Final Project: Diamond Price Prediction

BIHE

Negar Ighani

Summer Fall 2022

Pre-processing

در این مرحله همانطور که خواسته شده بود به آماده سازی دیتا برای آنکه بتوان در مراحل بعدی مدل های مختلف را بر روی آن اجرا کرد پرداخته شد. به این منظور اگر داده ای null بود تلاش شد پیدا شود و ردیف مربوط به آن داده پاک شود. همچنین ستون هایی که داده های آن ها در دسته بندی های مختلف قرار میگرفت یعنی cut, color & clarity به جای آنکه به شکل string داده هایشان نگه داری شود به ازاء هر دسته بندی عددی معرفی شد تا داده های مربوطه به آن ها map شوند.

به صورت کلی میتوان گفت که مسئله طرح شده در پروژه یعنی حدس زدن قیمت الماس با توجه به داده ی در نظر گرفته شده که دارای لیبل نهایی مورد نیاز ما یعنی قیمت هستند در دسته ی supervised learning قرار میگیرد زیرا در دیتا ست داده شده هم feature های مسئله که رنگ و عمق و باقی موارد را در برمیگیرد به ما داده شده و هم label نهایی و هدف ما آن است که بتوانیم رابطه ی داده های ارائه شده را پیدا کنیم در غالب تابعی که بر اساس متغیر های مسئله ی ما باشد . حال با توجه به اینکه لیبل نهایی این مسئله که قیمت است یک عدد حقیقی پیوسته است میتوان گفت که این مسئله یک مسئله ی regression است.

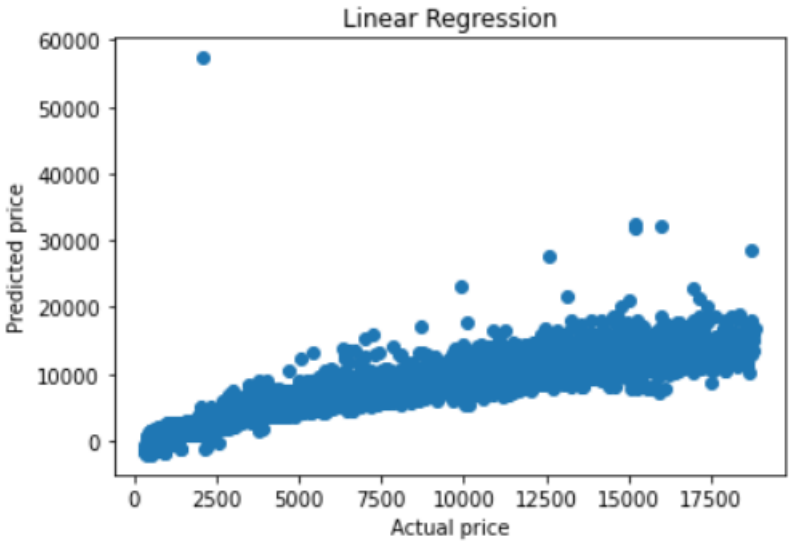
در ادامه ی توضیحات بالا پس در ادامه ی آماده کردن داده ها لیبل نهایی یعنی قیمت را جدا کرده و سپس به نسب خواسته شده و به کمک متد train_test_split تلاش شد در ابتدا داده های مربوط به تست جدا شود و از باقی داده ها train و validation را نیز مشخص شد. در نهایت برای بخش feature engineering تلاش شد که داده ها استاندارد شوند.

Learning Models

1.linear regression

اولین مدل امتحان شده linear regression بود که خروجی های مربوط به این مدل در زیر قابل مشاهده است.

```
Linear Regression:
Mean absolute error: 843.7189641060797
R2 score: 0.882143275970175
```

