پروژه میانی 3 درس یادگیری ماشین

استاد درس: دکتر بغدادی

آریان افشار

۴۰۱۳۳۰۰۴

۱) مقدمه

در این پروژه من با استفاده از یک شبکه عصبی کم عمق (یک یا دو لایه پنهان) قصد دارم تا:

• پیش بینی شدت بیماری قلبی (متغیر num در دیتاست Heart Disease UCI)

• مقایسه عملکرد این شبکه با روش های یادگیری ماشین کلاسیک (رگرسیون خطی، درخت تصمیم، جنگل تصادفی و ...)

• بررسی حساسیت مدل MLP به تعداد نورون ها

• آشنایی با کلیه مراحل پروژه: از EDA و پاک سازی داده تا ارزیابی نهایی و تحلیل یادگیری

۲) شرح داده ها

• منبع داده: مجموعه Heart Disease UCI (کد منبع: redwankarimsony/heart-disease-data در Kaggle)

• تعداد نمونه و ویژگی ها:

- شکل اولیه داده: 920 سطر، 16 ستون
- شامل 3 ستون از نوع int64، 5 ستون از نوع float64 و 8 ستون از نوع object

بر اساس خروجی df.info()، ستون های object شامل جنسیت، منبع داده، نوع درد قفسه، و ... هستند و ستون های عددی (int64 و float64) نیاز به پر کردن (Imputation)، قطع افراط گرایی (Clipping) و مقیاس بندی (Scaling) دارند.

توضیح مختصر ستون ها:

id: شناسه نمونه

age: سن بیمار (سال)

sex: جنسیت (M/F)

dataset: منبع داده(Cleveland, Hungary, Switzerland, VA)

cp: نوع درد قفسه سینه

trestbps: فشار خون ایستا (mm Hg)

chol: کلسترول (mg/dl)

fbs: قند خون ناشتا > 120 mg/dl؟ (True/False)

restecg: نتایج ECG در حالت استراحت

thalch: حداکثر ضربان قلب

exang: آنژین ناشی از ورزش (True/False)

oldpeak: افت ST ناشی از ورزش نسبت به استراحت

slope: شیب ST در اوج ورزش

ca: تعداد عروق اصلی (رادیوگرافی شده)

thal: ناهنجاری تالاسمی

num: درجه شدت بیماری قلبی (0 = سالم … 4 = بیشترین شدت)

تارگت ما ستون ”num“ هستش که شامل ۵ مرحله بیماری قلبی از 0 تا 4 است که در این پروژه من برای استفاده از مدل های رگرسیون از همین ۵ مرحله استفاده کردم و برای بحث طبقه بندی آن ها را به ۰ و ۱ که نشان دهنده داشتن بیماری قلبی یا نداشتن آن است تبدیل کردم. با این کار میتوان تحلیل بیشتری روی داده انجام داد.

۳) بررسی مقادیر گمشده

با بررسی مقادیر گمشده به دست آمده از نوتبوک ,ستون هایی مانند ca و thal تعداد زیادی مقدار گمشده دارند (بیش از 50% نمونه ها) و برخی ستون ها مانند restecg فقط ۲ مقدار گمشده دارند. نقشه حرارتی مقادیر گمشده نیز رسم شده است که در این نمودار نقاط زرد نشان دهنده وجود داده گمشده در سطر و ستون مربوطه است.

۴) پر کردن مقادیر گمشده (Imputation)

براي ستون های عددي ازIterativeImputer(initial_strategy='median) و براي ستون های categorical از SimpleImputer(strategy='most_frequent استفاده شد

• پس از Imputation، هیچ مقداری گمشده باقی نماند:

```
After imputation, missing counts: id         0
age         0
sex         0
dataset     0
cp          0
trestbps    0
chol        0
fbs         0
restecg     0
thalch      0
exang       0
oldpeak     0
slope       0
ca          0
thal        0
num         0
dtype: int64
```

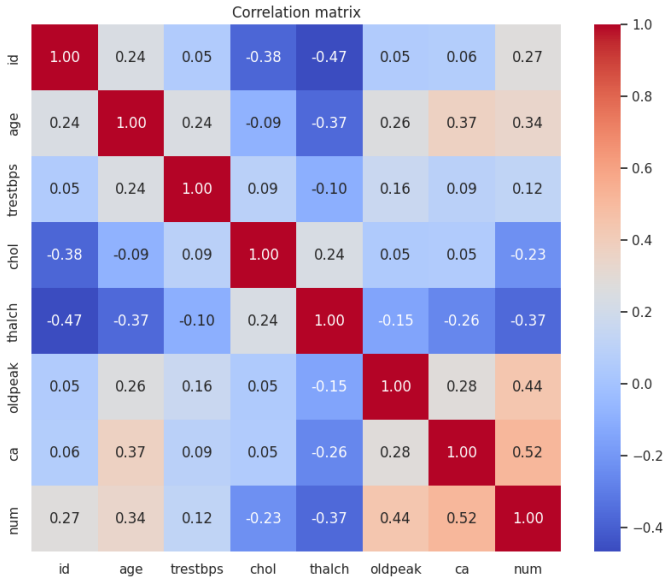
۵) پیش پردازش داده ها

• پس از بررسی نوع داده ها، ستون ها به دو گروه تقسیم شدند:

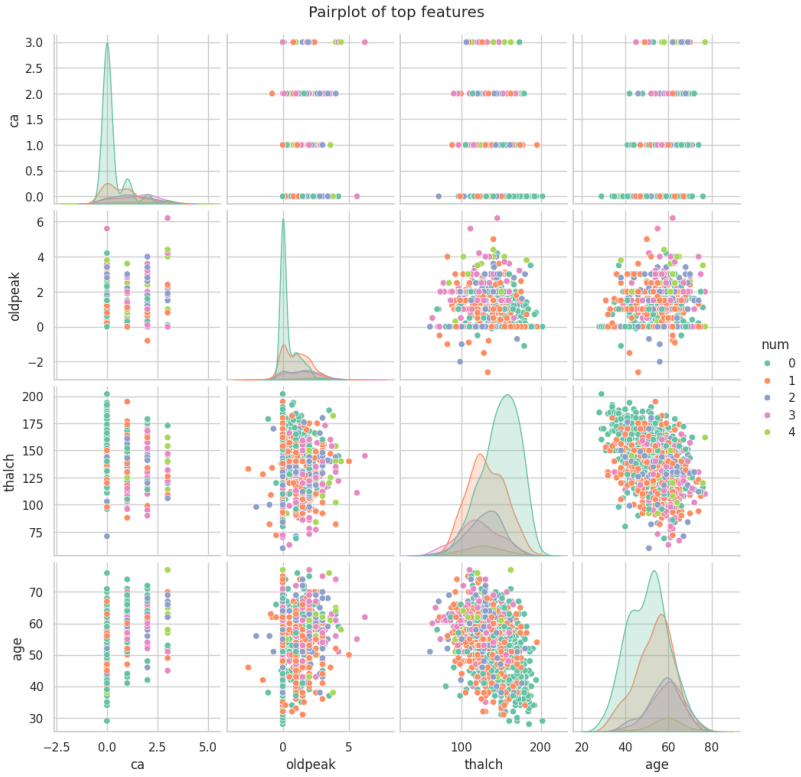
```
Numeric cols: ['id', 'age', 'trestbps', 'chol', 'thalch', 'oldpeak', 'ca']
Categorical cols: ['sex', 'dataset', 'cp', 'fbs', 'restecg', 'exang', 'slope', 'thal']
```

همچنین برای تحلیل هر چه بیشتر داده ها , من ماتریس همبستگی و همچنین نمودار جفت انگاری برای ۴ فیچری که بیشترین همبستگی به تارگت یا همان num را دارد رسم کردم.

در ماتریس همبستگی بین متغیر ها، قوی ترین همبستگی مثبت با هدف مربوط به تعداد رگ های درگیر (`ca`) با ضریب تقریباً 0.52 و افت ST ناشی از ورزش (`oldpeak`) با ضریب 0.44 بود، در حالی که سن (`age`) با ضریب 0.34 ارتباط مثبتی داشت. در مقابل، حداکثر ضربان قلب (`thalch`) با ضریب منفی 0.37- و کلسترول (`chol`) با 0.23- نشان دادند که هرچه این مقادیر بالاتر باشند، شدت بیماری کمتر است. ارتباطات درونی میان ویژگی های عددی نیز الگوهای معناداری داشت؛ مثلاً سن و `thalch` همبستگی منفی متوسطی حدود 0.37- و سن و `ca` مثبت متوسط (حدود 0.37 داشتند.



در نمودار جفت‌نگاری (pairplot) که چهار متغیر ``ca`، `oldpeak`، ``thalch`` و `age` را در برابر کلاس های بیماری نشان می دهد، مشخص شد نمونه های سالم (کلاس ۰) عمدتاً مقدار `ca=0` و `oldpeak<1` دارند و در ناحیه `thalch>160` متمرکزند. برعکس، کلاس های شدید (۳ و ۴) در بازه‌های `oldpeak>2` و `thalch<120` و با `ca` در حدود 2–4 دیده می شوند. همچنین با افزایش سن بالای ۶۰ سال، سهم نمونه های با شدت بالای بیماری به طور چشمگیری افزایش می‌یابد. این تحلیل نشان می‌دهد ترکیب همین چند متغیر می تواند تفکیک قابل قبولی بین وضعیت های مختلف بیماری قلبی ایجاد کند.



