

درس: علم داده

عنوان: پروژه نهایی (علم داده در تحصیل)

نام: علی پورقیصری

استاد: محمد‌مهدی رضاپور

شماره دانشجویی: 993613014

فهرست مطالب

[1- مقدمه 2](#_Toc168852795)

[2- مجموعه داده 3](#_Toc168852796)

[3- مدل سازی 8](#_Toc168852797)

[3-1- شبکه عصبی 8](#_Toc168852798)

[3-2- جنگل تصادفی 9](#_Toc168852799)

[3-3- جنگل تصادفی 10](#_Toc168852800)

[4- مراجع 12](#_Toc168852801)

1- مقدمه

در این گزارش قصد داریم که کد‌های همراه این فایل که پروژه پایانی درس علم داده است را شرح دهیم. در این گزارش سه نوع مجموعه داده استفاده شده است که هر بخش به ترتیب توسط علی پورقیصری، ملیکا آقاجانیان صباغ و مهدیس فتحی نجف آبادی نوشته شده است. در این گزارش از

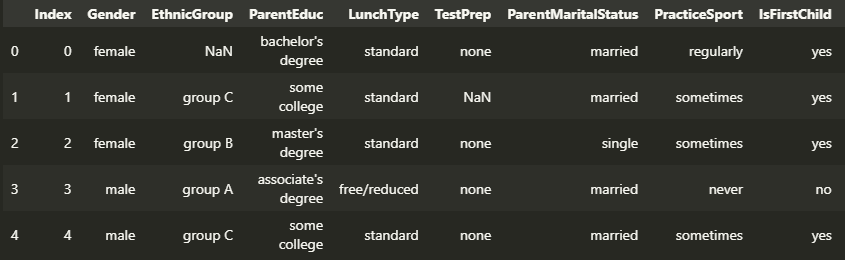
1. شبکه عصبی که یک مدل محاسباتی الهام گرفته شده از ساختار و عملکرد مغز انسان است و برای تشخیص الگوها و حل مسائل پیچیده طراحی شده است. از لایه های به هم پیوسته گره ها (نورون ها) تشکیل شده است، جایی که هر گره داده های ورودی را پردازش کرده و آن را از طریق یک تابع فعال سازی برای تولید خروجی عبور می دهد. شبکه یاد می گیرد که وظایف را با تنظیم وزن این اتصالات بر اساس خطای پیش بینی های خود در طول آموزش انجام دهد، فرآیندی که معمولاً توسط الگوریتم هایی مانند انتشار پس زمینه هدایت می شود. شبکه های عصبی به طور گسترده ای در کاربردهای مختلف از جمله تشخیص تصویر و گفتار، پردازش زبان طبیعی و تجزیه و تحلیل پیش بینی استفاده می شوند.

2- مجموعه داده

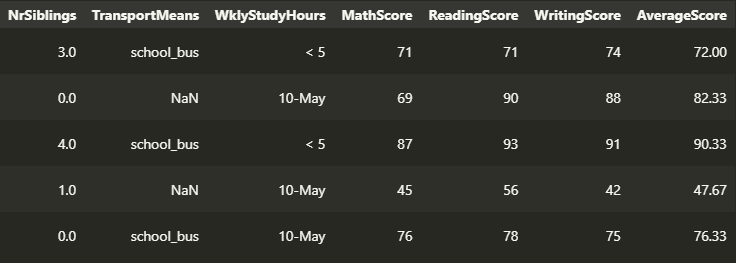
این مجموعه داده به نام Students test score: Extended Dataset یک مجموعه داده با داده های بیشتر برای مجموعه داده اولیه می‌باشد. این مجموعه داده شامل 30642 سطر و 16 ستون می‌باشد که به صورت زیر است.

1. شماره سطر
2. جنسیت
3. قومیت
4. تحصیلات والدین
5. نوع مدرسه
6. آزمون آمادگی
7. وضعیت تاهل والدین
8. تمرین ورزشی
9. آیا اولین فرزند است؟
10. تعداد خواهران / برادران
11. وسیله حمل و نقل
12. مجموع ساعات مطالعه هفتگی
13. نمره ریاضی
14. نمره خواندن
15. نمره نوشتن
16. نمره میانگین

این مجموعه داده نیاز به تمیز سازی و پیش پردازش دارد زیرا داده های خالی در بین این داده ها وجود دارد. این نوع داده ها را با چند روش می‌توان از بین برد که در ادامه توضیحات کافی را برای این بخش می‌دهیم.

