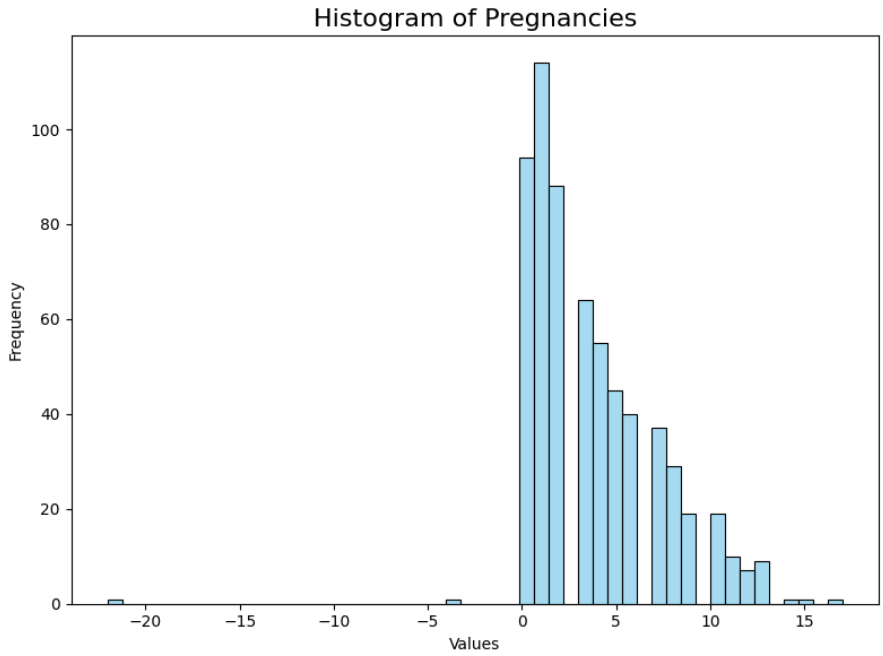
طبق نقشه حرارتی و مقادیر چاپ شده هم‌بستگی، به ترتیب شاخص توده‌ی بدنی، ریسک دیابت نوع ۲، تعداد دفعات بارداری و سن شخص بیشترین مقدار هم‌بستگی را با نتیجه آزمایش دارند. با اعمال بازه قابل قبول هم‌بستگی بیش از ۰.۱۵، این چهار ویژگی از دیگر ویژگی‌های موجود انتخاب شده‌اند.

الف-۴) در شکل ۲ تا ۵ تعداد مشاهدات مربوط به چهار ویژگی ذکر شده در بخش قبلی نمایش داده شده که بررسی آنها، حاضر بودن داده‌های غلط و غیر قابل قبولی همچون تعداد بارداری منفی و شاخص توده‌ی بدنی صفر را نشان می‌دهد.

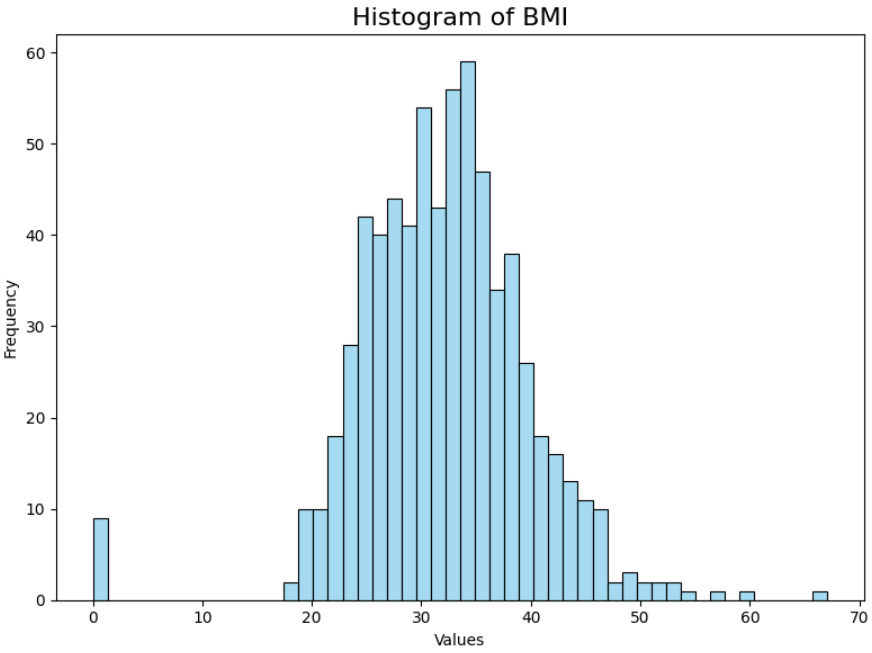
همچنین "ریسک دیابت نوع ۲" بازه بین ۰.۰۸ و ۲.۴۲ قابل قبول است که داده‌ها تقریبا این شرط را برآورده می‌کنند.