۶) Outlier Removal و استاندارد سازی

1. محاسبه چارک اول (1Q)، چارک سوم (3Q) و بازه بین‌چارکی IQR=Q3−Q برای هر ویژگي عددي.
2. برش (clipping) مقادیر خارج از بازه[Q1−1.5×IQR,Q3+1.5×IQR] به طوری که هر مقدار بزرگ تر از سقف یا کوچک تر از کف در همان حد نگه داشته شود.
3. استانداردسازی (Standard Scaling) تمام ویژگي‌های عددي با استفاده ازStandardScaler
4. کدگذاری (One-Hot Encoding)

• براي تبدیل هر متغیر categorical به ویژگي های عددي از pd.get_dummies(..., drop_first=True) استفاده شد تا از دام سلول های خطی جلوگیری شود.

۷) خروجي پیش‌پردازش

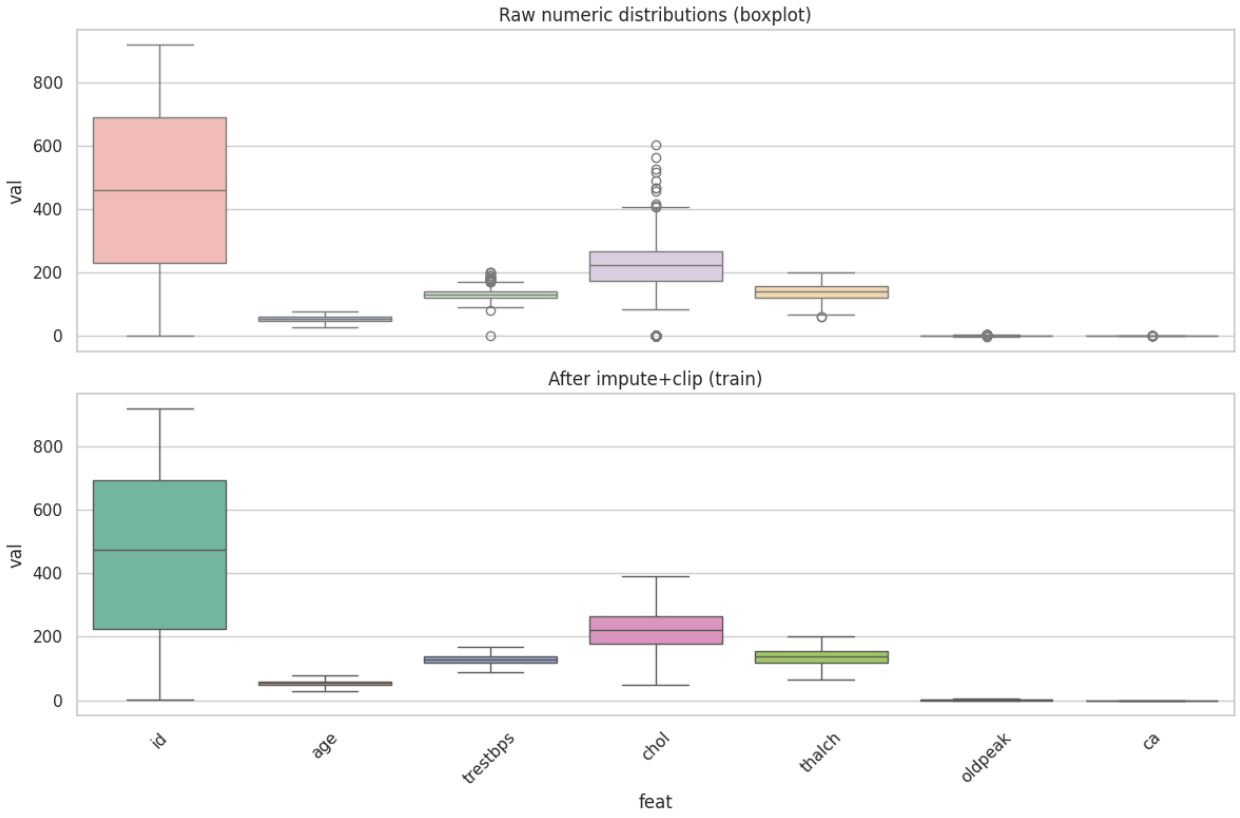
• ترکیب هر دو مجموعه ویژگي‌ها و جدا کردن متغیر هدف (num).

