

پروژه کارشناسی



Exploratory Analysis and Machine Learning-Based Modeling of the Divar Real Estate Market

تحلیل اکتشافی، پاکسازی داده و مدل‌سازی قیمت املاک در بازار دیوار با استفاده از یادگیری ماشین

استاد راهنمای پروژه : خانم دکتر زهرا صابری

دانشجو : ابوالفضل عزیزی

Github Repositories for this project:

<https://github.com/azizialborz25/divar-real-estate-analysis>

تمام فایل های مربوط به این پروژه اعم از notebook ها ، report ها و دیگر موارد؛
در گیت هاب بارگذاری شده اند و از طریق لینک فوق قابل دسترسی هستند.

تمامی کد های این پروژه که در اختیار شما قرار گرفته اند ، طوری نوشته شده اند که بدون مشکل و به راحتی میتوانید
آنها را اجرا کنید و خودتان خروجی ها را مشاهده کنید.

فهرست مطالب

فصل اول	7
شناخت مسئله، تعریف هدف و بررسی اولیه داده‌ها	7
1-1 مقدمه	7
1-2 تعریف مسئله	7
1-3 راه‌اندازی محیط پروژه	8
1-4 بارگذاری و بررسی اولیه dataset	8
1-5 پروفایل اولیه ویژگی‌ها	9
1-6 شناخت ساختار کسب و کاری داده	9
1-7 تعریف اولیه target	10
1-8 انتظار اولیه در برابر واقعیت داده	10
1-9 خروجی فصل اول	11
فصل دوم	12
تحلیل اکتشافی داده‌ها	12
2-1 مقدمه	12
2-2 بررسی اولیه کیفیت داده	12
2-3 بررسی Missing Value ها	13
2-4 بررسی توزیع قیمت فروش مسکونی	15
2-5 بررسی ویژگی‌های ساختاری و عوامل مؤثر بر قیمت	17
2-6 بررسی اطلاعات تعداد اتاق	20
2-7 بررسی construction year	21
2-8 نتیجه کلی EDA	22
2-9 جمع‌بندی تصمیم‌های Phase 2	23
فصل سوم	24
پاکسازی داده‌ها و مهندسی ویژگی	24
3-1 مقدمه	24
3-2 استانداردسازی متغیرهای غیر عددی	24
3-3 بررسی و مدیریت داده‌های گمشده	25
3-4 شناسایی مقادیر نامعتبر و مقادیر شدید	27

28	3-5 مهندسی ویژگی‌ها.....
29	3-6 آماده‌سازی هدف مدل.....
29	3-7 تشخیص ویژگی‌های نامناسب برای مدل.....
30	3-8 بررسی رکوردهای تکراری.....
30	3-9 مجموعه ویژگی‌های آماده برای مدل.....
31	3-10 خروجی نهایی مرحله پاک‌سازی.....
33	فصل چهارم.....
33	آماده‌سازی نهایی داده و طراحی فرآیند مدل‌سازی.....
33	4-1-مقدمه.....
33	4-2 بررسی ساختار زمانی داده.....
34	4-3 طراحی Train ، Validation و Test.....
34	4-4 جلوگیری از نشت اطلاعات.....
35	4-5 ساخت ویژگی زمانی.....
35	4-6 طراحی preprocessing.....
36	4-7 One-Hot Encoding و کنترل category های جدید.....
36	4-8 ابعاد داده پس از preprocessing.....
37	4-9 بررسی cardinality و ویژگی‌های categorical.....
37	4-10 بررسی Outlier ها.....
38	فصل پنجم.....
38	ساخت مدل‌های پایه و مدل نهایی.....
38	5-1 مقدمه.....
38	5-2 Baseline با پیش‌بینی میانه.....
39	5-3 Ridge Regression.....
39	5-4 ورود CatBoost.....
40	5-5 افزایش تعداد iteration.....
40	5-6 اهمیت ویژگی‌ها.....
41	5-7 بررسی user_type و city_listing_frequency.....
41	5-8 Ablation Study.....
41	5-9 حذف ویژگی ثابت cat2_slug.....

42	5-10 مقایسه عمق‌های مختلف CatBoost
43	5-11 آموزش مدل نهایی روی Train + Validation
44	فصل ششم
44	ارزیابی نهایی مدل و تحلیل خطا
44	6-1 نتیجه نهایی روی Test
44	6-2 پاک‌سازی نهایی قیمت‌های تکراری
45	6-3 مقایسه Test و Validation
45	6-4 تفاوت عملکرد میان انواع ملک
46	6-5 بررسی خطاهای بزرگ
47	6-6 بررسی عملکرد در سطوح مختلف قیمت
48	6-7 محدودیت مهم پروژه
48	6-8 جمع‌بندی مدل نهایی
50	فصل هفتم
50	جمع‌بندی کل پروژه
50	7-1 مسیر طی‌شده
50	7-2 مهم‌ترین یافته‌های پروژه
51	7-3 نتیجه نهایی پروژه

فصل اول

شناخت مسئله، تعریف هدف و بررسی اولیه داده‌ها

1-1 مقدمه

بازار مسکن یکی از حوزه‌هایی است که در آن تصمیم‌گیری تا حد زیادی به اطلاعات وابسته است. قیمت یک ملک معمولاً حاصل ترکیبی از عوامل مختلف است؛ از ویژگی‌های فیزیکی مانند مترáž، تعداد اتاق و سن بنا گرفته تا موقعیت جغرافیایی، نوع ملک، شرایط عرضه و وضعیت زمانی بازار. همین موضوع باعث می‌شود که تحلیل داده‌های مربوط به آگهی‌های ملکی، علاوه بر ارزش تحلیلی، زمینه مناسبی برای استفاده از روش‌های یادگیری ماشین فراهم کند.

در این پروژه، هدف اصلی تحلیل داده‌های آگهی‌های املاک منتشرشده در پلتفرم دیوار و بررسی امکان استفاده از این اطلاعات برای مدل‌سازی قیمت آگهی‌های ملکی بود. داده مورد استفاده مجموعه‌ای با یک میلیون آگهی و 60 ویژگی اولیه بود که اطلاعات متنوعی از انواع معاملات و خدمات مرتبط با املاک را در خود جای داده است. به دلیل حجم و تنوع داده، از همان ابتدای کار مشخص بود که پیش از رسیدن به مرحله مدل‌سازی، شناخت دقیق ساختار داده و تعیین دامنه مناسب مسئله اهمیت زیادی خواهد داشت. این موضوع در ادامه پروژه نیز یکی از عوامل اصلی تصمیم‌گیری‌های تحلیلی بود.

در شروع پروژه انتظار اولیه این بود که بتوان با استفاده از متغیرهایی مانند مترáž، تعداد اتاق، سال ساخت، شهر و محله، یک مدل رگرسیونی برای پیش‌بینی قیمت ایجاد کرد. با این حال، این انتظار اولیه به‌مرور و با بررسی واقعی داده‌ها اصلاح شد. مهم‌ترین دلیل این تغییر، ناهمگونی موجود در dataset بود؛ داده‌ها صرفاً شامل فروش آپارتمان‌های مسکونی نبودند و دسته‌های مختلفی مانند اجاره، فروش تجاری، اجاره تجاری، اجاره موقت و خدمات مشاوران املاک نیز در همان جدول قرار داشتند.

بنابراین نخستین مسئله‌ای که در پروژه با آن مواجه شدیم، انتخاب مدل نبود؛ بلکه تعریف درست مسئله بود.

1-2 تعریف مسئله

در ابتدای پروژه مسئله به شکل کلی «تحلیل بازار مسکن و پیش‌بینی قیمت ملک» تعریف شد. اما بررسی ساختار داده نشان داد که نمی‌توان تمام رکوردها را با یک تعریف واحد از قیمت تحلیل کرد.

برای مثال، در یک رکورد ممکن است قیمت فروش یک ملک ثبت شده باشد، در حالی که در رکورد دیگری اطلاعات مربوط به اجاره یا ودیعه وجود داشته باشد. در بخش دیگری نیز ممکن است رکورد مربوط به خدمات املاک باشد و اساساً قیمت ملک هدف اصلی آن آگهی نباشد.

در نتیجه، تصمیم گرفته شد که dataset به عنوان یک جدول کاملاً همگن در نظر گرفته نشود. در عوض، دسته‌های کسب‌وکاری مختلف حفظ شوند و مسئله مدل‌سازی قیمت به زیرمجموعه‌ای محدود شود که تعریف مشخص‌تری از target دارد.

در نهایت، مسیر اصلی مدل‌سازی روی آگهی‌های فروش املاک مسکونی قرار گرفت که با مقدار residential-sell ویژگی مربوط به دسته‌بندی مشخص می‌شوند. این تصمیم یکی از مهم‌ترین تصمیم‌های پروژه بود، زیرا باعث شد تعریف target با ماهیت رکوردها سازگار باشد و از ترکیب چند مسئله متفاوت در یک مدل جلوگیری شود.

این رویکرد در مستندات اولیه پروژه نیز به عنوان یک تصمیم طراحی اصلی ثبت شده است: کل dataset به عنوان یک مسئله مدل‌سازی همگن در نظر گرفته نشود و تمرکز مدل قیمت بر بخش فروش مسکونی قرار گیرد.

3-1 راه‌اندازی محیط پروژه

پس از مشخص شدن هدف کلی، ساختار پروژه ایجاد شد تا تحلیل از همان ابتدا به صورت مرحله‌ای و قابل بازتولید انجام شود.

محیط توسعه بر پایه Python 3.11 و یک virtual environment مستقل ایجاد شد. همچنین ساختار پوشه‌ها به گونه‌ای طراحی شد که داده خام، مستندات، Notebookها، گزارش‌ها و فایل‌های مدل از یکدیگر جدا باشند.

این موضوع صرفاً یک تصمیم سازمان‌دهی فایل‌ها نبود. از آنجا که dataset حجیم بود و قرار بود چندین مرحله تحلیل و مدل‌سازی روی آن انجام شود، تفکیک داده خام از خروجی‌های پردازش‌شده اهمیت زیادی پیدا کرد. همچنین تصمیم گرفته شد مستندات پروژه در طول انجام کار به روزرسانی شوند و تصمیم‌های مهم در فایل Project Log ثبت شوند.