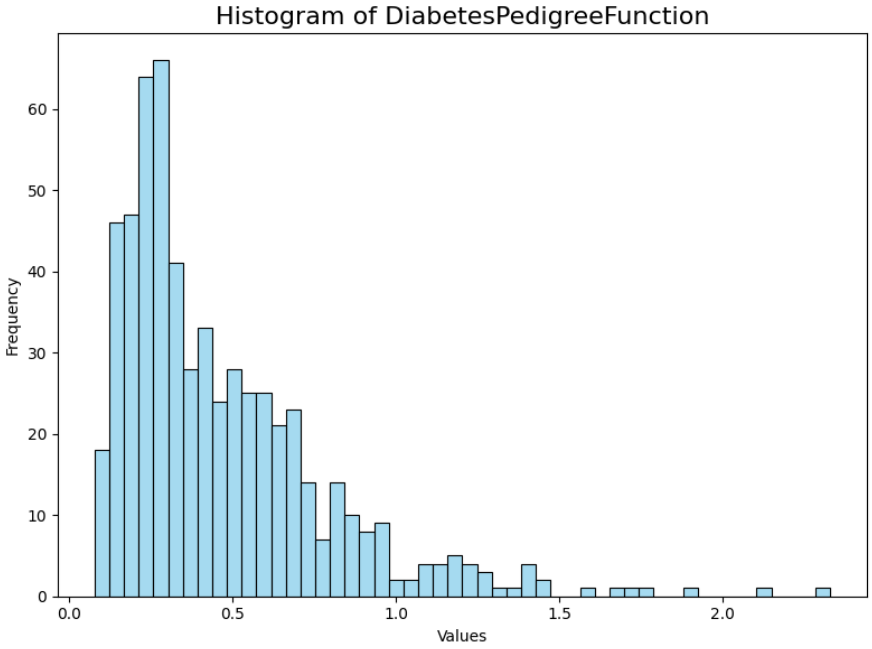
شکل ۱. نقشه حرارتی هم‌بستگی میان ویژگی‌ها و نتیجه آزمایشات