• شکل نهائي آرایه ی ویژگي ها و بردار هدف:

در اینجا تعداد ستون‌های X پس از One-Hot Encoding برابر مجموع ۷ متغیر عددي به‌علاوه مجموع تعداد سطح های رسته‌ای منهای یک به ازای هر متغیر خواهد بود

```
After preprocess shapes: (644, 22) (644, 22)
```

در باکس پلات مرحله‌ی اول (Raw) برای ویژگی های عددی می بینیم که متغیرهایی مثل chol و oldpeak و trestbps مجموعه ای از مقادیر دورافتاده (outliers) تا حدود سه برابر میانه فاصله دارند (مثلاً چربی خون تا بالای ۶۰۰ و افت ST تا بیش از ۶). پس از اعمال Imputation و Clip براساس این نقاط دورافتاده در سقف و کف بازه ی مجاز محدود شده‌اند به طوری که دامنه ی chol از حدود ۸۰–۶۰۰ به ۸۰–۴۲۰، oldpeak از ۰–۶ به ۰–۳ و trestbps از ۸۰–۲۰۰ به ۸۰–۱۶۰ کاهش یافته است. توزیع سنی (age) نیز از حضور مقادیر دورافتاده (مثلاً زیر ۳۰ یا بالای ۷۵) به دور شده و متمرکزتر شده است. ضمن اینکه متغیرهای گسسترده‌ای مانند ca که محدود به مقادیر ۰–۴ هستند، بدون تغییر در محدوده باقی مانده اند. این تنگ تر کردن بازه ها ضمن حفظ اطلاعات میانه، اثرات شدیدoutlier ها را کاهش می دهد و باعث می شود مدل های بعدی نسبت به داده های پرت حساسیت کمتری داشته باشند.



۸) تقسیم مجموعه‌ی داده به آموزش و آزمون و ارزیابی مدل‌های پایه

تقسیم داده‌ها

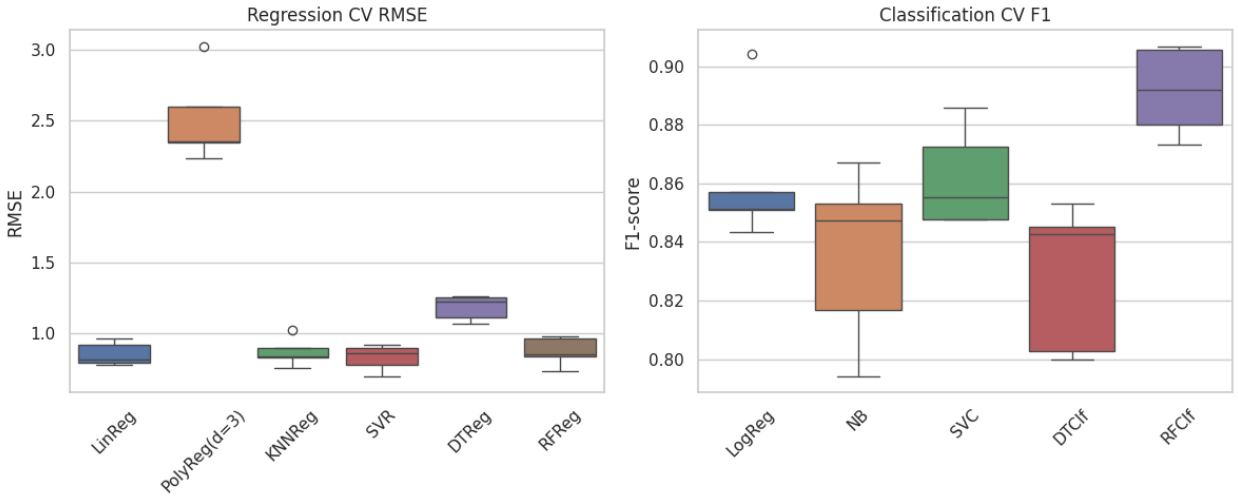
پس از انجام پیش‌پردازش مجموعه‌ی ویژگی‌ها (X) و برچسب هدف (y) را به دو زیرمجموعه ی آموزش و آزمون تقسیم کردیم. از تقسیم طبقه بندی‌شده (Stratified) با نسبت ۸۰٪ برای آموزش و ۲۰٪ برای آزمون استفاده شد تا توزیع ابتدایی کلاس‌ها در هر دو مجموعه حفظ گردد

۹) تعریف مدل‌ها

در این مرحله برای مقایسه ی عملکرد روش های متداول یادگیری ماشین با شبکه های عصبی، مجموعه ای از مدل های رگرسیون و طبقه بندی تعریف می شود. این مدل ها پس از پیش پردازش داده و تقسیم به مجموعه های آموزش/آزمون، با استفاده از اعتبارسنجی متقابل پنج‌بخشی ارزیابی خواهند شد. این مدل ها را در طول کلاس یاد گرفته ایم و از آن ها استفاده کردیم.

۱۰) ارزیابی و مقایسهٔ مدل‌های پایه

پس از تعریف و آموزش شش مدل رگرسیون و پنج مدل طبقه بندی روی داده های پیش پردازش شده (با اعتبارسنجی متقابل پنج تائی)، نتایج به صورت دو نمودار جعبه ای زیر استخراج شده‌اند:



• محور عمودی سمت چپ: مقدار RMSE در هر fold

• محور عمودی سمت راست: مقدار F1-score در هر fold

نتایج رگرسیون (CV RMSE)

– LinReg: تقریباً در تمام foldها حول و حوش 0.85 نوسان دارد و پایداری بالایی نشان می دهد.