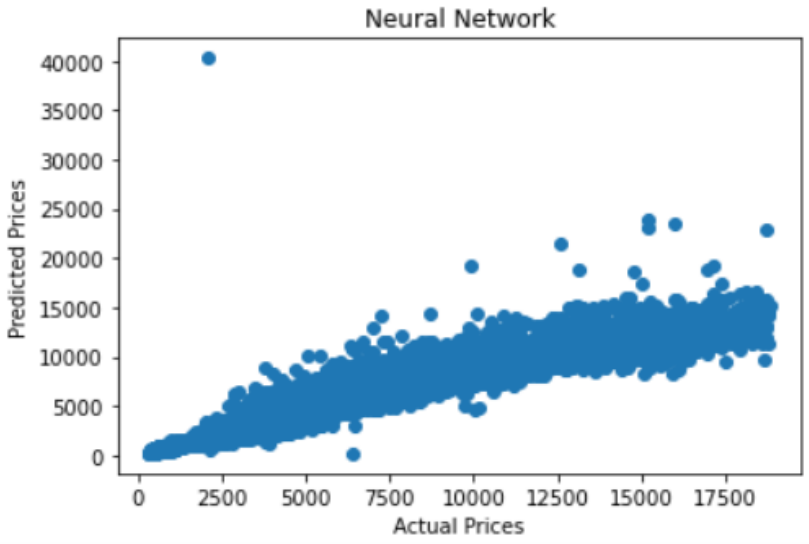


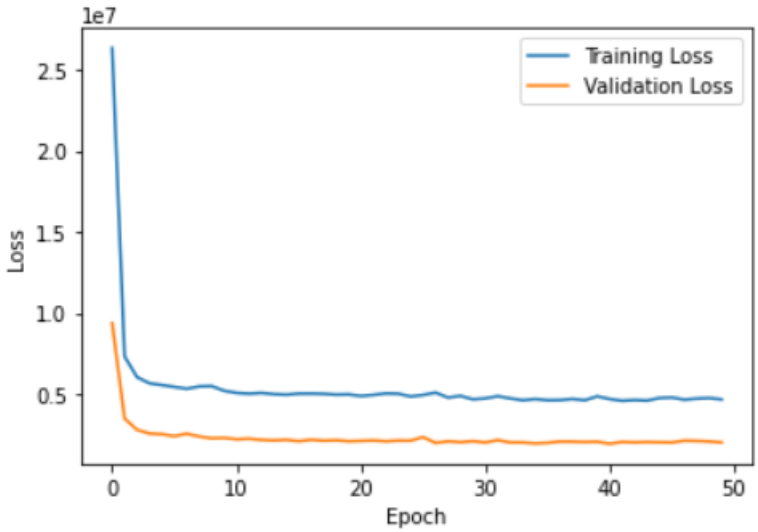
در این مدل تلاش میشود تا رابطه ی داده ها به صورت یک تابع خطی در نظر گرفته شود و بر آن اساس یک تابع تولید شود که داده های تست با قرار گرفتن در آن لیبل پیش بینی شده را تولید نمایند. استفاده از این مدل برای حل مسئله ی ارائه شده یکی از راه های منطقی بود هر چند که پیش فرض آن این است که رابطه ی متغیر های مسئله پیچیده نیست و قابل مدل سازی در غالب تابع خطی است و در صورتی که با رابطه ی پیچیده ای رو به رو باشیم احتمالا جواب خوبی نمیتوان از این مدل گرفت هر چند که عملکرد خوبی در اینجا داشته و امتیاز r2 آن ۰.۸۸ شده است.

2.Neural Network

مدل بعدی که امتحان شد NN بود که خروجی آن در زیر قابل مشاهده است:

```
Mean Absolute Error: 717.424077099555
R2 Score: 0.9016333771560954
```