شکل ۲. تعداد مشاهدات هر مقدار منحصر به فرد در تعداد دفعات بارداری

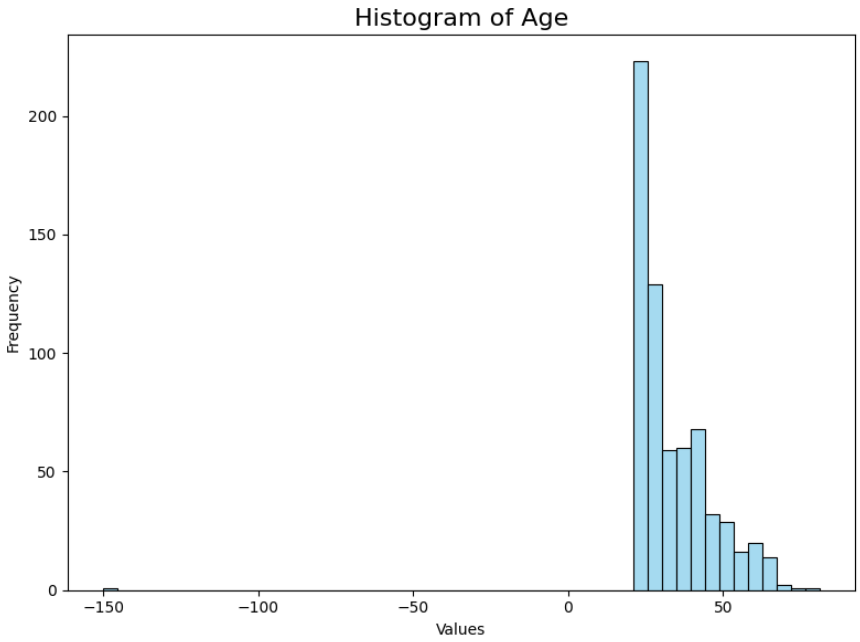


شکل ۳. تعداد مشاهدات هر مقدار منحصر به فرد در شاخص توده بدنی



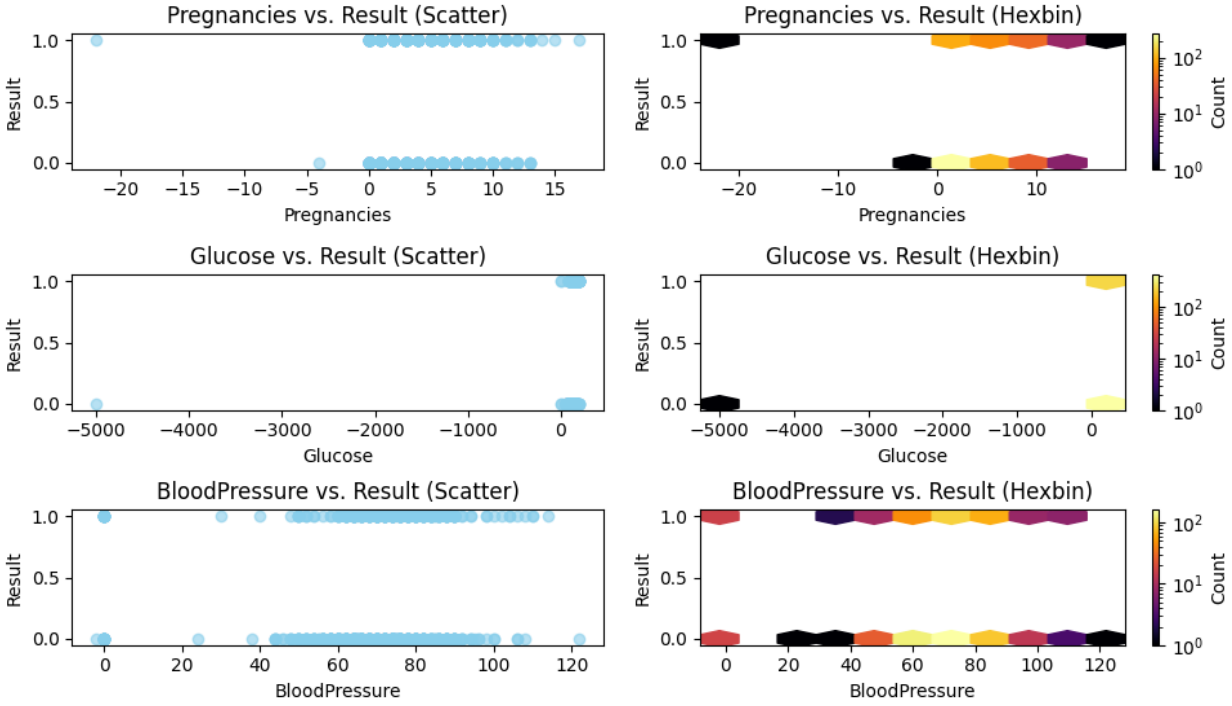
شکل ۴. تعداد مشاهدات هر مقدار منحصر به فرد در ریسک دیابت نوع ۲

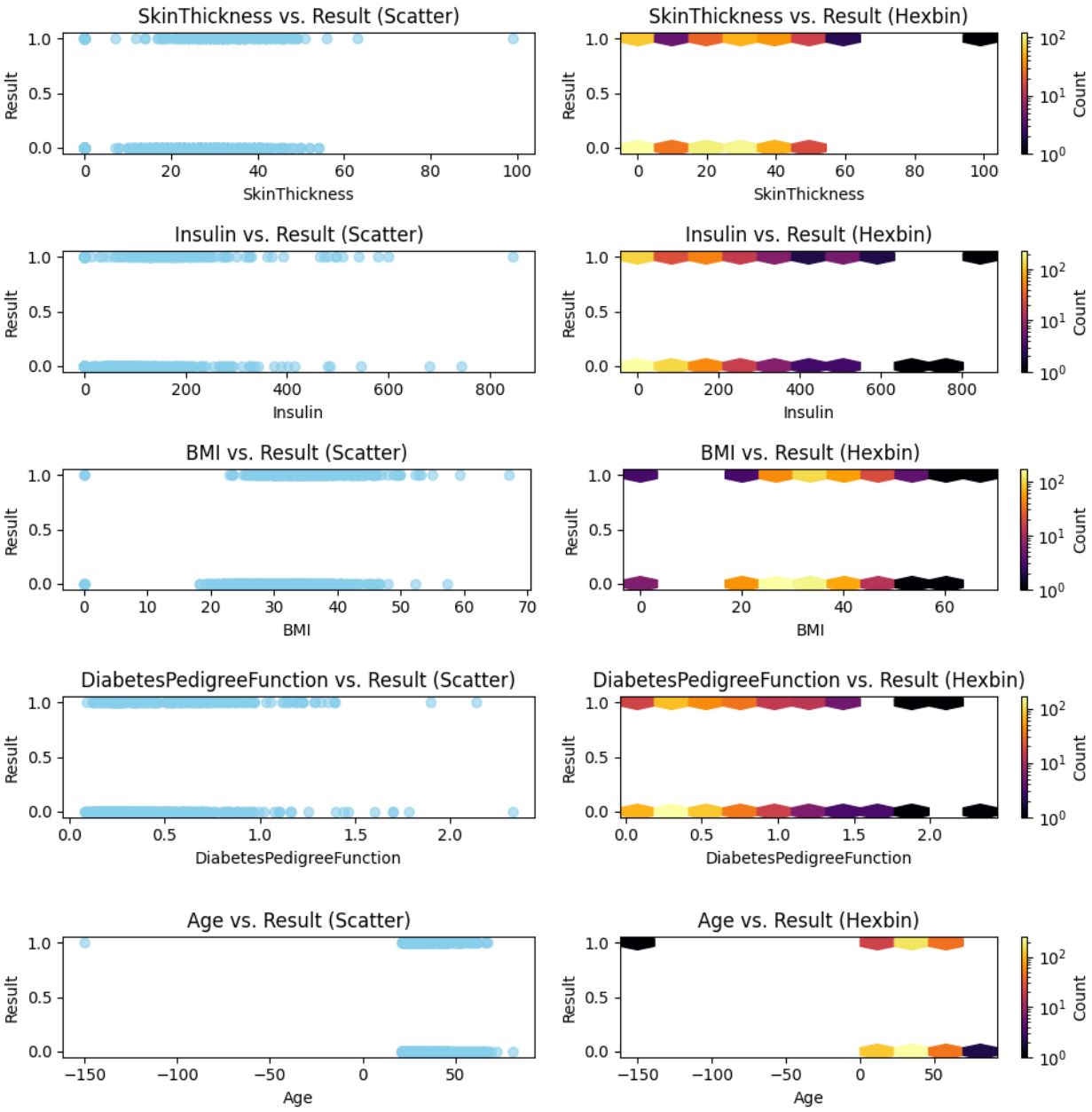
داده‌های مربوط به سن نیز مانند ویژگی "ریسک دیابت نوع ۲" نیازی به دستکاری ندارند زیرا بازه سنی منطقی را پوشش می‌دهند.