در اینجا می‌توانیم محتویات 5 سطر اول را ببینیم تا به یک درکی از داده برسیم(هر دو عکس متعلق به 5 سطر واحد هستند).

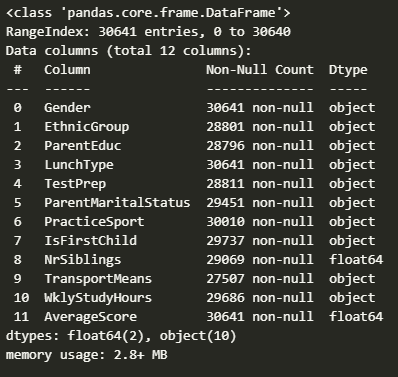
1- پنج سطر اول و 9 ستون اول مجموعه داده



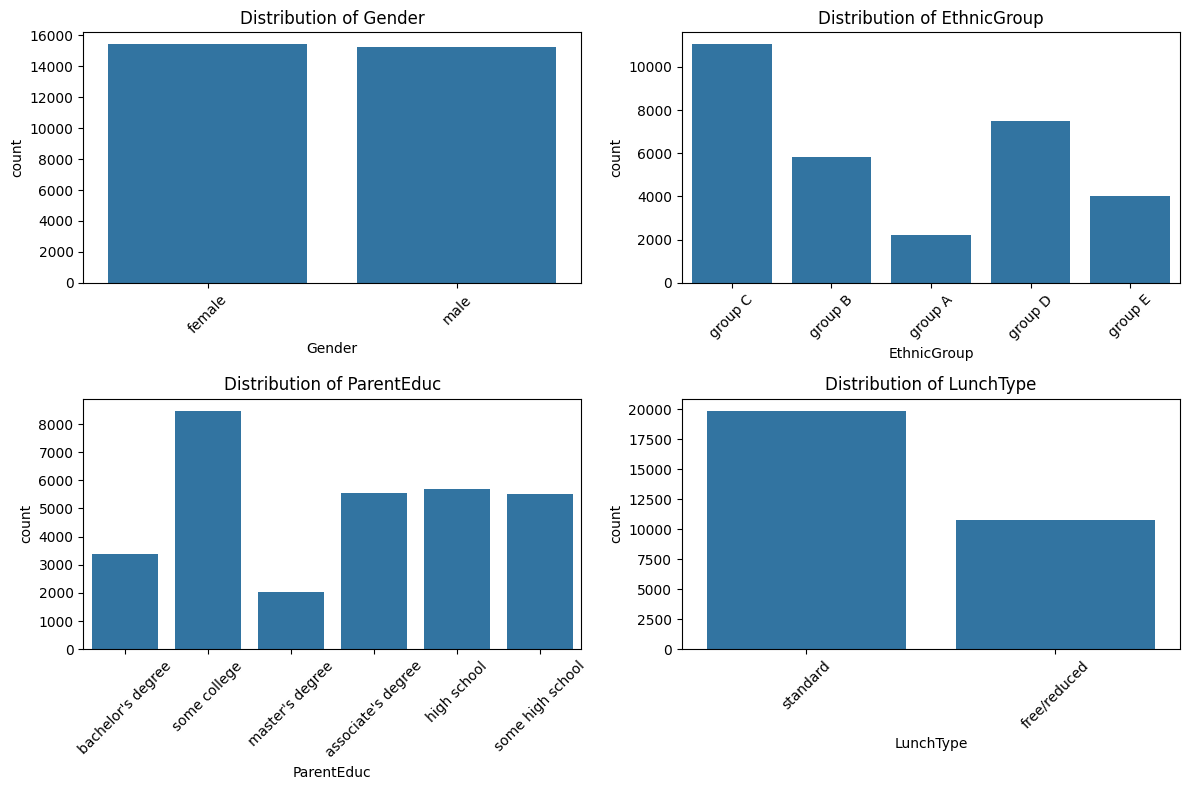
2- پنج سطر اول و 7 ستون دوم مجموعه داده

حال ستون‌هایی که به درد ما نمی‌خورند را حذف می‌کنیم. این ستون‌ها شامل شماره سطر، نمرات ریاضی، خواندن و نوشتن است زیرا نیاز به شماره سطر برای آموزش نداریم و همچنین برچسب نهایی که میانگین است، با احتساب سه سطر آخر به وجود آمده است.