به این ترتیب پروژه از ابتدا با یک ساختار نسبتاً منظم آغاز شد و تحلیل‌ها در Notebookهای مجزا پیش رفتند.

4-1 بارگذاری و بررسی اولیه dataset

در نخستین بررسی عملی، dataset با موفقیت بارگذاری شد.

نتایج اولیه نشان داد که داده شامل:

1,000,000 رکورد

60 ستون

حدود 1.47 گیگابایت مصرف حافظه در حالت بارگذاری شده

هیچ رکورد کاملاً تکراری

است.

نبودن duplicate row در این مرحله نتیجه مثبتی به نظر می‌رسید، اما هنوز نمی‌توانستیم نتیجه بگیریم که داده از نظر کیفیت کاملاً مناسب است. «تکراری نبودن رکوردها» با «درست بودن اطلاعات موجود در رکوردها» تفاوت دارد و همین تفاوت در مراحل بعدی پروژه اهمیت زیادی پیدا کرد.

در همین مرحله مشخص شد که انواع مختلف داده‌های متنی، عددی و ترکیبی در dataset وجود دارند. بنابراین از همان ابتدا مشخص بود که بخش قابل توجهی از کار مربوط به استانداردسازی و تفسیر صحیح ویژگی‌ها خواهد بود.

1-5 پرو فایل اولیه ویژگی‌ها

در مرحله بعد، برای هر ستون اطلاعاتی مانند نوع داده، تعداد مقادیر یکتا و میزان missingness بررسی شد.

هدف از این مرحله پیدا کردن feature های «خوب» و «بد» نبود. مسئله مهم‌تر این بود که بفهمیم هر ستون در واقع چه نقشی در ساختار کسب‌وکاری داده دارد.

در این بررسی مشخص شد که تعداد قابل توجهی از ستون‌ها دارای missing value هستند و برخی ستون‌ها درصد بسیار بالایی از مقادیر خالی دارند. در نگاه اول ممکن بود حذف چنین ستون‌هایی یک راه حل ساده به نظر برسد، اما بررسی دقیق‌تر نشان داد که missingness در بسیاری از موارد به نوع آگهی وابسته است.

برای مثال، خالی بودن برخی اطلاعات در یک آگهی زمین کاملاً طبیعی است، زیرا ویژگی‌هایی مانند تعداد اتاق یا سال ساخت ممکن است اساساً برای آن نوع ملک معنا نداشته باشند.

بنابراین یک تصمیم مهم در همان مراحل ابتدایی شکل گرفت:

وجود missing value به‌تنهایی دلیل کافی برای حذف یک ویژگی یا رکورد نیست.

این تصمیم بعدها نقش مهمی در طراحی مرحله Data Cleaning و Feature Engineering پیدا کرد.

1-6 شناخت ساختار کسب و کاری داده

یکی از مهم‌ترین نتایج Phase 1 این بود که dataset بیشتر از آنکه یک جدول ساده قیمت ملک باشد، نماینده‌ای از چند فرآیند متفاوت در پلتفرم آگهی‌های املاک است.

داده‌ها شامل دسته‌هایی مانند فروش مسکونی، اجاره مسکونی، فروش و اجاره تجاری، اجاره موقت و خدمات املاک بودند.

این مسئله یک پیامد مستقیم برای مدل‌سازی داشت. اگر همه این رکوردها در یک مدل قیمت قرار می‌گرفتند، target معنای یکسانی برای همه رکوردها نداشت.

در نتیجه، به جای اینکه مسئله را با الگوریتم پیچیده‌تر حل کنیم، ابتدا مسئله را دقیق‌تر تعریف کردیم. این یکی از مهم‌ترین تجربه‌های پروژه بود: کیفیت تعریف مسئله می‌تواند از انتخاب الگوریتم مهم‌تر باشد.

1-7 تعریف اولیه target

پس از تعیین دامنه مدل‌سازی، price_value به عنوان target اصلی انتخاب شد.

با این حال، در همان مراحل اولیه مشخص شد که توزیع قیمت خام به شدت نامتقارن است. میانگین قیمت فاصله زیادی با میانه داشت و وجود تعداد محدودی قیمت بسیار بزرگ باعث ایجاد دنباله سنگینی در توزیع می‌شد.

در این مرحله هنوز تصمیم نهایی درباره مدل گرفته نشده بود، اما مشخص بود که استفاده مستقیم از قیمت خام احتمالاً برای یک مسئله رگرسیون مناسب نخواهد بود.

بنابراین تصمیم گرفته شد در مراحل بعدی target به مقیاس لگاریتمی منتقل شود. در نهایت این تبدیل با استفاده از $\log_{1p}(\text{price_value})$ انجام شد و خروجی آن با نام log_price در فرآیند مدل‌سازی مورد استفاده قرار گرفت. مستندات پروژه نیز از ابتدا این تبدیل را به عنوان target مورد انتظار برای رگرسیون ثبت کرده بودند.

1-8 انتظار اولیه در برابر واقعیت داده

در ابتدای پروژه انتظار می‌رفت ویژگی‌هایی مانند متر اژ، تعداد اتاق، سال ساخت و موقعیت جغرافیایی رابطه نسبتاً مستقیمی با قیمت داشته باشند و پس از یک مرحله پاک‌سازی معمول بتوان مدل پیش‌بینی قیمت را آموزش داد. اما همین مرحله ابتدایی نشان داد که مسئله پیچیده‌تر است.

چند واقعیت از همان ابتدا مشخص شد:

نخست اینکه داده از چند فرآیند کسب‌وکاری مختلف تشکیل شده است.

دوم اینکه missingness در بسیاری از ویژگی‌ها تصادفی نیست و با نوع آگهی ارتباط دارد.

سوم اینکه قیمت دارای توزیع بسیار نامتقارن و مقادیر بسیار بزرگ است.

چهارم اینکه ویژگی‌های عددی و متنی در کنار یکدیگر قرار دارند و بسیاری از اطلاعات مهم مانند شهر و محله ماهیت categorical دارند.

و در نهایت، مشخص شد که «تمیز کردن داده» قرار نیست صرفاً شامل حذف null ها و outlier ها باشد؛ بلکه باید معنای کسب‌وکاری هر نوع داده نیز در نظر گرفته شود.

این تغییر نگاه، مسیر مراحل بعدی پروژه را مشخص کرد.

9-1 خروجی فصل اول

در پایان Phase 1، مسئله از یک هدف کلی یعنی «تحلیل بازار مسکن» به یک مسئله مشخص‌تر تبدیل شد:

تحلیل و مدل‌سازی قیمت آگهی‌های فروش املاک مسکونی در dataset آگهی‌های دیوار، با در نظر گرفتن ویژگی‌های فیزیکی، جغرافیایی و زمانی آگهی‌ها.

همچنین تصمیم‌های زیر به‌عنوان مبنای ادامه پروژه تثبیت شدند:

1. کل dataset نباید به‌عنوان یک مسئله همگن مدل‌سازی شود.
2. مسیر اصلی مدل‌سازی روی residential-sell متمرکز شود.
3. price_value به‌عنوان target اصلی انتخاب شود.
4. به دلیل skew شدید قیمت، از تبدیل لگاریتمی target استفاده شود.
5. missing value ها بدون بررسی معنای کسب‌وکاری حذف نشوند.
6. داده خام دست‌نخورده باقی بماند و preprocessing در مراحل بعدی انجام شود.
7. ساختار پروژه و مستندسازی همزمان با تحلیل پیش برود.

به این ترتیب، فصل اول بیشتر از آنکه به ساخت مدل مربوط باشد، به این پرسش پاسخ داد که «دقیقاً چه چیزی را قرار است مدل کنیم و با چه داده‌ای طرف هستیم؟»

پاسخ به همین پرسش بود که زمینه را برای ورود به مرحله Exploratory Data Analysis فراهم کرد.

فصل دوم

تحلیل اکتشافی داده‌ها

2-1 مقدمه

پس از آنکه در مرحله نخست ساختار کلی dataset و دامنه مسئله مشخص شد، وارد مرحله‌ای شدیم که هدف آن نه ساخت مدل، بلکه شناخت رفتار واقعی داده‌ها بود. در این مرحله تلاش شد قبل از هرگونه پاک‌سازی یا تغییر در داده، مشخص شود که متغیرها چگونه توزیع شده‌اند، چه ارتباطی با یکدیگر دارند، missing valueها چه الگویی دارند و آیا فرض‌های اولیه درباره بازار مسکن با آنچه واقعاً در داده دیده می‌شود سازگار هستند یا خیر.

این مرحله در عمل یکی از تعیین‌کننده‌ترین بخش‌های پروژه بود. بسیاری از تصمیم‌هایی که در مراحل بعدی درباره outlier، target، ویژگی‌های ملکی و حتی ساختار مدل گرفته شد، مستقیماً از نتایج EDA به دست آمد.

2-2 بررسی اولیه کیفیت داده

هدف

نخستین هدف EDA این بود که بدون اعمال هیچ preprocessing، وضعیت عمومی dataset را بررسی کنیم.

در این مرحله سه سؤال اصلی مطرح بود:

آیا رکوردهای تکراری وجود دارند؟

آیا ستون‌های تکراری یا بدون اطلاعات وجود دارند؟

نوع داده‌های موجود چیست و missing valueها در چه وضعیتی قرار دارند؟

نتایج

نتیجه بررسی duplicate rowها برابر صفر بود. همچنین ستون ثابت یا constant feature نیز شناسایی نشد.

این نتیجه از یک طرف امیدوارکننده بود، زیرا نشان می‌داد dataset از نظر وجود رکوردهای کاملاً تکراری مشکل آشکاری ندارد. اما از طرف دیگر، در همان مرحله مشخص شد که تعداد زیادی از ستون‌ها missing value دارند.