شکل ۵. تعداد مشاهدات هر مقدار منحصر به فرد در سن افراد

الف-۵) برسی نمودارهای Hexbin





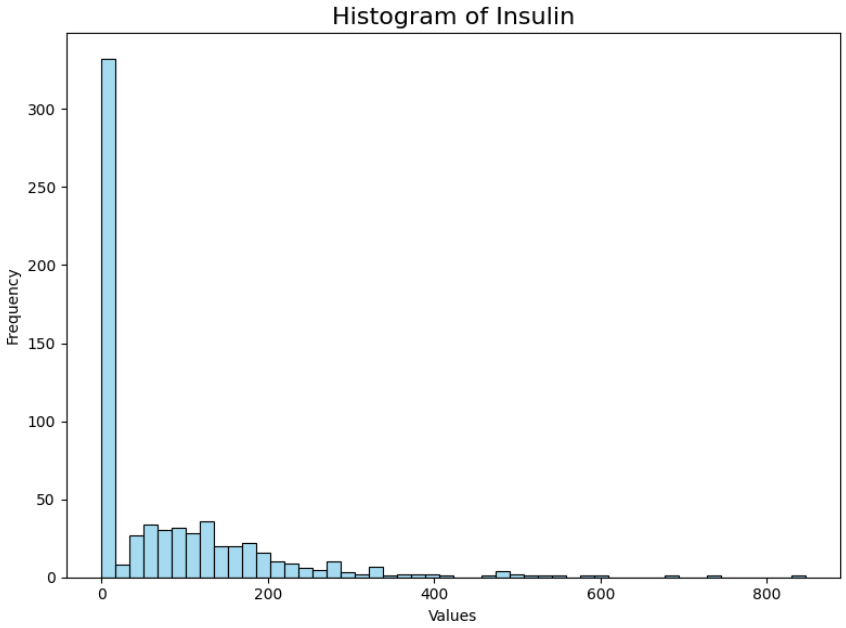
شکل ۶. نمودارهای hexbin و scatter که ۸ ویژگی داده‌ها را به صورت جدا در کنار نتایج نشان می‌دهند.

با بررسی شکل ۶ می‌توان متوجه وجود پراکندگی میان داده‌ها و حضور داده‌های پرت را در بیشتر ویژگی‌ها مشاهده کرد که در ادامه در مرحله پیش‌پردازش به این مشکلات رسیدگی می‌شود.

ب) پیش‌پردازش دادگان

ب-۱) اگر بر اساس تعداد داده‌های گم‌شده و میزان هم‌بستگی داده‌ها قضاوت شود، دو ویژگی "ضخامت پوست" به دلیل فراوانی داده‌های گم‌شده و همینطور هم‌بستگی پایین(۰.۰۹) و "ریسک دیابت نوع ۲" به دلیل داشتن ۱۷۸ داده گم‌شده(بیشترین میان دیگر ویژگی‌ها) باید حذف شوند و می‌توانند بر پیش‌بینی مدل اثر منفی بگذارند.