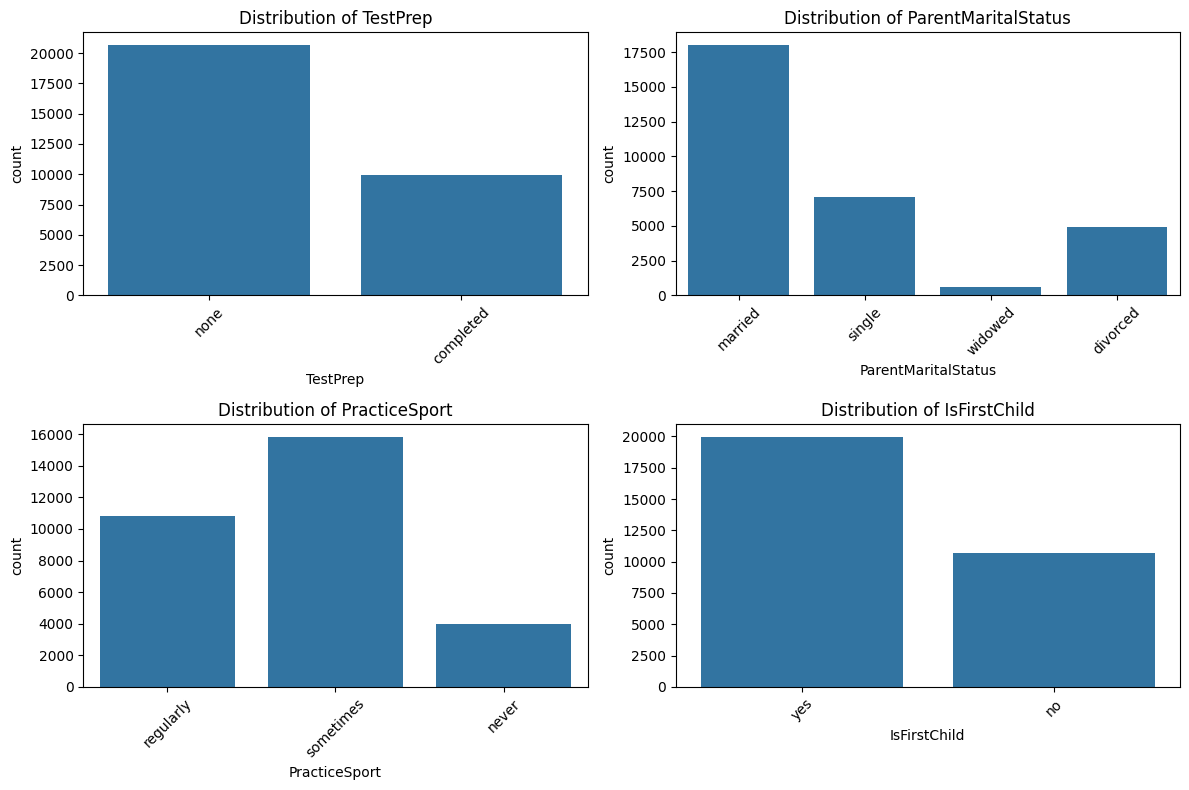
برای درک کلی مجموعه داده، در ادامه به نمایش ستون‌های این مجموعه می‌پردازیم. در ابتدا می‌خواهیم نوع ستون‌های مجموعه داده را ببینیم.