در اینجا یک نکته روش‌شناختی مهم مطرح شد. وجود مقدار خالی به خودی خود به معنای خراب بودن رکورد نیست. به‌خصوص در dataset ای که چند نوع آگهی را کنار هم قرار داده، ممکن است یک feature برای یک نوع آگهی کاملاً مرتبط باشد ولی برای نوع دیگری اصلاً کاربرد نداشته باشد.

بنابراین تصمیم گرفتیم در این مرحله هیچ preprocessing انجام ندهیم و dataset را همان‌طور که دریافت شده بود حفظ کنیم.

این تصمیم باعث شد تحلیل‌های بعدی بر مبنای داده خام انجام شوند و قبل از اینکه تغییری در داده ایجاد کنیم، تصویر واقعی dataset را ببینیم.

2-3 بررسی Missing Value ها

هدف

در ادامه تمرکز اصلی روی missing value ها قرار گرفت.

انتظار اولیه این بود که بتوان با بررسی درصد missingness، تعدادی از ستون‌های کم‌اطلاعات را حذف کرد و سپس preprocessing را ساده‌تر کرد. اما نتایج خیلی زود نشان دادند که چنین تصمیمی می‌تواند باعث از دست رفتن بخش مهمی از اطلاعات شود.

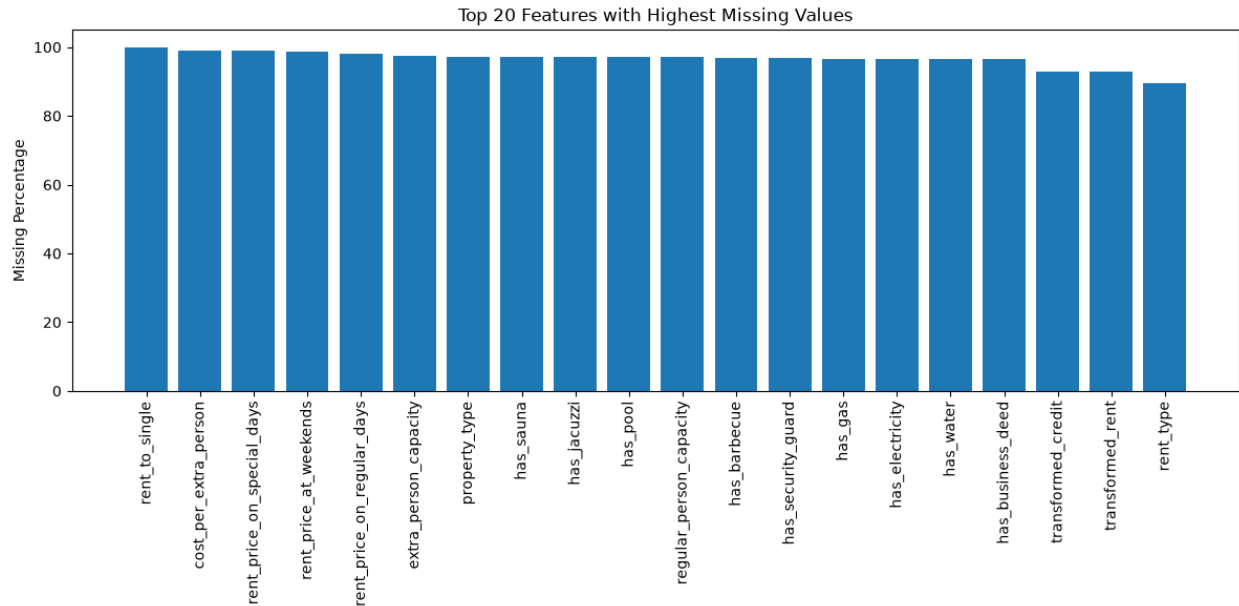
یافته اصلی

بیش از دو سوم ویژگی‌ها در dataset دارای بیش از 50 درصد missing value بودند.

اگر این موضوع صرفاً از دید آماری بررسی می‌شد، حذف تعداد زیادی از این ستون‌ها تصمیمی منطقی به نظر می‌رسید. اما زمانی که missingness را در ارتباط با دست‌بندی آگهی‌ها بررسی کردیم، مشخص شد که این الگو تا حد زیادی ناشی از ساختار کسب‌وکاری dataset است.

برای مثال، بعضی ویژگی‌های مربوط به اجاره یا ودیعه برای آگهی فروش معنای مستقیمی ندارند. به همین ترتیب، ویژگی‌هایی که برای یک ملک مسکونی ساختمانی کاربرد دارند، ممکن است برای یک زمین فاقد معنا باشند.

بنابراین missingness در این dataset را نمی‌توان به‌طور ساده معادل «اطلاعات ناقص تصادفی» در نظر گرفت.



واکنش تحلیلی

در این مرحله تصمیم گرفتیم درصد missingness را به‌تنهایی معیار حذف feature قرار ندهیم.

به عبارت دیگر، به جای اینکه بگوییم:

«این ستون 80 درصد خالی است، پس باید حذف شود»

سؤال دقیق‌تر را مطرح کردیم:

«چرا این ستون خالی است و آیا خالی بودن آن خودش بخشی از ساختار داده است؟»

این تغییر دیدگاه در ادامه پروژه بسیار مهم شد.

برای مثال، در مرحله Feature Engineering مشخص شد که missing بودن تعداد اتاق و سال ساخت تقریباً با یکدیگر

هم‌راستا است و بخش عمده‌ای از این missingness به زمین‌ها مربوط می‌شود. بنابراین در نهایت به جای حذف این

اطلاعات، indicatorهایی برای مشخص کردن وجود یا عدم وجود این اطلاعات ایجاد کردیم.

این تصمیم در مستندات Phase 2 نیز ثبت شد که missing valueها نباید صرفاً بر اساس درصدشان حذف شوند و باید

سازوکار ایجاد آنها در نظر گرفته شود.

2-4 بررسی توزیع قیمت فروش مسکونی

2-4-1 تمرکز بر بخش فروش مسکونی

پس از بررسی کلی dataset، تمرکز تحلیل به زیرمجموعه فروش مسکونی منتقل شد.

در این مرحله مشخص شد که حدود 535 هزار رکورد در محدوده اصلی فروش مسکونی قرار می‌گیرند. البته عدد دقیق model-ready بعداً، پس از حذف target های نامعتبر، به 533,261 رکورد رسید.

هدف این مرحله بررسی رفتار قیمت و پاسخ به یک سؤال اساسی بود:

آیا می‌توان قیمت را به صورت مستقیم به عنوان target یک مدل رگرسیون استفاده کرد؟

2-4-2 توزیع قیمت خام

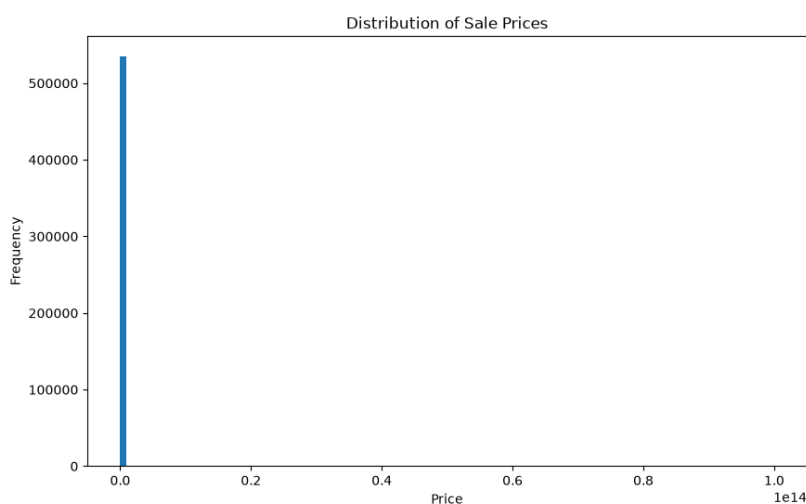
نتایج بسیار واضح بودند.

قیمت خام توزیعی به‌شدت right-skewed داشت و مقدار skewness تقریباً برابر 112.7 بود.

این مقدار برای ما یک هشدار جدی بود. چنین چولگی شدیدی نشان می‌داد که بخش بزرگی از داده‌ها در محدوده قیمت‌های پایین‌تر قرار دارند، در حالی که تعداد محدودی از رکوردها قیمت‌های بسیار بزرگی دارند.

در نتیجه میانگین قیمت فاصله قابل توجهی با میانه داشت.

این رفتار را می‌توان با مقایسه نمودار histogram و boxplot نیز مشاهده کرد. در گزارش تصویری پروژه، این قسمت از تحلیل با نمودار توزیع قیمت خام و نمودار جعبه‌ای قابل نمایش است.





2-4-3 بررسی مقادیر بسیار بزرگ

در ادامه، مقادیر extreme price بررسی شدند.

در اینجا یک تصمیم مهم گرفته شد: هر مقدار بزرگی الزاماً خطا نیست.

در بازار مسکن، وجود املاک بسیار گران قیمت می تواند واقعی باشد. بنابراین صرفاً بر اساس فاصله یک مقدار از Q1 و Q3 نمی توان گفت که آن مقدار اشتباه است.

این موضوع بعدها در مورد building_size نیز تکرار شد. برخی متراژهای بسیار بزرگ در واقع مربوط به زمین بودند و نمی توانستیم آنها را صرفاً به دلیل بزرگ بودن حذف کنیم.