در ادامه با بررسی نمودار شکل ۷، ویژگی "سطح انسولین در خون" برای بیش از ۳۰۰ داده مقدار صفر دارد که برای انسان زنده ممکن نیست و خارج بازه قابل قبول می‌باشد. همچنین این ویژگی دارای هم‌بستگی نسبتاً پایینی با نتایج است(۰.۱۱)(شکل۱).



شکل ۷. تعداد مشاهدات هر مقدار منحصر به فرد برای ویژگی سطح انسولین خون

با در نظر گرفتن داده‌های غیر قابل قبول، هم‌بستگی پایین و همچنین تعداد نسبتاً زیادی از داده‌های گم‌شده، این ویژگی در کنار ویژگی "ریسک دیابت نوع ۲" از گزینه‌های حذف کردن است. در ادامه با تربیت مدل‌ها می‌توان طبق جدول ۴ این نتیجه را گرفت که با نگه داشتن ویژگی "ریسک دیابت نوع ۲" و حذف ویژگی "سطح انسولین در خون" می‌توان مدلی با دقت بالاتر آموزش داد.

جدول ۴. مقایسه دقت مدل‌های متفاوت برای شرایطی که ویژگی انسولین خون، ریسک دیابت نوع ۲ و یا هر دو از داده‌های آموزش مدل حذف شوند

Model	Accuracy		
	Insulin and DiabetesPedigreeFunction	Insulin	DiabetesPedigreeFunction
Logistic Regression	0.766	0.773	0.766
KNN	0.734	0.727	0.688
Decision Tree	0.669	0.701	0.669
Random Forest	0.747	0.759	0.720
SVM	0.753	0.753	0.740

ویژگی‌های "تعداد دفعات بارداری"، "سطح گلوکز در خون"، و "سن" به ترتیب ۱۳۳، ۱۱۴، و ۱۱۳ داده گم‌شده داشته ولی به دلیل عدم حضور داده‌های پرت زیاد، با استناد به شکل ۲،۳و ۵، و همبستگی متوسط تا قوی این ویژگی‌ها به نتیجه (شکل۱)، این سه ویژگی برای کمک به پیچیدگی مجموعه داده حفظ شده‌اند.

برای تنظیم داده‌گان گم‌شده در هر ستون(ویژگی)، از مقدار میانه [3]مربوط به همان ویژگی استفاده شده. بین مقدار میانگین و میانه، طبق جدول۵ مقدار میانه به دلیل تولید اعداد صحیح برای دادگان استفاده شده. زیرا اعداد غیر صحیح برای تعداد بارداری معنی ندارند.

جدول ۵. مقایسه دو مقدار میانه و میانگین برای ویژگی‌های مختلف مجموعه دادگان

Index	median	Mean
Pregnancies	3	3.7
Glucose	117	113.42
Blood pressure	72	68.78
Skin thickness	23	20.386
Insulin	34	80.12
BMI	32.3	32.08
Diabetes pedigree function	0.368	0.466
Age	29	33.157

مقادیر منفی برای "تعداد دفعات بارداری" و مقادیر کوچکتر و مساوی صفر نیز از دیگر ویژگی‌ها نیز در این مرحله تبدیل به داده‌های گم‌شده شدند تا در مرحله بعد کنار دیگر دادگان گم‌شده برابر با مقدار میانه فرض شوند تا ویژگی‌ها در بازه‌های قابل قبول و منطقی قرار داشته‌باشند.

ب-۲) استانداردسازی[4] و نرمال‌سازی[5] تکنیک‌های پیش پردازشی هستند که برای تبدیل ویژگی‌های مجموعه داده‌ها به مقیاس مشابه قبل از وارد کردن آنها به الگوریتم‌های یادگیری ماشین استفاده می‌شود.

زمانی که توزیع داده ها گاوسی نیست (مانند ویژگی سن در شکل۵) و زمانی که داده های مورد نظر مرکزیت صفر دارند، استانداردسازی ترجیح داده می شود. برعکس، زمانی که داده ها دارای توزیع گاوسی است (مانند ویژگی شاخص توده بدن)، استانداردسازی در داده‌ها برای پیش‌پردازش مدل یادگیری ماشین مفید است. هرچند لزوماً اینطور نیست. بر خلاف عادی سازی، استانداردسازی همیشه محدوده مرزی ندارد. بنابراین، هر گونه پرتی درون داده ها تحت تاثیر آن قرار نخواهد گرفت.

نرمال سازی زمانی مناسب است که ویژگی ها مقیاس ها یا واحدهای متفاوتی داشته باشند و زمانی که الگوریتم به بزرگی مقادیر متکی است. انتخاب بین استانداردسازی و عادی سازی به ویژگی های خاص داده ها و الزامات الگوریتم یادگیری ماشین بستگی دارد.