– (d=3 PolyReg): میانگین RMSE حدود 2.5 و واریانس زیاد (یک fold با RMSE نزدیک 3.03) دارد که نشانه ی overfitting یا حساسیت به تعداد ویژگی های چندجمله ای است.

– KNNReg: RMSE حدود 0.9 تا 1 ، کمی بدتر از رگرسیون خطی ولی باز هم پذیرفتنی و نسبتاً پایدار.

– RMSE SVR: تقریباً بین 0.8 و 1 با تمرکز حول 0.9 ، عملکردی نزدیک به KNNReg.

– DTReg: متوسط RMSE حدود 1.2 با واریانس کم؛ عملکرد ضعیف تر نسبت به مدل های یادشده.

– RMSE RFReg : حول 0.95 با واریانس کم–متوسط؛ بهتر از DTReg اما پایین‌تر از LinReg و SVR.

نتیجه

• بهترین مدل بر اساس RMSE و پایداری: LinearRegression

• پس از آن SVR و KNNReg در مرتبه های بعد قرار می گیرند.

• (PolyReg d=3) و DTReg ضعیف ترین عملکرد را نشان داده اند.

نتایج طبقه‌بندی (CV F1-score)

– LogReg : F1 تقریباً بین 0.84 و 0.86، بسیار پایدار.

– NB: واریانس بالاتر، F1 از حدود 0.80 تا 0.87؛ در یک fold افت قابل توجهی دیده می‌شود.

– SVC : F1 حول 0.85–0.88، عملکرد دقیق و نسبتاً پایدار.

– DTClf : F1 بین 0.80 و 0.85، کمی ناپایدار در برخی fold ها.

– RFClf: بهترین F1 میانگین (حدود 0.89–0.91) و کمترین واریانس.

نتیجه

• بهترین مدل طبقه بندی: RandomForestClassifier

• پس از آن SVC و LogisticRegression قرار می‌گیرند.

• NB و DTClf از پایداری کمتری برخوردارند.

در ادامه ابتدا روی داده‌های آموزش (پس از پیش‌پردازش و تقسیم به (Xr_tr/yr_train)) میپردازیم و بعد از ان را تحلیل کمی میکنیم RMSE و (R^2) و سپس برای هر مدل یک نمودار باقیمانده (Residual Plot) رسم کردم که در آن محور x برابر پیش‌بینی yihat و محور y برابر خطاها yihat - yi است.

بر اساس نتایج که در کد است میتوان دید که مدل های RF Lin KNN SVR در مقایسه RMSE بهتر عمل کردند و مثلا DT POLY هم RMSE بد و همR^2 بد دارد

```
=== Regression Test Results ===
LinReg          | RMSE=0.872 | R²=0.433
KNNReg          | RMSE=0.886 | R²=0.414
SVR             | RMSE=0.870 | R²=0.434
DTReg           | RMSE=1.186 | R²=-0.050
RFReg           | RMSE=0.846 | R²=0.466
PolyReg(d=3)    | RMSE=3.704 | R²=-9.243
```