بنابراین از همان ابتدا اصل زیر در پروژه شکل گرفت:

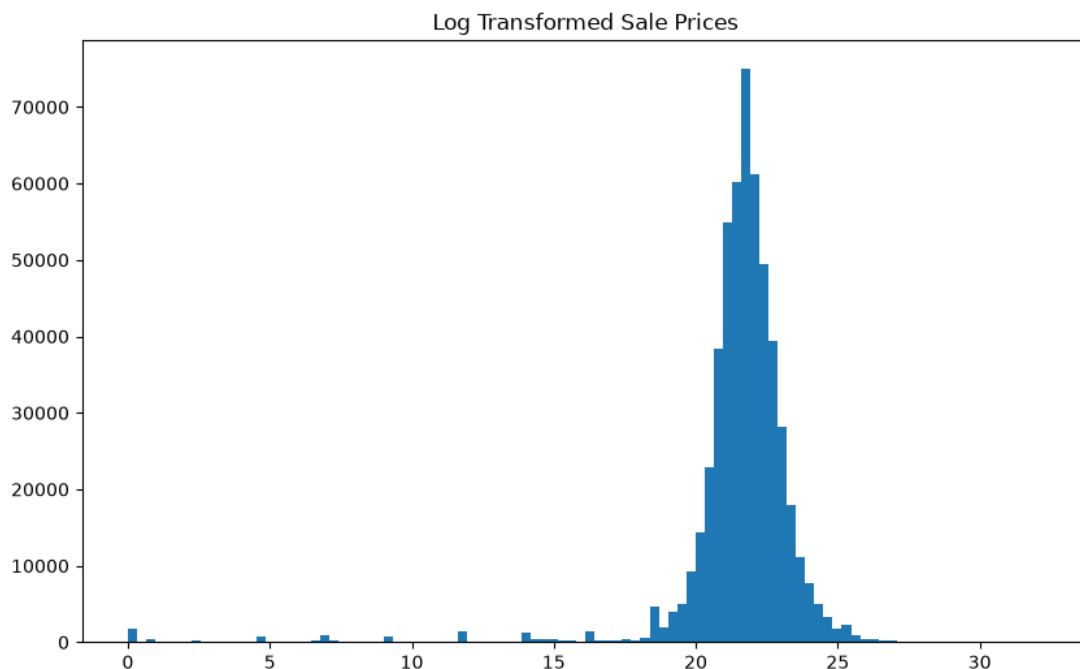
تشخیص outlier باید با توجه به معنای کسب و کاری متغیر انجام شود، نه صرفاً بر اساس یک آستانه آماری.

2-4-4 تصمیم درباره تبدیل لگاریتمی target

با توجه به چولگی بسیار شدید قیمت، تبدیل لگاریتمی به عنوان راه حل مناسب بررسی شد.

هدف این نبود که extreme value ها را حذف کنیم؛ بلکه می خواستیم اثر فاصله بسیار زیاد بین قیمت های مختلف را در فرآیند یادگیری کنترل کنیم.

در نهایت تصمیم گرفته شد که target مدل سازی به شکل $\log_{10}(\text{price_value})$ تعریف شود.



این تصمیم بعداً با محاسبات دقیق نیز تأیید شد target. لگاریتمی نهایی با استفاده از همین تبدیل ساخته شد و در تمام مدل‌های اصلی مورد استفاده قرار گرفت.

این تصمیم از نظر مدل‌سازی اهمیت زیادی داشت، زیرا باعث شد مسئله به جای پیش‌بینی مستقیم قیمت‌های بسیار skewed، به پیش‌بینی لگاریتم قیمت تبدیل شود.

2-5 بررسی ویژگی‌های ساختاری و عوامل مؤثر بر قیمت

پس از بررسی target، تمرکز روی ویژگی‌هایی قرار گرفت که انتظار می‌رفت بیشترین ارتباط را با قیمت داشته باشند.

در این مرحله چهار گروه اصلی بررسی شدند:

نوع ملک

موقعیت جغرافیایی

متراژ

اطلاعات مربوط به اتاق و سال ساخت

1-5-2 نوع ملک

یکی از مهم‌ترین نتایج EDA، مشخص شدن ساختار سه‌گانه بخش فروش مسکونی بود.

سه دسته اصلی عبارت بودند از:

آگهی apartment-sell: 303,380

آگهی house-villa-sell: 121,748

آگهی plot-old: 109,836

این اعداد نشان دادند که حتی درون بخش فروش مسکونی نیز با سه نوع متفاوت از دارایی روبه‌رو هستیم.

در ابتدا ممکن بود انتظار داشته باشیم تمام این رکوردها با یک مدل ساده قابل پیش‌بینی باشند. اما همین تحلیل نشان داد که زمین از نظر ساختاری با آپارتمان و خانه تفاوت اساسی دارد.

برای مثال، نبود تعداد اتاق برای زمین یک missing value تصادفی نیست؛ بلکه کاملاً قابل انتظار است.

به همین دلیل تصمیم گرفتیم ویژگی is_land را در مراحل بعدی ایجاد کنیم تا مدل بتواند این تفاوت ساختاری را درک کند.

2-5-2 شهر و محله

تحلیل city-level نشان داد که حجم آگهی‌ها به‌شدت در شهرهای مختلف متفاوت است.

در میان شهرهای پرتکرار، تهران، مشهد، کرج، اصفهان، شیراز و تبریز سهم قابل توجهی از داده را داشتند.

از طرف دیگر، قیمت ملک نیز به موقعیت جغرافیایی وابستگی داشت. بنابراین از همان مرحله EDA مشخص شد که حذف city و neighborhood باعث از دست رفتن بخشی از اطلاعات مهم بازار خواهد شد.

این نتیجه بعدها در مدل CatBoost نیز خود را نشان داد؛ در تحلیل feature importance، city_slug، مهم‌ترین feature مدل نهایی ablation بود.

بنابراین تصمیم EDA برای حفظ ویژگی‌های جغرافیایی، در مرحله مدل‌سازی نیز تأیید شد.

2-5-3 بررسی building size

متراژ ساختمان یکی از متغیرهایی بود که از ابتدا انتظار داشتیم رابطه مشخصی با قیمت داشته باشد.

نتایج توصیفی نشان دادند که توزیع building_size نیز به شدت کشیده است.

مقادیر میانه و صدک‌های اصلی تقریباً به این شکل بودند:

میانه: 115

صدک 75: حدود 200

صدک 90: حدود 400

صدک 95: حدود 1,000

صدک 99: حدود 7,500

صدک 99.5: حدود 23,000

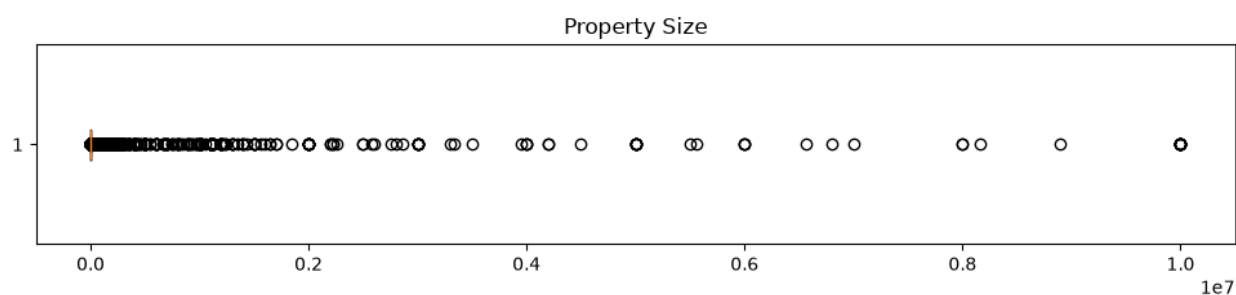
صدک 99.9: حدود 700,000

بیشینه: 10,000,000

در نگاه اول، این اعداد می‌توانستند نشان دهنده وجود تعداد زیادی خطای ثبت باشند. اما بررسی دسته بندی ملک نشان داد که بخش قابل توجهی از مقادیر بسیار بزرگ مربوط به زمین‌ها هستند.

بنابراین اگر صرفاً بر اساس صدک 99 یا IQR رکوردها حذف می‌شدند، احتمال حذف بخشی از اطلاعات واقعی زمین‌ها وجود داشت.

این یکی از نقاطی بود که پروژه از یک پاکسازی آماری ساده فاصله گرفت و به سمت پاکسازی مبتنی بر business context رفت.



2-5-4 رابطه متراژ و قیمت

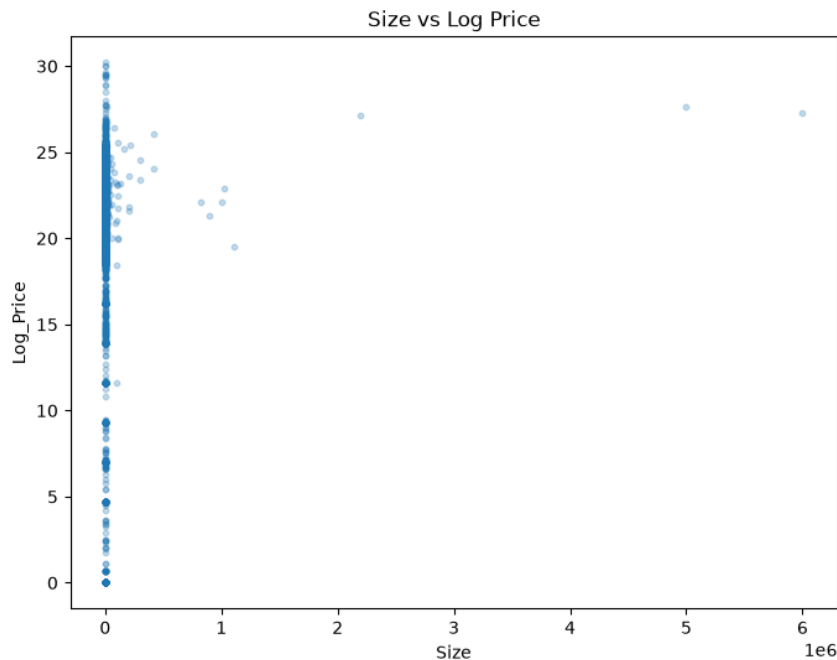
بررسی همبستگی نشان داد که ارتباط بین building size و log price مثبت است، اما به اندازه‌ای قوی نیست که بتوان قیمت را صرفاً بر اساس متراژ توضیح داد.

Spearman correlation تقریباً برابر 0.205 بود.

این نتیجه با انتظار اولیه ما تفاوت داشت.