استانداردسازی عموماً برای الگوریتم‌هایی که داده‌های مرکز صفر را فرض می‌کنند، مانند رگرسیون خطی، رگرسیون لجستیک و SVM مناسب است در حالی که عادی‌سازی برای الگوریتم‌های مبتنی بر فاصله میان داده‌ها مناسب‌تر است همچون K-Nearest Neighbors (KNN).

[3] Median
[4] Standardizing
[5] Normalization

این استدلال در جدول۶ قابل مشاهده است که در بین مدل‌های مختلف، نرمال‌سازی مجموعه داده، عملکرد مدل KNN را از حالتی که استانداردسازی انجام شده بالاتر برده‌است. در حالی که دیگر الگوریتم‌ها از فرآیند استانداردسازی سود بیشتری برده‌اند.

جدول ۶. مقایسه عملکرد مدل‌های یادگیری ماشین زمانی که نرمال‌سازی و یا استانداردسازی انجام شده‌باشد

Model	Accuracy		
	Simple	Normalized	Standardized
Logistic Regression	0.773	0.773	0.779
KNN	0.727	0.76	0.753
Decision Tree	0.747	0.695	0.714
Random Forest	0.753	0.773	0.766
SVM	0.753	0.766	0.773

روابط مربوط به استانداردسازی و نرمال‌سازی داده‌های یک مجموعه به صورت زیر می‌باشد.

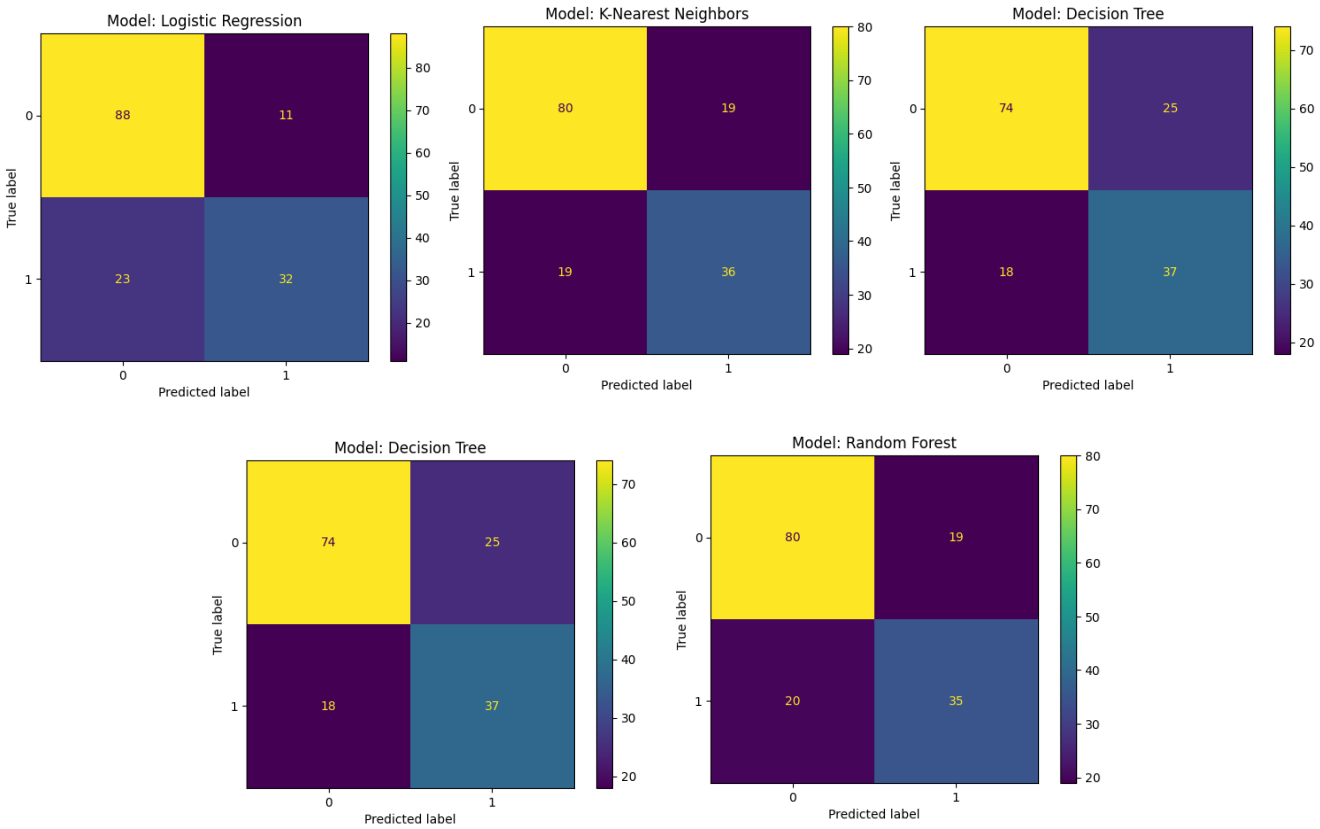
$$X_{standardized} = \frac{X - \mu}{\sigma}, \qquad X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

در روابط بالا X، X_{min} و X_{max} مقادیر اصلی دادگان در هر ویژگی و سپس کمترین و بیشترین مقدار آن‌ها، μ میانگین هر ویژگی، و σ انحراف معیار است.