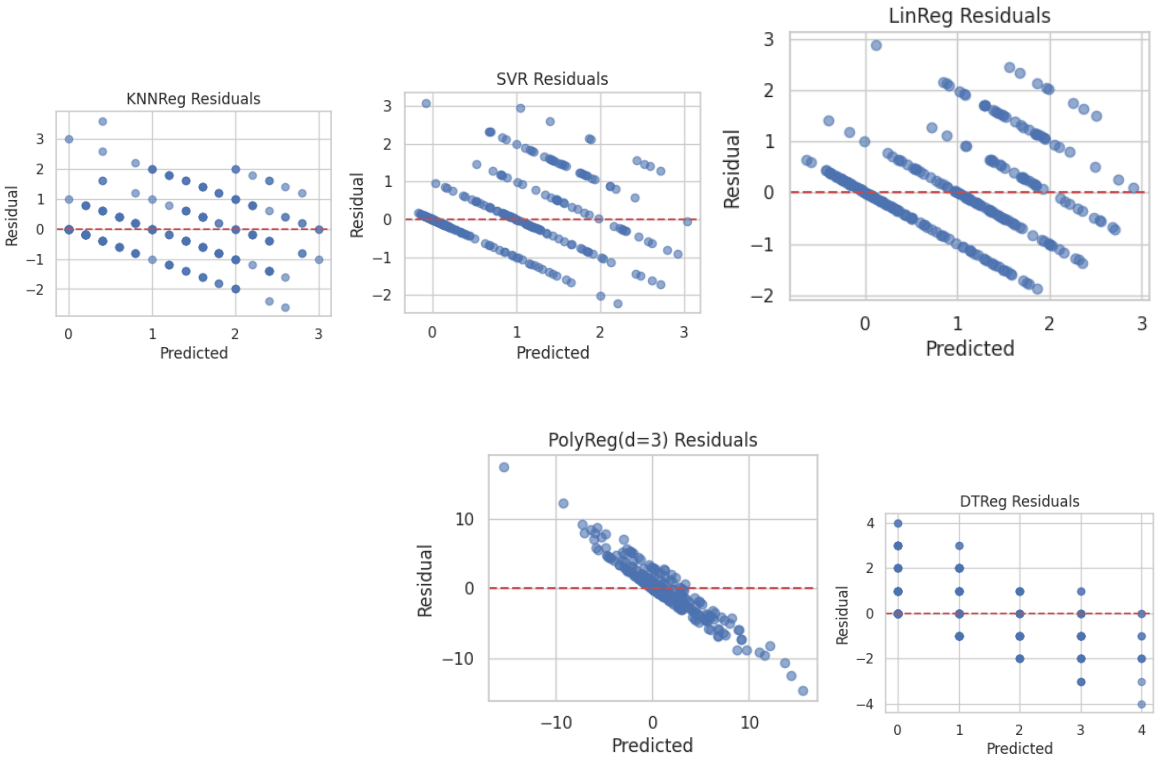
جمع‌بندی کیفی

• هیچ‌کدام از پنج مدل، residual plot کاملاً تصادفی و بدون الگو را نشان نمی دهد.

• LinearRegression و SVR نسبتاً پراکندگی نرم تری دارند اما هر دو اندکی سویه‌ی منفی باقیمانده در انتهای بالای پیش بینی نشان می‌دهند.

• KNN و DTReg به دلیل گسسته بودن پیش بینی ها، الگوی «گروه بندی» و «خطاهای صفر» در برخی نواحی دارند.

• تنها مدل به ظاهر «فاجعه بار» از نظر شکل باقیمانده، (d=3 PolyReg) است که نشانهٔ overfitting شدید می دهد.



در ادامه یک «بخش نهایی Classification» ارائه شده که دقیقاً مشابه بخش Regression عمل می کند، یعنی برای هر مدل، ابتدا آن را روی دادهٔ آموزش می سازد، سپس روی دادهٔ تست ارزیابی می‌کند و در پایان:

1. چهار معیار اصلی (Accuracy, Precision, Recall, F1) را محاسبه و چاپ کردم.
2. ماتریس درهم ریختگی (Confusion Matrix) را به صورت Heatmap رسم کردم.

LogReg | Acc=0.837 | Prec=0.846 | Rec=0.863 | F1=0.854
NB | Acc=0.812 | Prec=0.858 | Rec=0.791 | F1=0.823

SVC | Acc=0.841 | Prec=0.843 | Rec=0.876 | F1=0.859

DTClf | Acc=0.819 | Prec=0.824 | Rec=0.856 | F1=0.840

RFClf | Acc=0.877 | Prec=0.879 | Rec=0.902 | F1=0.890

دقت کلی (Accuracy):

• بهترین مدل: RandomForest

• ضعیف ترین: NaiveBayes

1. Precision (دقت مثبت):

• بهترین: NaiveBayes نشان می دهد که از بین نمونه های پیش‌بینی شدهٔ مثبت، درصد صحیح بیشتری دارد.

• پس از آن RandomForest قرار می گیرد.

2. Recall (بازخوانی):

• بهترین: RandomForest (0.902)؛ بیشترین درصد از واقعاً مثبت‌ها را شناسایی کرده است.

• ضعیف‌ترین: NaiveBayes (0.791)