انتظار داشتیم متراژ یکی از قوی‌ترین متغیرهای توضیح دهنده قیمت باشد. اما مقدار همبستگی نشان داد که رابطه قیمت و متراژ به تنهایی کافی نیست و عواملی مانند شهر، محله، نوع ملک و سایر ویژگی‌ها نقش مهمی دارند.

در واقع همین نتیجه یکی از دلایلی بود که در ادامه به سمت مدل‌های غیرخطی و دارای قابلیت کار با categorical feature‌ها رفتیم.



2-6 بررسی اطلاعات تعداد اتاق

ویژگی rooms_count در dataset به شکل عددی استاندارد نبود.

مقادیر شامل دسته‌هایی به زبان فارسی بودند؛ از جمله «یک»، «دو»، «سه»، «چهار»، «پنج یا بیشتر» و «بدون اتاق».

در EDA مشخص شد که این feature در کنار اطلاعات مربوط به construction year رفتار missing مشابهی دارد.

تقریباً 20.6 درصد از آگهی‌های فروش فاقد اطلاعات اتاق بودند.

اما بررسی دقیق‌تر نشان داد که این missingness ارتباط زیادی با زمین‌ها دارد.

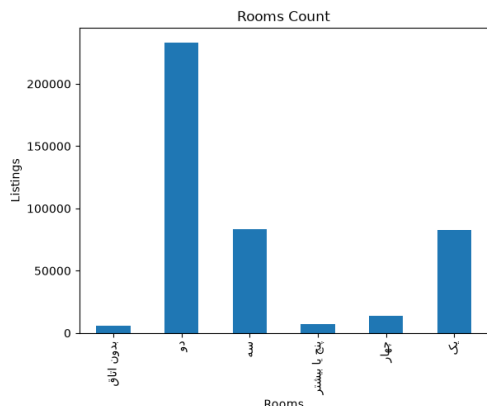
بنابر این نتیجه این نبود که:

20 درصد داده ناقص است و باید impute شود.

بلکه نتیجه این بود که:

«نبود اطلاعات اتاق احتمالاً بخشی از ساختار نوع ملک است»

همین موضوع در Phase 3 باعث شد علاوه بر تبدیل room count به مقدار عددی، یک feature برای نشان دادن وجود یا عدم وجود این اطلاعات ایجاد کنیم.



2-7 بررسی construction year

سال ساخت نیز وضعیت مشابهی داشت.

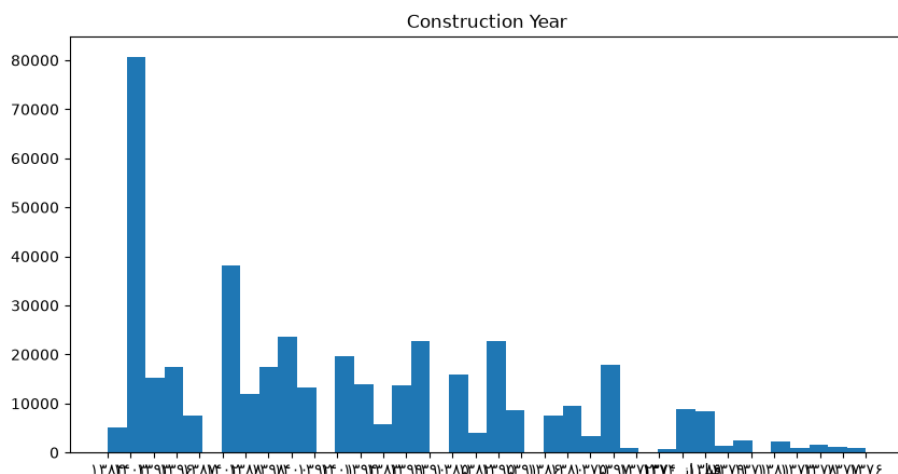
این متغیر در همه رکوردها به صورت عدد ساده ذخیره نشده بود و مقادیری مانند «قبل از ۱۳۷۰» در آن دیده می‌شد.

از طرف دیگر، missingness آن با missing بودن room information هم راستا بود.

در نتیجه تصمیم گرفته شد که این feature نیازمند یک تبدیل معنایی باشد، نه صرفاً تبدیل type.

مقادیر سال ساخت باید ابتدا نرمال شوند، سپس به شکل عددی تبدیل شوند و در نهایت بتوان از آنها ویژگی جدیدی مانند سن ملک استخراج کرد.

این تصمیم در مرحله بعد به ساخت construction_year_clean و property_age منجر شد.



2-8 نتیجه کلی EDA

در پایان Phase 2 ، تصویر بسیار دقیق‌تری از dataset نسبت به شروع پروژه داشتیم.

چند نتیجه اصلی مسیر ادامه پروژه را مشخص کردند:

اول، dataset یک مجموعه کاملاً همگن نبود و انواع مختلف آگهی‌ها را در خود جای داده بود.

دوم، missingness در بسیاری از ویژگی‌ها وابسته به نوع آگهی بود و نمی‌توانستیم آن را صرفاً noise تلقی کنیم.

سوم، قیمت خام به شدت چوله بود و استفاده از log transformation منطقی به نظر می‌رسید.

چهارم، متراژ با قیمت رابطه مثبت داشت، اما همبستگی حدود 0.205 نشان داد که قیمت تابع یک متغیر واحد نیست.

پنجم، city و neighborhood از نظر تحلیلی ارزشمند بودند و نباید به دلیل categorical بودن کنار گذاشته می‌شدند.

ششم، land listings از نظر ساختاری با ساختمان‌های مسکونی تفاوت داشتند و باید به‌صورت صریح شناسایی می‌شدند.

هفتم، outlierها را نمی‌توان صرفاً با یک قانون آماری حذف کرد، زیرا بخشی از مقادیر extreme ممکن است واقعاً مربوط به املاک بزرگ یا زمین باشند.

در نتیجه، پایان EDA به جای یک dataset تمیز، چیزی مهم‌تر در اختیارمان گذاشت: یک درک نسبتاً روشن از اینکه چه چیزهایی در داده واقعاً مسئله هستند و چه چیزهایی فقط در نگاه اول غیرعادی به نظر می‌رسند.

همین موضوع مبنای Phase 3 شد؛ یعنی مرحله‌ای که قرار بود داده را برای مدل‌سازی آماده کنیم، بدون اینکه اطلاعات معنادار بازار را به دلیل قواعد ساده آماری از بین ببریم .

2-9 جمع‌بندی تصمیم‌های Phase 2

در پایان این مرحله، تصمیم‌های زیر تثبیت شدند:

1. مدل اصلی روی فروش مسکونی متمرکز شود .
 2. Target با استفاده از $\log1p(\text{price_value})$ تبدیل شود .
 3. zero price به‌عنوان target نامعتبر در نظر گرفته شود .
 4. Outlier های عددی بدون بررسی معنای کسب‌وکاری حذف نشوند .
 5. land listings به‌صورت صریح شناسایی شوند .
 6. City و neighborhood حفظ شوند .
 7. room count به شکل عددی استاندارد شود .
 8. construction year نیازمند normalization و تبدیل معنایی باشد .
 9. Missingness به‌عنوان بخشی از اطلاعات dataset در نظر گرفته شود .
 10. پیش از ورود به مدل‌سازی، feature engineering و cleaning دقیق انجام شود .
- در این نقطه، انتظار اولیه پروژه برای یک مدل نسبتاً ساده و مستقیم تا حدی تغییر کرده بود. حالا مشخص بود که موفقیت مدل بیش از آنکه به انتخاب یک الگوریتم پیچیده وابسته باشد، به نحوه آماده‌سازی و تفسیر داده وابسته خواهد بود.
- به همین دلیل Phase 3 با تمرکز بر Data Cleaning و Feature Engineering آغاز شد.

فصل سوم

پاکسازی داده‌ها و مهندسی ویژگی

3-1 مقدمه

پس از پایان تحلیل اکتشافی، مشخص شد که مسئله اصلی فقط انتخاب یک الگوریتم یادگیری ماشین نیست. داده خام Divar پیش از آنکه بتواند وارد مرحله مدل‌سازی شود، به چند اصلاح اساسی نیاز داشت. بخشی از متغیرها به شکل متنی ذخیره شده بودند، بعضی مقادیر از نظر معنایی نامعتبر بودند، میزان داده‌های گم‌شده در برخی ویژگی‌ها قابل توجه بود و بعضی از متغیرها نیز در نگاه اول مناسب به نظر می‌رسیدند اما می‌توانستند باعث نشت اطلاعات از هدف به مدل شوند.

در این مرحله تلاش کردم پاکسازی را به شکلی انجام دهم که تا حد امکان اطلاعات واقعی آگهی‌ها حفظ شود. هدف حذف هر چیزی که «غیر عادی» به نظر می‌رسید نبود؛ بلکه باید میان داده نامعتبر، داده گم‌شده و داده‌ای که صرفاً به دلیل ساختار بازار مقدار بزرگی دارد تفاوت ایجاد می‌شد.

این موضوع به‌خصوص در داده‌های ملکی اهمیت داشت. برای مثال، یک `building_size` بسیار بزرگ لزوماً به معنی خطای ثبت داده نیست؛ ممکن است آگهی مربوط به یک زمین یا ملک با کاربری متفاوت باشد. بنابراین هر حذف یا تبدیل باید بر اساس منطق داده انجام می‌شد، نه صرفاً بر اساس یک آستانه آماری.

3-2 استانداردسازی متغیرهای غیر عددی

یکی از اولین مشکلاتی که در داده‌ها مشاهده شد، وجود متغیرهایی بود که از نظر مفهومی عددی بودند اما در فایل CSV به شکل متن ذخیره شده بودند.

نمونه مشخص آن `rooms_count` بود. مقادیر این ستون شامل عبارت‌هایی مانند «بدون اتاق»، «یک»، «دو»، «سه»، «چهار» و «پنج یا بیشتر» بود. این قالب برای نمایش انسانی مناسب است، اما برای بسیاری از الگوریتم‌های یادگیری ماشین به یک نمایش عددی نیاز دارد.