ج) انتخاب، آموزش و ارزیابی مدل

ج-۱ الی ۳) در این بخش از تمرین که در کد به صورت کامل‌تری دسته‌بندی انجام شده، تقسیم دادگان به مجموعه آموزش و تست، فراخوانی مدل‌ها و محاسبه دقت مدل بر اساس ماتریس سردرگمی به ترتیب از مراحل آموزش این مرحله بوده.

ابتدا نتایج مربوط به آموزش مدل بدون دستکاری هایپرپارامترها و پس از اجرای استانداردسازی حساب شده که در جدول۶ کنار ماتریس‌های سردرگمی شکل۸ قابل مشاهده است.



شکل ۸. مجموعه نمودارهای ماتریس سردرگمی

در مرحله بعد با تعریف هایپرپارامترها و آموزش مجدد مدل‌ها، نتایج جدول ۷ ایجاد شده‌است.

جدول ۷. مقایسه عملکرد مدل‌های یادگیری ماشین پس از بهینه‌سازی هایپرپارامترهای مدل‌ها

Model	Accuracy	
	Training Set	Test Set
Logistic Regression	0.764	0.779
KNN	0.817	0.753
Decision Tree	0.756	0.727

هایپر پارامترهای انتخاب شده و بازه تغییراتشان به شرح زیر می‌باشد.

رگرسیون لجستیک:

max_iter: این هایپرپارامتر حداکثر تعداد تکرارها را برای همگرا شدن الگوریتم بهینه سازی تعیین می کند. اگر الگوریتم در تعداد تکرارهای مشخص شده همگرا نشود، متوقف می شود و راه حل فعلی را برمی گرداند. بازه تعیین شده برای این هایپرپارامتر ۱۰، ۱۰۰ و ۲۵۰ بوده که مقدار ۱۰ مقدار بهینه بوده. (مقدار بهینه: ۱۰)

C: پارامتر C معکوس قدرت منظم‌سازی[6] را نشان می دهد. مقادیر کوچکتر C، منظم‌سازی قوی‌تر را مشخص می‌کند، به این معنی که مدل کمتر برازش داده‌های آموزشی خواهد داشت. مقادیر بالاتر C به مدل اجازه می‌دهد تا داده‌های آموزشی را با دقت بیشتری منطبق کند، که به طور بالقوه منجر به بیش از حد برازش می‌شود. بازه تعیین شده برای این هایپرپارامتر بین ۱ و ۰.۱ است. (مقدار بهینه: ۱)

K-نزدیکترین همسایگان (KNN):

n_neighbors: این هایپرپارامتر تعداد همسایه هایی را که باید در هنگام پیش بینی برای یک نقطه داده جدید در نظر گرفته شود را مشخص می‌کند. مقدار کوچکتر n_neighbors، مدل را نسبت به تغییرات محلی در داده‌ها حساس‌تر می‌کند و به طور بالقوه منجر به بیش‌برازش می‌شود. برعکس، یک مقدار بزرگتر از n_neighbors منجر به یک مرز تصمیم هموارتر می شود، که در صورت پیچیده بودن مجموعه داده ممکن است منجر به عدم تناسب شود. بازه تعیین شده برای این هایپرپارامتر ۳، ۵، و ۷ است. (مقدار بهینه: ۵)

درخت تصمیم:

max_depth: فراپارامتر max_depth حداکثر عمق درخت تصمیم را تعیین می کند. درخت عمیق‌تر می‌تواند الگوهای پیچیده‌تری را در داده‌ها ثبت کند، اما مستعد بیش از حد برازش است. برعکس، یک درخت کم عمق کمتر احتمال دارد که بیش از حد مناسب بیش‌برازش شود، اما ممکن است تمام تفاوت های ظریف در داده ها را در بر نگیرد. بازه تعیین شده برای این هایپرپارامتر ۳، ۵، و ۷ است. (مقدار بهینه: ۳)

min_samples_split: این هایپرپارامتر حداقل تعداد نمونه های مورد نیاز برای تقسیم یک گره داخلی را مشخص می کند. افزایش min_samples_split منجر به شکاف های کمتری در درخت می شود و در نتیجه مدلی ساده‌تر با شانس کمتری برای بیش از حد برازش ایجاد می شود. با این حال، تنظیم بیش از حد آن ممکن است باعث شود که مدل با داده های آموزشی مناسب نباشد. بازه تعیین شده برای این هایپرپارامتر ۲ الی ۶ است. (مقدار بهینه: ۲)