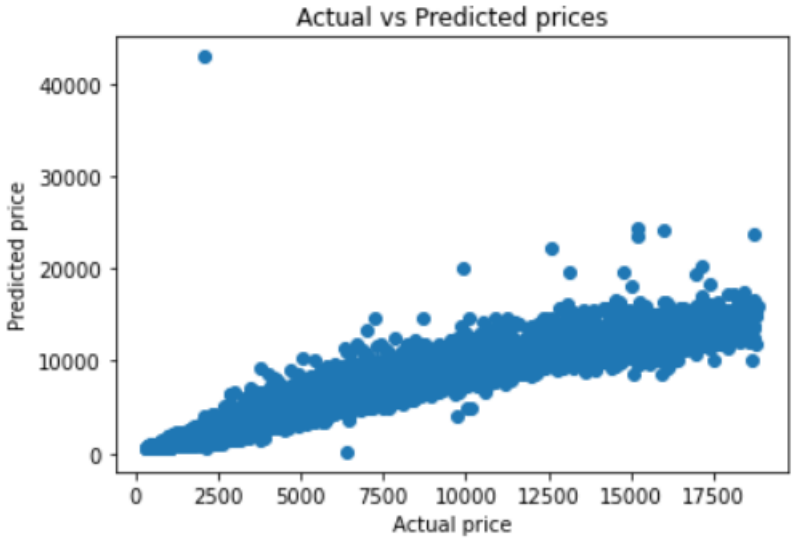
به صورت کلی میدانیم که شبکه های عصبی این امکان را به ما میدهند تا بتوانیم ارتباطات پیچیده تری را بین feature ها و خروجی در نظر بگیریم به نوعی که لزوما مانند regression خطی نباشد. بنابراین طبق گفته ی صورت پروژه برای آنکه مدل مد نظر ساخته شود لایه های مختلفی در نظر گرفته شده است. به این صورت که ابتدا یک sequential model ساخته شده و بعد لایه ی ورودی و سپس دو لایه ی hidden و در نهایت لایه ی خروجی در نظر گرفته شده است. بین این لایه های برای جلوگیری از overfitting دو لایه ی dropout در نظر گرفته شده است. در واقع ممکن است بدون در نظر گرفتن آن مدل تولید شده بیش از حد بر اساس داده هایی باشد که روی آن train شده باشد و برای داده های جدید به خوبی عمل نکند . این لایه ها به این صورت عمل میکنند که برای هر دوره برای تعدادی از نود ها احتمال drop شدن در نظر میگیرد که معمولا این احتمال بین ۰.۲ تا ۰.۵ در نظر گرفته میشود که در این پروژه ۰.۲ مقدار دهی شده است. بر اساس این احتمال تعدادی از نود ها وزن ۰ در آن دوره ی train میگیرند و سبب میشود که هر نود بتواند مستقل از دیگر نود ها بتواند به داده های ورودی مختلف پاسخ دهد و وابستگی را تا حد ممکن کاهش دهیم تا overfitting رخ ندهد. همچنین همانطور که در نمودار دوم مشخص است عملیات بهینه سازی و تلاش برای کاهش loss در طی epochهای مختلف که ما اینجا ۵۰ در نظر گرفتیم آورده شده است. در نهایت همانطور که مشخص

است این مدل توانسته امتیاز r2 کمی بالا تر از regression را به دست بیاورد زیرا توانسته پیچیدگی های داده را بهتر مدل سازی کند.

2.1Parameter Tuning

در مدلی که برای NN ساخته شده بود hyperparameter های مختلفی وجود داشت که مقدار هر یک میتوانست بر دقت مدل تولید شده تاثیر گذارد. در این قسمت به کمک GridSearchCV ۴ عدد از پارامتر های مدل را tune کردیم تا ببینیم بهترین نتیجه با چه مقادیری برای پارامتر ها به دست می آید که طی آن عملکرد مدل از حالت امتحان شده ۱ در صد افزایش یافت. مقادیر نهایی که بهترین نتیجه را به همراه داشتند در زیر آورده شده است.

```
Mean absolute error: 666.1416338464088
r2 score: 0.9151151886677592
```

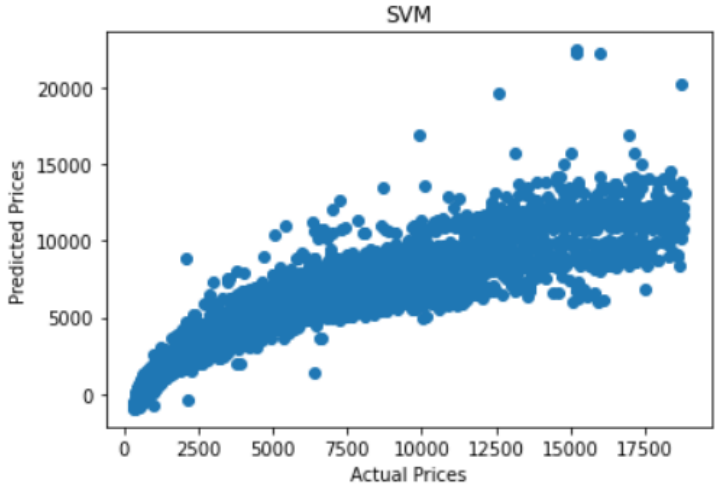


Best score by using: {'activation': 'relu', 'kernel_initializer': 'RandomNormal', 'num_hidden_layers': 1, 'num_units': 6}

3.Support Vector Machine

مدل سومی که پیاده شد SVM بود که خروجی آن در زیر قابل مشاهده است.

```
Mean Absolute Error: 897.8454664101046
R2 Score: 0.8410213259144209
```