بنابراین یک نگاشت مشخص برای این مقادیر ایجاد شد:

«بدون اتاق» به 0

«یک» به 1

«دو» به 2

«سه» به 3

«چهار» به 4

«پنج یا بیشتر» به 5

در اینجا انتخاب مقدار 5 برای «پنج یا بیشتر» به معنی ادعای اینکه تمام این املاک دقیقاً پنج اتاق دارند نبود. این مقدار صرفاً یک نمایش عددی محدود برای حفظ اطلاعات ترتیبی این دسته بود. در واقع آگهی‌ای که «پنج یا بیشتر» اتاق دارد از نظر اطلاعاتی با آگهی چهار اتاقه متفاوت است، اما تعداد دقیق اتاق‌ها از داده در دسترس نیست.

این تصمیم یکی از نمونه‌هایی بود که نشان داد در پروژه‌های واقعی، تبدیل داده فقط یک عملیات فنی نیست؛ باید معنای متغیر نیز حفظ شود.

استانداردسازی سال ساخت

construction_year نیز وضعیت مشابهی داشت. بخشی از مقادیر به صورت اعداد فارسی و بخشی به صورت عبارات‌های متنی ثبت شده بودند. همچنین مقادیری مانند «قبل از ۱۳۷۰» وجود داشت که مستقیماً قابل استفاده به عنوان عدد نبودند.

برای حل این مسئله، ابتدا ارقام فارسی به قالب عددی استاندارد تبدیل شدند و سپس عبارات‌های خاص نیز به مقدار عددی مناسب تبدیل شدند. برای نمونه، «قبل از ۱۳۷۰» با مقدار 1369 نمایش داده شد.

در این مرحله یک نسخه استاندارد شده از سال ساخت ایجاد شد که بعداً برای ساخت ویژگی‌های جدید مورد استفاده قرار گرفت.

3-3 بررسی و مدیریت داده‌های گمشده

بعد از استانداردسازی متغیرها، مرحله بعد بررسی دقیق missing values بود. تحلیل اکتشافی نشان داد که داده‌های گمشده در دیتاست پراکنده نیستند و در بسیاری از موارد الگوی ساختاری دارند.

در مجموعه آگهی‌های فروش مسکونی، حدود 20.6 درصد از رکوردها اطلاعاتی درباره تعداد اتاق و سال ساخت نداشتند. نکته مهمتر این بود که این دو متغیر در بسیاری از رکوردها تقریباً همزمان گم شده بودند. بررسی بیشتر نشان داد که این مسئله تا حد زیادی با نوع آگهی ارتباط دارد و در آگهی‌های زمین طبیعی است که اطلاعاتی مانند تعداد اتاق وجود نداشته باشد.

این مشاهده باعث شد رویکرد ساده‌ای مثل حذف تمام ردیف‌های دارای missing کنار گذاشته شود.

اگر تمام ردیف‌هایی که rooms_count یا construction_year نداشتند حذف می‌شدند، بخشی از ساختار واقعی بازار، به‌خصوص آگهی‌های زمین، از داده حذف می‌شد. از طرف دیگر، پر کردن همه این مقادیر با میانگین نیز می‌توانست معنای داده را مخدوش کند.

به همین دلیل علاوه بر نگهداشتن missing values برای استفاده در مرحله preprocessing، دو متغیر کمکی نیز ایجاد شد:

has_rooms_info

و

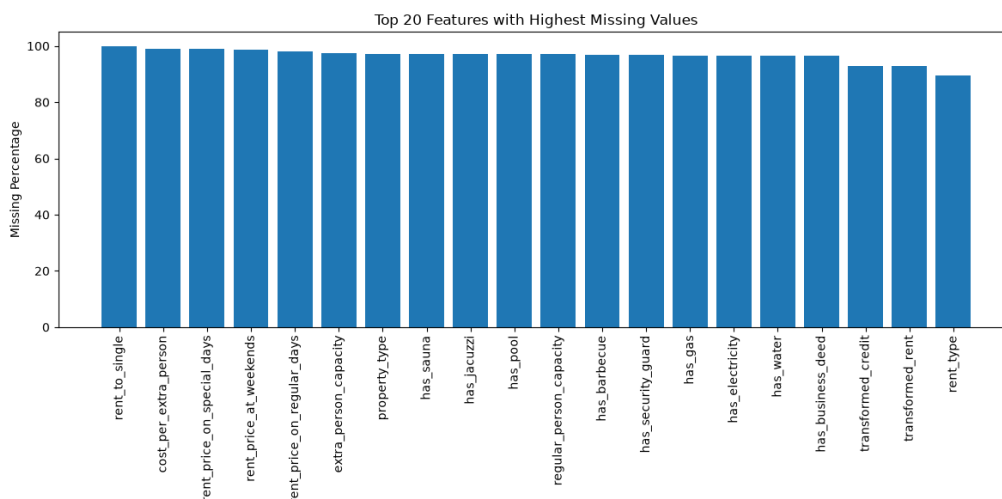
has_construction_year

این دو متغیر مشخص می‌کنند که آیا اطلاعات اصلی مربوط به اتاق و سال ساخت در آگهی وجود داشته است یا خیر. این تصمیم اهمیت زیادی داشت، زیرا «نبودن اطلاعات» در یک آگهی ملکی ممکن است خودش بخشی از رفتار ثبت آگهی باشد و صرفاً یک نقص تصادفی در دیتاست نباشد.

سایر missing values

در مقابل، بعضی ویژگی‌ها missing بسیار کمی داشتند. برای مثال building_size فقط تعداد بسیار محدودی مقدار گمشته داشت. بنابراین لازم نبود برای چنین متغیرهایی راهکار پیچیده‌ای طراحی شود و imputation آماری در مرحله preprocessing کافی بود.

neighborhood_slug شرایط متفاوتی داشت و missing بودن آن بسیار بیشتر بود. در اینجا نیز حذف رکوردها منطقی نبود، زیرا نبودن اطلاعات محله می‌توانست به دلیل نحوه ثبت آگهی باشد. بنابراین missing این متغیر به عنوان یک دسته مستقل در مرحله encoding حفظ شد.



3-4 شناسایی مقادیر نامعتبر و مقادیر شدید

یکی از حساس‌ترین قسمت‌های پاک‌سازی مربوط به متغیر قیمت بود.

در داده‌های فروش مسکونی، مقدار صفر برای price_value از نظر اقتصادی نمی‌توانست قیمت واقعی یک ملک فروشی باشد. بنابراین 1,703 رکورد دارای قیمت صفر به مقدار گم‌شده تبدیل شدند.

در نتیجه، تعداد رکوردهای فاقد قیمت از 23,744 به 25,447 رسید.

این تبدیل یک تفاوت مهم با حذف رکورد ایجاد کرد. رکوردهای فاقد قیمت مستقیماً از مجموعه مورد استفاده برای آموزش مدل حذف شدند، اما داده اصلی همچنان حفظ شد و مشخص بود که مشکل از کجا ایجاد شده است.

پس از این اصلاح، تعداد رکوردهای معتبر قابل استفاده برای مدل به 533,261 رسید.

مسئله اندازه ملک

building_size نیز توزیع بسیار شدیدی داشت. میانه اندازه ملک حدود 115 متر مربع بود، اما در صدک‌های بالاتر مقدارها به سرعت افزایش پیدا می‌کردند:

صدک 75 حدود 200 مترمربع

صدک 90 حدود 400 مترمربع

صدک 95 حدود 1,000 مترمربع

صدک 99 حدود 7,500 مترمربع

صدک 99.5 حدود 23,000 مترمربع

صدک 99.9 حدود 700,000 مترمربع

بیشینه حدود 10 میلیون مترمربع

در نگاه اول چنین اعدادی می‌توانستند شبیه خطای داده به نظر برسند. اما بررسی cat3_slug و تفاوت میان آپارتمان، خانه/ویلا و زمین نشان داد که نمی‌توان صرفاً با یک قانون آماری همه این موارد را حذف کرد.

به همین دلیل تصمیم گرفته شد مقادیر شدید اندازه ملک حذف نشوند.

در عوض، برای کمک به مدل در مواجهه با توزیع شدید، ویژگی دیگری به نام log_building_size ساخته شد.

همچنین ویژگی is_land ایجاد شد تا مدل بتواند تفاوت ساختاری آگهی‌های زمین را در نظر بگیرد.

این تصمیم یکی از مهم‌ترین نقاط این مرحله بود: داده پرت با داده اشتباه یکی فرض نشد.

3-5 مهندسی ویژگی‌ها

بعد از پاک‌سازی اولیه، مرحله مهندسی ویژگی آغاز شد. هدف در این بخش ساخت متغیرهایی بود که اطلاعات موجود در داده خام را برای مدل قابل استفاده‌تر کنند، بدون اینکه اطلاعات مربوط به هدف به عنوان ورودی وارد مدل شود.

سن ملک

از سال ساخت استاندارد شده، ویژگی `property_age` ایجاد شد:

سن ملک با استفاده از سال مرجع 1403 محاسبه شد.

در داده‌های موجود، این ویژگی برای 425,119 رکورد مقدار معتبر داشت. میانه سن حدود 8 سال، میانگین حدود 9.46 سال و بیشینه حدود 35 سال بود.

استفاده از سن ملک در کنار سال ساخت باعث شد مدل بتواند اثر عمر ساختمان را به شکلی مستقیم‌تر دریافت کند.

شاخص‌های وجود اطلاعات

همان‌طور که در بخش missingness توضیح داده شد، دو ویژگی زیر اضافه شدند:

`has_rooms_info`

`has_construction_year`

این ویژگی‌ها نشان می‌دهند که اطلاعات مربوطه در آگهی ثبت شده است یا خیر.

تبدیل لگاریتمی اندازه ملک

به دلیل توزیع شدید `building_size`، ویژگی:

`log_building_size`

نیز ساخته شد.

این ویژگی قرار نبود جایگزین اندازه اصلی شود. هر دو ویژگی حفظ شدند تا مدل بتواند هم از اندازه خام و هم از نمایش فشرده‌تر آن استفاده کند.