جنگل تصادفی:

n_estimators: تعداد درختان در مجموعه جنگل تصادفی. افزایش تعداد درختان به طور کلی عملکرد مدل را بهبود می بخشد، اما پیچیدگی محاسباتی را نیز افزایش می دهد. درختان بیشتر به کاهش بیش از حد برازش و بهبود تعمیم مدل کمک می کند. بازه تعیین شده برای این هایپرپارامتر ۱۰۰ و ۲۰۰ است. (مقدار بهینه: ۱۰۰)

[6] Regularization

max_depth: مشابه درخت تصمیم، max_depth حداکثر عمق هر درخت را در الگوریتم جنگل تصادفی کنترل می کند. درختان عمیق تر می توانند الگوهای پیچیده تری را ثبت کنند، اما ممکن است منجر به بیش از حد برازش شوند. تنظیم یک مقدار مناسب برای max_depth به تعادل پیچیدگی و تعمیم مدل کمک می‌کند. بازه تعیین شده برای این هایپرپارامتر ۵ و ۱۰ است. (مقدار بهینه: ۵)

ماشین بردار پشتیبان (SVM):

C: پارامتر منظم‌سازی C مبادله بین حداکثر کردن حاشیه و به حداقل رساندن خطای طبقه بندی را کنترل می کند. مقادیر کوچکتر C منجر به حاشیه بزرگتر می شود اما ممکن است برخی از نکات آموزشی را به اشتباه طبقه بندی کند. مقادیر بزرگتر C، طبقه بندی صحیح هر نقطه تمرین را اولویت بندی می‌کند، اما ممکن است منجر به حاشیه کمتر و به طور بالقوه بیش از حد برازش شود. بازه تعیین شده برای این هایپرپارامتر ۰.۱ و ۱ است. (مقدار بهینه : ۰.۱)

kernel: تابع هسته، نوع مرز تصمیم ایجاد شده توسط SVM را تعیین می‌کند. انتخاب های رایج عبارتند از خطی، تابع پایه شعاعی[۷]، و چند جمله ای. هر هسته تأثیر خاص خود را بر روی مرز تصمیم دارد و انتخاب به ویژگی های مجموعه داده و مشکل موجود بستگی دارد. بازه تعیین شده برای این هایپرپارامتر موارد ذکر شده در توضیح تابع هسته است. (مقدار بهینه : خطی)

ج-۴) بایاس و واریانس دو مفهوم کلیدی در یادگیری ماشینی هستند که به توانایی یک مدل برای ثبت دقیق الگوهای اساسی در داده ها مربوط می شود. بایاس خطای حاصل از تقریب زدن مسائل دنیای واقعی با استفاده از مدل‌های ساده است. یک مدل با بایاس بالا تمایل به ساده‌سازی بیش از حد داده‌ها دارد و ممکن است الگوهای مربوطه را از دست بدهد که منجر به عدم تناسب می‌شود. از سوی دیگر، واریانس به حساسیت مدل به نوسانات کوچک در داده های آموزشی اشاره دارد. یک مدل واریانس بالا، نویز را در داده های آموزشی ذخیره می‌کند که منجر به بیش از حد برازش می شود.

اکنون، با مقایسه درخت تصمیم‌گیری[۸] و جنگل‌های تصادفی از نظر بایاس و واریانس، درخت تصمیم‌گیری معمولاً واریانس بالایی دارند زیرا تمایل دارند با گرفتن نویز همراه با الگوهای زیربنایی، داده‌های آموزشی را بیش از حد برازش دهند. این می تواند منجر به مدلی شود که به خوبی به داده های دیده نشده تعمیم نمی‌یابد. از سوی دیگر، جنگل‌های تصادفی، این واریانس بالا را با میانگین‌گیری پیش‌بینی‌های درختان تصمیم‌گیری چندگانه کاهش می‌دهند، در نتیجه باعث کاهش بیش‌برازش و بهبود عملکرد تعمیم می‌شوند. در نتیجه، جنگل‌های تصادفی اغلب از نظر بایاس و واریانس بهتر از درخت‌های تصمیم‌گیری فردی عمل می‌کنند و تعادل بهتری بین گرفتن الگوهای اساسی و تعمیم به داده‌های جدید ایجاد می‌کنند.

در نتایج حاصل از تربیت مدل نیز استدلال بالا قابل قبول است، زیرا طبق جدول۶، دقت مدل‌های جنگل تصادفی نسبت به درخت‌های تصمیم‌گیری بالاتر است.

[7] RBF
[8] Decision Tree