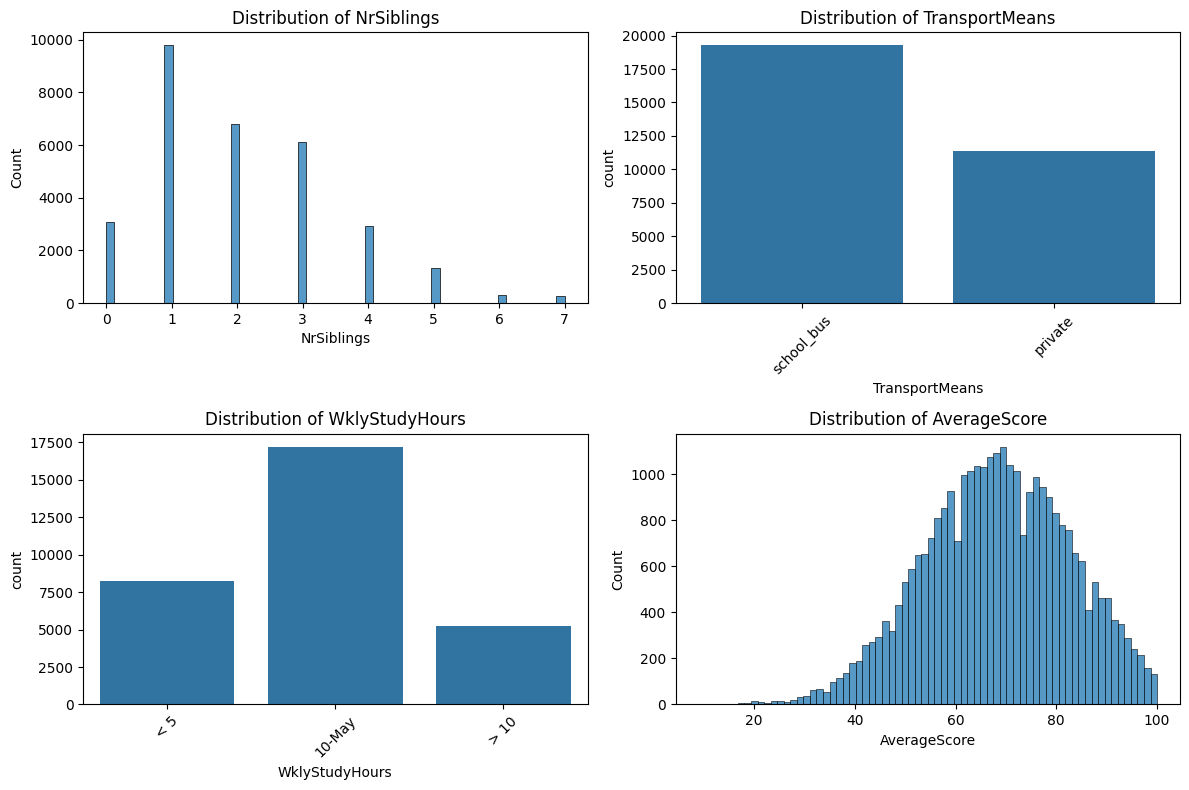
3- نوع هر ستون از مجموعه داده



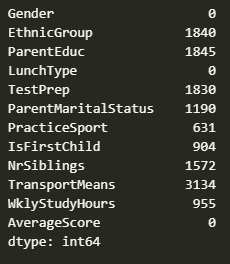
4- توزیع داده ستون های جنسیت، قومیت، تحصیلات والدین و نوع مدرسه



5- توزیع داده ستون‌های آزمون آمادگی، وضعیت تاهل والدین، تمرین ورزشی و آیا فرزند اول است

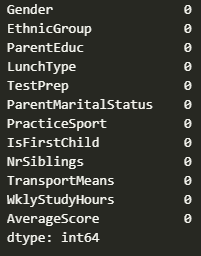


6- توزیع داده ستون‌های تعداد خواهران / برادران، وسیله حمل و نقل، مجموع ساعات مطالعه هفتگی و نمره میانگین

حال به سراغ داده‌های از دست رفته می‌رویم. اگر داده‌های از دست رفته را نشان دهیم، به صورت زیر هستند.

7- مجموعه داده های از دست رفته

این داده‌ها را در اینجا با استفاده از پر تکرار ترین مقدار، پر می‌کنیم. پس از این کار می‌توان دید که دیگر داده گم شده نداریم.



8- پر کردن داده های از دست رفته

حال تعداد سطر های تکراری را اندازه گیری می‌کنیم و می‌بینیم که 132 سطر تکراری داریم. اقدام به حذف این داده‌ها می‌کنیم. پس از اینکه این کار را کردیم نیازمندیم که مقادیر ستون‌های چند مقداری را به عدد تبدیل کنیم که این کار را با استفاده از labelencoder انجام می‌دهیم.