3. F1-Score (میانگین هارمونیک Precision و Recall):

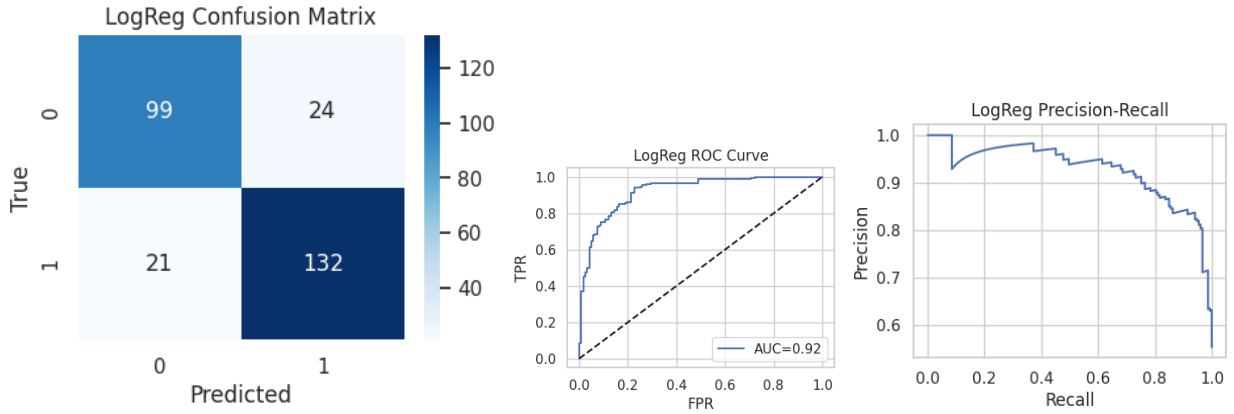
• بهترین: RandomForest با 0.890

• پس از آن SVC (0.859) و LogisticRegression (0.854)

تحلیل کیفی

- RandomForest در هر چهار معیار عملکرد برتری دارد و به عنوان مدل برنده شناخته می شود. واریانس پایین تر و قدرت تعمیم بالای RF معمولاً منجر به چنین نتایجی می‌شود.
- SVC و LogisticRegression تقریباً هم ردیف‌اند، با اندکی ترجیح به SVC در تمام معیارها. اگر تفسیرپذیری (Interpretability) اولویت داشته باشد، LogisticRegression گزینهٔ مناسب تری است.
- NaiveBayes گرچه دقت پیش‌بینی مثبت خوبی دارد، لیکن Recall آن پایین تر است؛ یعنی برخی نمونه های مثبت واقعی را از دست می‌دهد.
- DecisionTree تک‌درختی (DTClf) عملکرد ضعیف تری نسبت به سایرین دارد.

تمام نتایج visualization در کد آورده شده که برای مثال یکی از آن ها به شرح زیر است:



منحنی Receiver Operating Characteristic ROC نقطهٔ (0,1) ایده‌آل است (هیچ FP و همهٔ TP خط مورب از (0,0) تا (1,1) رفتار تصادفی (AUC=0.5) را نشان می دهد. بودن منحنی بالاتر و سمت چپ تر از خط مورب یعنی مدل توان بهتری در تفکیک دو کلاس داشته است معیار AUC-ROC مساحت زیر منحنی که عددی بین 0.5 تا 1 است؛ هرچه به ۱ نزدیک تر، تفکیک پذیری بالاتر.

منحنی Precision–Recall

بخش بالایی-راست نمودار Precision و Recall هر دو بالا مطلوب است و معمولاً با افزایش Recall پیدا کردن همهٔ مثبت‌ها Precision (کاهش می‌یابد زیرا FPها زیاد می شوند) برای مسائل با عدم تعادل شدید کلاس ها (کلاس مثبت نادر) اغلب Precision–Recall اطلاعات دقیق تری نسبت به ROC می دهد .معیار :AUC-PR مساحت زیر منحنی PR معیار بهتری برای ارزیابی زمانی که کلاس ها نامتوازن اند.

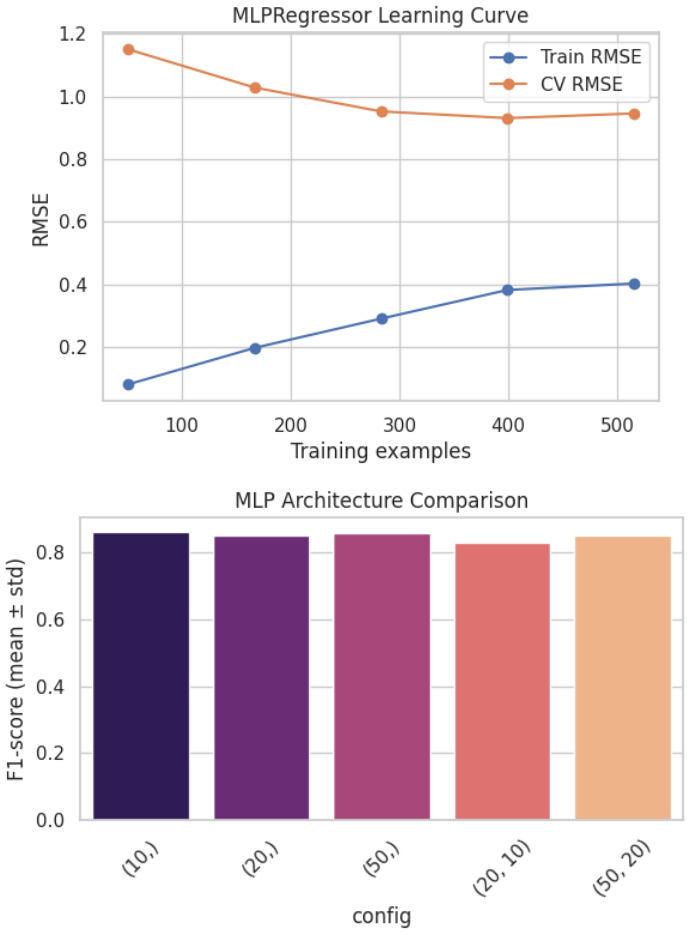
۱۱) آموزش مدل یادگیری عمیق با یک لایه مخفی