فراوانی آگهی در شهر

ویژگی دیگری به نام `city_listing_frequency` نیز بررسی و ساخته شد. ایده این ویژگی این بود که تعداد آگهی‌های موجود در هر شهر می‌تواند اطلاعاتی درباره ساختار بازار و حجم فعالیت آن شهر در اختیار مدل قرار دهد.

اما در همین مرحله یک نکته مهم درباره نشت اطلاعات مشخص شد: اگر این فراوانی با استفاده از کل دیتاست محاسبه شود، اطلاعات validation و test وارد ویژگی آموزش می‌شود.

بنابراین در طراحی نهایی، این ویژگی باید فقط از داده‌های train محاسبه شده و سپس به validation و test نگاشت شود. این موضوع بعداً در مرحله آماده‌سازی مدل به صورت صریح اجرا شد.

3-6 آماده‌سازی هدف مدل

در بررسی‌های مرحله قبل مشخص شده بود که price_value توزیع بسیار چوله‌ای دارد و چند مقدار بسیار بزرگ باعث ایجاد فاصله شدید میان قیمت‌های معمول و انتهای توزیع می‌شوند.

به همین دلیل هدف اصلی مدل به صورت تبدیل شده تعریف شد:

$$\log_price = \log1p(price_value)$$

این تبدیل باعث شد مدل به جای تلاش برای یادگیری مستقیم مقادیر خام قیمت، روی مقیاس لگاریتمی کار کند.

بعداً صحت این تبدیل نیز بررسی شد و مشخص شد که مقدار log_price دقیقاً با np.log1p(price_value) منطبق است و اختلاف عددی معناداری وجود ندارد.

در همین نقطه یک اصل مهم پروژه نیز تثبیت شد price_per_sqm: فقط به عنوان یک متغیر مشتق‌شده برای تحلیل استفاده شود و هرگز به عنوان predictor وارد مدل نشود؛ زیرا قیمت کل ملک مستقیماً در ساخت آن دخیل است و استفاده از آن می‌توانست باعث نشت هدف شود.

3-7 تشخیص ویژگی‌های نامناسب برای مدل

در جریان آماده‌سازی مشخص شد که همه 60 ستون اولیه برای مدل‌سازی مناسب نیستند.

برای نمونه، price_mode در داده‌های فروش عملاً فقط یک مقدار، یعنی «مقطوع»، داشت. چنین متغیری اطلاعاتی برای تفکیک نمونه‌ها ایجاد نمی‌کرد و بنابراین کنار گذاشته شد.

همچنین تعداد قابل توجهی از متغیرهای مرتبط با اجاره، رهن و اعتبار در داده‌های فروش مسکونی به طور کامل فاقد مقدار بودند. این متغیرها نیز در مدل نهایی نقشی نداشتند.

این مرحله کمک کرد تا مدل با مجموعه‌ای کوچک‌تر و معنادارتر از ویژگی‌ها کار کند، به جای اینکه صرفاً تمام ستون‌های موجود در فایل را وارد الگوریتم کنیم.

3-8 بررسی رکوردهای تکراری

موضوع دیگری که در این مرحله بررسی شد، duplicate ها بود.

در بررسی ترکیب ویژگی‌های مورد استفاده، 105,806 ردیف دارای ترکیب ویژگی یکسان بودند. این رکوردها به 36,199 گروه ترکیبی تکراری تقسیم می‌شدند و در 34,570 گروه، حتی target نیز بین رکوردهای یکسان متفاوت بود. این نتیجه بسیار مهم بود.

اگر صرفاً بر اساس یکسان بودن ویژگی‌ها همه duplicate ها حذف می‌شدند، ممکن بود آگهی‌های واقعی مختلفی که مشخصات ملکی مشابه داشتند اما قیمت متفاوتی ثبت کرده بودند، از بین بروند.

بنابراین تصمیم گرفته شد duplicate ها به شکل کورکورانه حذف نشوند.

این تصمیم با ماهیت داده نیز سازگار بود. دو آگهی می‌توانند هر دو مربوط به یک آپارتمان 100 متری در یک محله باشند، اما توسط دو کاربر مختلف، در زمان‌های مختلف یا با قیمت‌های متفاوت ثبت شده باشند. بنابراین یکسان بودن feature ها الزاماً به معنی یکسان بودن رکوردها نیست.

3-9 مجموعه ویژگی‌های آماده برای مدل

پس از این مراحل، مجموعه اولیه ویژگی‌های عددی، دسته‌ای و باینری مشخص شد.

ویژگی‌های عددی شامل مواردی مانند:

building_size، log_building_size، rooms_count_numeric، construction_year_model، created_month_index، city_listing_frequency، land_size، property_age بودند.

ویژگی‌های دسته‌ای شامل:

cat2_slug، cat3_slug، city_slug، neighborhood_slug، user_type بودند.

همچنین سه ویژگی باینری:

has_rooms_info، has_construction_year، is_land

به عنوان اطلاعات کمکی در نظر گرفته شدند.

البته این مجموعه در مرحله مدل‌سازی دوباره مورد ارزیابی قرار گرفت و مشخص شد که بعضی از ویژگی‌ها، مانند user_type و city_listing_frequency، به دلیل رفتار نامتوازن و مسئله تعمیم زمانی نیاز به بررسی بیشتر دارند.

10-3 خروجی نهایی مرحله پاکسازی

در پایان این مرحله، مجموعه داده از حالت خام و ناهمگون به یک مجموعه قابل استفاده برای مدل‌سازی تبدیل شد. از 558,708 آگهی فروش مسکونی اولیه، پس از کنار گذاشتن رکوردهایی که target معتبر نداشتند، 533,261 رکورد برای مدل‌سازی باقی ماندند. در این مجموعه:

target به صورت log_price تعریف شد.

قیمت‌های صفر نامعتبر تشخیص داده شدند.

متغیرهای متنی عددی استاندارد شدند.

سن ملک استخراج شد.

شاخص‌های وجود اطلاعات ایجاد شدند.

اندازه ملک به مقیاس لگاریتمی نیز تبدیل شد.

متغیرهای نامناسب یا کاملاً خالی کنار گذاشته شدند.

price_per_sqm از مجموعه predictorها حذف شد تا leakage ایجاد نشود.

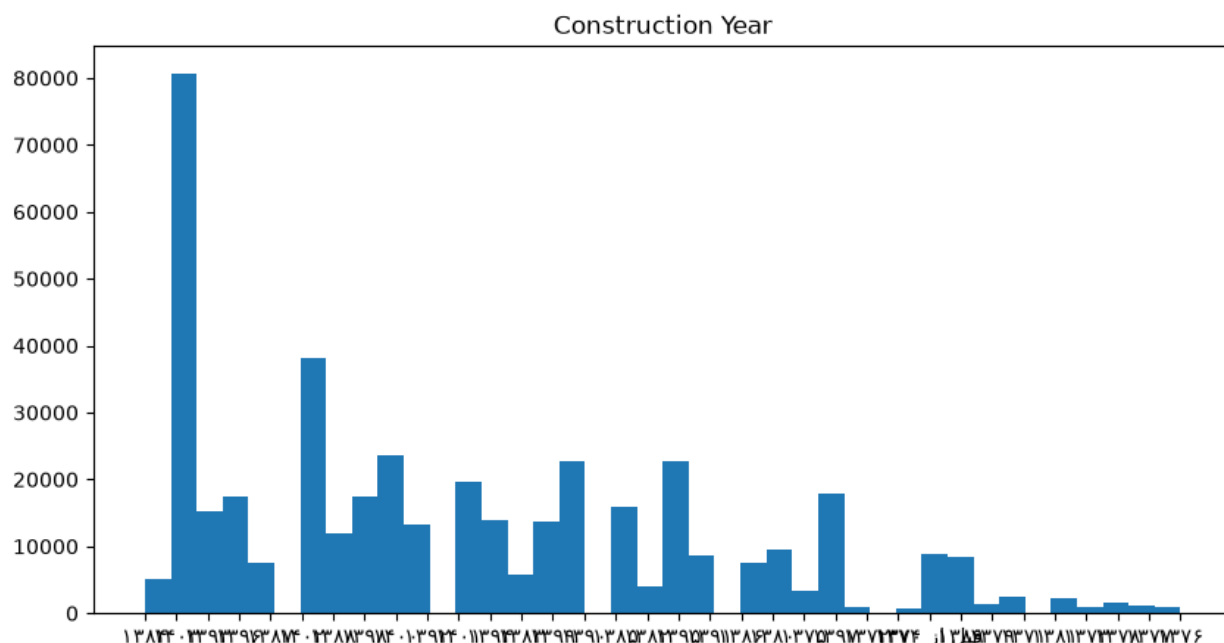
Duplicate ها بدون بررسی معنایی حذف نشدند.

مقادیر شدید به صورت کورکورانه حذف نشدند.

در این مرحله تصور اولیه این بود که پس از پاکسازی، مسئله قیمت‌گذاری باید به شکل محسوسی ساده‌تر شود. با این حال، هنوز یک سؤال مهم باقی مانده بود: آیا این ویژگی‌ها واقعاً می‌توانند قیمت ملک را در دوره‌های زمانی آینده توضیح دهند یا فقط روی داده‌های گذشته عملکرد خوبی خواهند داشت؟ همین سؤال مسیر مرحله بعد را مشخص کرد.

بررسی تاریخ آگهی‌ها نشان داد که داده‌ها از فوریه 2021 تا مارس 2025 را پوشش می‌دهند، اما توزیع زمانی یکنواخت نیست. بخش بزرگی از داده‌ها مربوط به ماه‌های میانی و پایانی سال 2024 بود. بنابراین استفاده از یک random split معمولی می‌توانست تصویری بیش از حد خوش‌بینانه از عملکرد مدل ایجاد کند.