بعد از آن ستون‌ها را به داده‌های آموزشی و آزمایشی تقسیم می‌کنیم و آنها را با استفاده از StandardScaler که در کلاس هم گفته شد، تبدیل می‌کنیم. حال داده های ما آماده است و موقع آموزش و آزمایش مدل‌ها می‌باشد.

3- مدل سازی

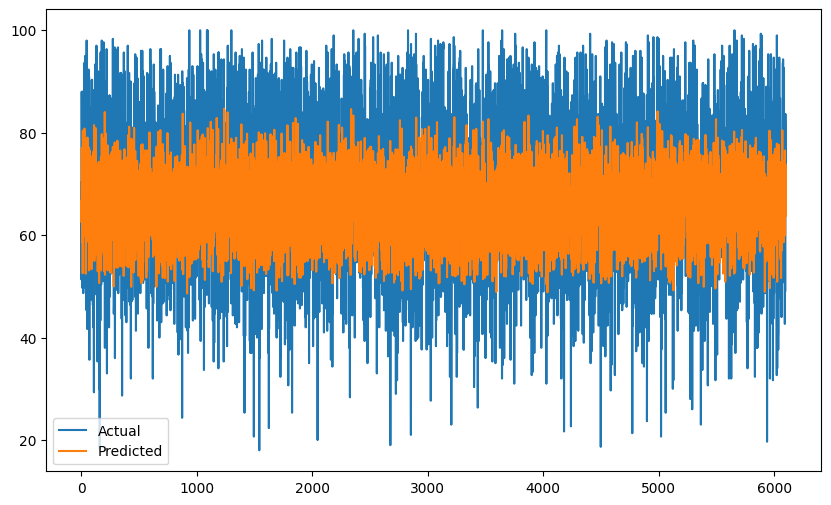
3-1- شبکه عصبی

با استفاده از کتابخانه keras یک شبکه عصبی به این صورت می‌سازیم که لایه اول 64 نورون ورودی و تابع فعال سازی relu دارد سپس 20 درصد dropout و لایه دوم 32 نورون با تابع فعال سازی relu و 20 درصد dropout، در آخر یک لایه تک نرونی با تابع فعال سازی خطی.

تابع هزینه ای که برای این شبکه در نظر گرفته شده MSE و بهینه ساز adam می‌باشد. تابع هزینه‌ چیزی است که مدل باید برای محاسبه خطا (error) از آن استفاده کند. تابع هزینه معیاری برای اندازه‌گیری تفاوت بین پیش‌بینی‌های مدل و مقادیر واقعی است. Adam مخفف Adaptive Moment Estimation یک الگوریتم بهینه‌سازی رایج و قدرتمند است که ترکیبی از روش‌های RMSProp و مومنتم (momentum) است و به طور خودکار نرخ یادگیری را تنظیم می‌کند.

با آموزش این مدل به RMSE = 13.01 و MAE = 10.59 دست می‌یابیم که با توجه به دیتاست و نتایج منتشر شده توسط کاربران در سایت کگل، نتیجه مطلوبی است.

در آخر نتیجه را به تصویر کشیدیم.