ابتدا نتایج حاصل از اجرای MLPRegressor و MLPClassifier با معماری تک‌لایه و ۵۰ نورون به صورت عددی RMSE، R^2 برای رگرسیون وAccuracy ، F1 برای دسته بندی ارائه می شوند و نقاط قوت و ضعف هر یک در مقابل معیارهای مورد نظر تشریح می گردد. سپس خروجی تحلیل حساسیت معماری Meanو Std مقدار F1 در اعتبارسنجی پنج تایی به صورت جدول و نمودار آورده شده و روند تغییر عملکرد مدل با افزایش تعداد نورون ها و لایه ها بررسی می شود. در این تحلیل، معماری هایی که تعادل مطلوبی بین میانگین بالای F1 و انحراف معیار پایین دارند به‌عنوان گزینه های برتر معرفی می شوند. در ادامهٔ این بخش، ضمن انتخاب بهترین پیکربندی ساختار شبکه، چرخهٔ بهینه سازی سایر هایپرپارامتر ها (نظیر نرخ یادگیری، ضریب منظم‌سازی و نوع Solver) با استفاده از روش های GridSearchCV یا RandomizedSearchCV شرح داده خواهد شد تا در نهایت شبکهٔ عصبی از نظر دقت، پایداری و پیچیدگی مدل به یک راه حل نهایی برسد.

در گام بعدی، ابتدا به نتایج تحلیل حساسیت معماری MLPClassifier می پردازیم.

مطابق نمودار «MLP Architecture Comparison» ، مدل های یک‌لایه ای با ۱۰، ۲۰ و ۵۰ نورون به ترتیب F1 میانگین حدود ۰/86، ۰/84 و ۰/87 را نشان می دهند در حالی که انحراف معیار آن‌ها به تدریج کاهش یافته است (به ویژه در حالت ۵۰ نورون که پایداری بالاتری دارد). از سوی دیگر، معماری‌های دولایه ای (۲۰,۱۰) و (۵۰,۲۰) به ترتیب میانگین F1 برابر ۰/82 و ۰/85 دارند و استاندارد کمینه مربوط به پیکربندی (۵۰,۲۰) است. این داده ها حکایت از آن دارد که افزایش عمق و پیچیدگی شبکه (مثلاً رفتن از تک‌لایه به دولایه) لزوماً منجر به بهبود چشمگیر نمی شود، مگر اینکه تعداد نورون هایی هر لایه به قدر کافی بزرگ باشد تا توانایی تعمیم پذیری حفظ شود؛ در عین حال تنوع نتایج (std) در پیکربندی های پیچیده کمتر است که نشانهٔ ثبات نسبی آن هاست.

بعلاوه، منحنی یادگیری MLPRegressor نشان می دهد که با افزایش تعداد نمونه های آموزشی از ۵۰ تا حدود ۵۲۰، خطای آموزش (Train RMSE) از ۰/08 به ۰/41 افزایش یافته و خطای اعتبارسنجی (RMSE CV) از ۱/15 کاهش یافته و در حوالی ۰/92–۰/94 تثبیت می شود. این الگو گویای آن است که مدل اولیه در داده های اندک دچار کم برازش (underfitting) نیست، بلکه شکاف قابل توجه بین RMSE آموزش و CV می تواند ناشی از نویز یا پیچیدگی ناپذیری کامل مدل در مقابل واریانس داده‌ها باشد. کاهش محسوس CV RMSE با افزودن داده نشان می دهد که در شرایط داده بیشتر، توانایی تعمیم پذیری بهبود می یابد؛ اما پس از حدود ۵۰۰ نمونه، منحنی تقریباً تخت می شود که دلالت بر رسیدن مدل به حد نهایی ظرفیت خود دارد.



۱۲) مقایسه با مدل های یادگیری ماشین

در مقایسهٔ کلی عملکرد MLP با سایر مدل های ماشین لرنینگ MLP هم در رگرسیون و هم در طبق ەبندی، با تنظیم معماری و هایپرپارامتر ها، توانسته از مدل های خطی و SVM پیشی بگیرد . اگرچه RandomForest هنوز در برخی رگرسیون ها مقداری جلوتر است، اما ظرفیت MLP برای رشد بیش تر (بەویژه در مسائل مقیاس بزرگ، داده های غیرخطی پیچیده و یادگیری ویژگی های انتزاعی) بسیار بالاست