در مدل تولید شده از SVR استفاده شده که دارای hyperparameter های مختلفی است از جمله C و epsilon. در واقع نقش اپسیلون آن است که اگر تفاوت بین مقدار y حدس زده شده از واقعی آن کم تر از اپسیلون باشد هیچ اروری در نظر گرفته نمیشود و بدون در نظر گرفتن پنالتی آن مقدار قبول میشود. C نیز میتواند مشخص کننده ی یک trade-off باشد بین نکه در train ارور کمی داشته باشیم و در test . در واقع اگر این مقدار کم باشد مرزی که تعیین میکنیم بسیار گسترده است و شاید سبب شود که داده هایی با ارور بالا پیش بینی شوند اما منجر به داشتن مدلی میشود که احتمال overfit آن کم اما از طرف دیگر احتمال underfit آن زیاد است. از طرف دیگر اگر مقدار آن را بزرگ تر نظر بگیریم مرزی که تعیین میکنیم بسیار محدود تر است و سبب میشود بیشتر نقاط را به صورت درست و با ارور کم تری پیش بینی نماییم اما از طرف دیگر احتمال overfit آن بسیار بالا تر است. بنابراین با توجه به توضیحات بالا تلاش شده یک مقدار معقولی برای این دو در نظر گرفته شود و سپس مدل ساخته و پیش بینی نهایی آن به دست آمده است. همانطور که از امتیاز نهایی آن مشخص است هر چند که از مدل های دیگر امتیاز پایین تری به دست آورده اما عملکرد ضعیفی نداشته است و قابل قبول بوده است.

4. Clustering

مدل آخری که پیاده سازی شد clustering بود که نتایج آن در زیر آورده شده است.

```
cluster
0    1974.456452
1    3950.056897
2    7527.264522
3    2848.119185
Name: price, dtype: float64
```

این روش به صورت کلی یک روشunsupervised است و تلاش میشود که داده ها را بر اساس شباهت هایشان دسته بندی کند. هر چند که به صورت کلی با توجه به اینکه label نهایی داده ها یعنی قیمت آن ها در دیتاست ما موجود است و میتوانیم به روش supervised که در بسیاری از موارد از دقت بالاتری برخوردار است استفاده کنیم و به صورت کلی این مسئله جزو این دسته از مسائل نیست اما این روش نیز امتحان شده است. دسته بندی بر اساس feature های الماس ها صورت گرفته است و دسته بندی مد نظر ماست که در نهایت بتوانیم بگوییم بر اساس ویژگی ها قیمت آن دسته حدودا شبیه هم هست. در این حالت باید بین میانگین قیمت دسته های مختلف تفاوت معقولی وجود داشته باشد و اگر قیمت های دسته بندی ها مثل هم یا نزدیک به هم باشد به این معناست که دسته بندی خیلی خوب صورت نگرفته است. برای تعیین تعداد دسته بندی ها در این مرحله اینگونه عمل کردم که اعداد مختلف را امتحان کردم و میانگین قیمت دسته بندی ها را مقایسه کردم و در نهایت تعدادی که منطقی ترین جواب را داشت نگه داشتم. همانطور که در عکس بالا و نتیجه ی نهایی مشخص است اگر الماس ها را در ۴ دسته قرار دهیم میانگین قیمت ها به شکل بالا خواهد بود که خیلی هم جواب دقیقی و دسته بندی خوبی به نظر نمیرسد اما تعداد هر چه بالا تر میرفت میانگین قیمت ها بسیار به هم نزدیک میشدند که نشان دهنده ی آن بود که دسته بندی به خوبی عمل نکرده و دسته ها مشابه هم هستند.

5.Analysis of Results

در نهایت نتیجه ی نهایی به این صورت است که برای استفاده از روش های مختلف مهم است که بدانیم با چه مسئله ای رو به رو هستیم تا بتوانیم از متد های مناسب آن استفاده کنیم. مسئله ی ما یک مسئله ی regression بود که برای حس این مسائل استفاده از NN, SVM & Linear Regression راه های منطقی به نظر میرسند. هر ۳ این روش ها در مسئله ای که با آن رو به رو بودیم عملکردی قابل قبول داشتند و امتیاز r2 آن ها بالا تر از ۸۰ بود. به ترتیب NN سپس Linear regression و در نهایت SVM بهترین مدل ها را برای ما فراهم کردند. با tune کردن پارامتر های مسئله و انتخاب بهترین پارامتر ها با توجه به داده ها در NN حتی پاسخ آن بهتر نیز شد. در نهایت clustering نیز با دسته بندی بر اساس feature های مسئله صورت گرفت که جواب نهایی آن خیلی دقیق نبود و به نظر میرسد این نوع دسته بندی بر اساس ویژگی ها نمیتواند پاسخ گوی پیچیدگی داده هایی که با آن مواجه هستیم باشد.