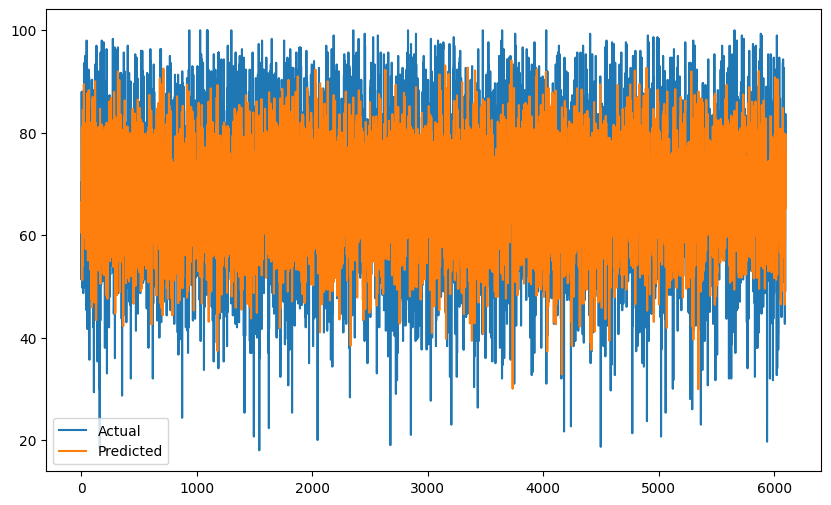
9- خروجی شبکه عصبی

3-2- جنگل تصادفی

در اینجا با استفاده از کتابخانه sklearn که یک کتابخانه یادگیری ماشین است، به کمک ما آمده است. در ابتدا مدل را تعریف کرده و دو متغیر n\_estimatiors که تعداد درخت های موجود در جنگل و random\_state که به اصطلاح بذر مولد اعداد تصادفی، برای اطمینان از تولید نتایج تکرارپذیر است.

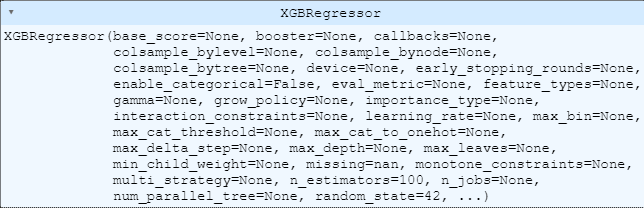
نتایج به دست آمده از این مدل هم RMSE = 13.89 و MAE = 11.22 است که این نیز با توجه به دیتاست و نتایج منتشر شده توسط کاربران در سایت کگل، نتیجه مطلوبی است.

تصویر به دست آمده از نتایج این مدل نیز به صورت زیر است.



10- خروجی جنگل تصادفی

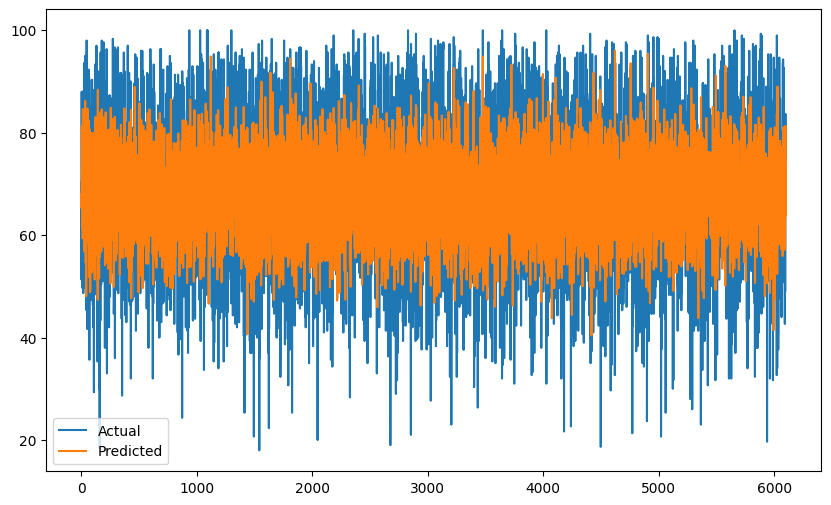
3-3- جنگل تصادفی

 در آخر با استفاده از کتابخانه XGboost مدل را پیاده سازی کرده که دو متغیر ورودی درست مثل جنگل تصادفی دارد. با آموزش این مدل، یکسری متغیر اولیه تایین می‌شود که به شرح زیر است.

11- مقادیر اولیه متغیر های XGBoost

نتایج به دست آمده از این مدل هم RMSE = 12.78 و MAE = 10.41 است که این نیز با توجه به دیتاست و نتایج منتشر شده توسط کاربران در سایت کگل، نتیجه مطلوبی است.

خروجی مقادیر پیش بینی شده این مدل هم به شرح زیر است.



12- خروجی XGBoost

4- مراجع

* [StudentPerformanceRegressor (kaggle.com)](https://www.kaggle.com/code/josephchan524/studentperformanceregressor-rmse-12-26-r2-0-26/comments#1198239)
* [Image classification with modern MLP models (keras.io)](https://keras.io/examples/vision/mlp_image_classification/)
* [RandomForestClassifier — scikit-learn 1.5.0 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html)
* [GradientBoostingClassifier — scikit-learn 1.5.0 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html)