از اینجا، پروژه وارد مرحله آماده‌سازی مدل و طراحی یک ارزیابی زمانی شد؛ جایی که مسئله فقط «مدل چقدر خوب یاد می‌گیرد؟» نبود، بلکه این بود که «مدل چقدر می‌تواند روی آگهی‌های مربوط به یک بازه زمانی بعدتر تعمیم پیدا کند؟»



جمع‌بندی فصل سوم

فصل پاکسازی و مهندسی ویژگی عملاً مرز میان داده خام و داده قابل مدل‌سازی بود. در این مرحله مشخص شد که بخش مهمی از کیفیت یک پروژه یادگیری ماشین نه از انتخاب الگوریتم، بلکه از تصمیم‌هایی می‌آید که قبل از آموزش مدل گرفته می‌شوند.

مهم‌ترین تصمیم این مرحله حفظ اطلاعات واقعی بازار در کنار کنترل خطاهای واضح بود. به همین دلیل نه همه missingها حذف شدند، نه همه outlierها، و نه همه رکوردهای تکراری. در عوض، برای هر مسئله تلاش شد دلیل ایجاد آن بررسی شود و سپس یک راه حل متناسب با ساختار داده انتخاب شود.

با پایان این مرحله، داده برای ورود به فرآیند رسمی train/validation/test و ساخت pipeline آماده بود. فصل بعد به همین مرحله اختصاص پیدا می‌کند: آماده‌سازی نهایی داده، طراحی split زمانی، preprocessing، کنترل leakage و ساخت baseline و مدل‌های اولیه.

فصل چهارم

آماده‌سازی نهایی داده و طراحی فرآیند مدل‌سازی

4-1 مقدمه

با پایان مرحله پاک‌سازی، مسئله اصلی پروژه از «آیا داده قابل استفاده است؟» به «چگونه باید این داده را برای ارزیابی منصفانه مدل آماده کرد؟» تغییر کرد.

در این مرحله دیگر هدف صرفاً ساخت یک جدول تمیز نبود. باید مشخص می‌شد کدام رکوردها برای آموزش، اعتبارسنجی و آزمون استفاده شوند، preprocessing چگونه انجام شود، چه ویژگی‌هایی واقعاً وارد مدل شوند و مهم‌تر از همه، چگونه از نشت اطلاعات میان بخش‌های مختلف داده جلوگیری شود.

از همان ابتدا مشخص بود که یک تقسیم تصادفی ساده برای این پروژه ایده‌آل نیست. آگهی‌های ملکی به زمان وابسته‌اند و بازار نیز در طول زمان تغییر می‌کند. اگر آگهی‌های ماه‌های مختلف به صورت تصادفی بین train و test پخش شوند، مدل ممکن است اطلاعات بسیار مشابهی از یک دوره زمانی را هم در آموزش و هم در آزمون ببیند.

بنابراین تصمیم گرفته شد ارزیابی نهایی بر اساس زمان انجام شود.

4-2 بررسی ساختار زمانی داده

اولین کار بررسی دقیق created_at_month بود.

داده‌های فروش مسکونی بازه‌ای از فوریه 2021 تا مارس 2025 را پوشش می‌دادند و در مجموع 39 ماه مختلف در آن دیده می‌شد. اما تعداد آگهی‌ها در این ماه‌ها تقریباً یکنواخت نبود.

بخش قابل توجهی از داده‌ها مربوط به ماه‌های سال 2024 بود، در حالی که ماه‌های ابتدایی‌تر بسیار کم‌تراکم‌تر بودند. همچنین در انتهای بازه، ژانویه تا مارس 2025 تعداد آگهی‌ها دوباره بسیار کاهش پیدا می‌کرد.

این وضعیت باعث شد یک random split معمولی انتخاب مناسبی به نظر نرسد. در چنین شرایطی، ممکن بود مدل روی داده‌هایی ارزیابی شود که از نظر زمانی بسیار نزدیک به داده‌های آموزش هستند و در نتیجه عملکرد آن بهتر از شرایط واقعی نمایش داده شود.

در یک کاربرد واقعی، سناریوی منطقی‌تر این است که با استفاده از آگهی‌های گذشته، قیمت آگهی‌های آینده پیش‌بینی شود.

به همین دلیل تصمیم نهایی بر اساس یک split زمانی گرفته شد.

4-3 طراحی Train ، Validation و Test

تقسیم زمانی نهایی به این شکل انجام شد:

Train: 2024 تا سپتامبر

Validation: 2024 اکتبر و نوامبر

Test: 2024 دسامبر تا مارس 2025

نتیجه این تقسیم:

تعداد رکورد	مجموعه
328,025	Train
136,590	Validation
68,646	Test

بنابراین حدود 61.5 درصد داده در train ، حدود 25.6 درصد در validation و حدود 12.9 درصد در test قرار گرفت.

این تقسیم بندی از نظر زمانی یک ترتیب طبیعی ایجاد می‌کرد:

آموزش ← دوره بعدی برای اعتبارسنجی ← دوره بعدی برای آزمون نهایی

در نتیجه test واقعاً نقش داده آینده را پیدا کرد.

4-4 جلوگیری از نشت اطلاعات

یکی از موضوعاتی که در این مرحله اهمیت بیشتری پیدا کرد، جلوگیری از leakage بود.

نمونه مشخص آن city_listing_frequency بود. این ویژگی تعداد آگهی‌های مربوط به یک شهر را نشان می‌داد. اگر تعداد آگهی‌ها با استفاده از کل دیتاست محاسبه می‌شد، در محاسبه ویژگی train از اطلاعات validation و test نیز استفاده می‌شد.

در ظاهر این مسئله کوچک به نظر می‌رسید، اما از نظر روش‌شناسی می‌توانست ارزیابی مدل را غیرواقعی کند.

بنابراین روش نهایی به این صورت تعیین شد:

ابتدا فراوانی شهرها فقط روی train محاسبه شود؛ سپس همین mapping روی validation و test اعمال شود.

به این ترتیب validation و test هیچ نقشی در ساخت این ویژگی نداشتند.

این تصمیم یکی از مهم‌ترین کنترل‌های روش‌شناختی پروژه بود، زیرا هدف فقط گرفتن یک عدد بهتر برای metric نبود؛ هدف این بود که metric واقعاً نشان دهد مدل در شرایط استفاده از داده‌های آینده چه عملکردی دارد.

4-5 ساخت ویژگی زمانی

در تحلیل اکتشافی مشخص شده بود که استفاده مستقیم از `created_at_month` به عنوان یک متغیر دسته‌ای می‌توانست تعداد زیادی `category` ایجاد کند و در عین حال ارتباط زمانی را به شکل مناسبی نمایش ندهد.

بنابراین یک ویژگی عددی به نام:

`created_month_index`

ساخته شد.

این متغیر ترتیب زمانی ماه‌ها را حفظ می‌کرد و به مدل اجازه می‌داد تغییرات زمانی را به صورت یک روند عددی دریافت کند.

این انتخاب همچنین تعداد ویژگی‌های `categorical` را کنترل می‌کرد.

4-6 طراحی preprocessing

پس از مشخص شدن `split`، مسئله بعدی آماده‌سازی ویژگی‌ها بود.

سه گروه اصلی برای `preprocessing` تعریف شد:

1. ویژگی‌های عددی

2. ویژگی‌های `categorical`

3. ویژگی‌های `binary`

برای ویژگی‌های عددی، مقادیر گم‌شده با `median` مربوط به داده آموزش جایگزین شدند.

برای ویژگی‌های `categorical`، `missing` به یک دسته مشخص با نام `__MISSING__` تبدیل شد.

این تصمیم مخصوصاً برای متغیرهایی مانند `neighborhood_slug` و `user_type` اهمیت داشت. در این داده‌ها `missing` بودن بعضی ویژگی‌ها به اندازه‌ای زیاد بود که حذف آن‌ها عملاً بخش بزرگی از نمونه‌ها را از بین می‌برد.

در نتیجه، `missing` به جای اینکه همیشه یک خطا در نظر گرفته شود، به عنوان یک وضعیت قابل مشاهده برای مدل حفظ شد.

4-7 One-Hot Encoding و کنترل category های جدید

برای ویژگی‌های categorical از One-Hot Encoding استفاده شد و تنظیم `handle_unknown="ignore"` در نظر گرفته شد.

این تنظیم به دلیل ماهیت `split` زمانی اهمیت داشت.

برای مثال ممکن بود یک شهر یا محله در `validation` یا `test` وجود داشته باشد که در `train` دیده نشده باشد. اگر `encoder` چنین category هایی را به عنوان خطای تلقی می‌کرد، فرآیند پیش‌بینی متوقف می‌شد.

در داده واقعی نیز دقیقاً چنین مواردی مشاهده شد.

در `validation` دو شهر خارج از category های `train` وجود داشتند و در `test` شهر جدیدی نسبت به `train` مشاهده نشد. برای `neighborhood` نیز 29 category جدید در `validation` و 48 category جدید در `test` وجود داشت.

بنابراین استفاده از `handle_unknown="ignore"` فقط یک انتخاب تئوری نبود و با ساختار واقعی داده سازگار بود.

4-8 ابعاد داده پس از preprocessing

پس از `fit` کردن preprocessing فقط روی `train` و اعمال آن روی `validation` و `test`، تعداد ویژگی‌های نهایی به 1,550 ستون رسید.

ابعاد سه مجموعه به این صورت بود:

ویژگی پس از preprocessing	ردیف	مجموعه
1,550	328,025	Train
1,550	136,590	Validation
1,550	68,646	Test

این نکته که هر سه مجموعه دقیقاً تعداد ویژگی یکسانی داشتند، یکی از کنترل‌های مهم `pipeline` بود.

همچنین بعد از preprocessing دیگر `missing value` در ماتریس ورودی مدل وجود نداشت.

در اینجا یک نکته فنی دیگر نیز مشخص شد: برخی `flag`ها در ابتدا از نوع `boolean` بودند و استفاده مستقیم از آن‌ها در ترکیب با `pipeline` باعث خطا در `SimpleImputer` شد. بنابراین این متغیرها به نوع عددی کوچک (`int8`) تبدیل شدند.

این اتفاق نمونه‌ای از مشکلاتی بود که در مرحله پیاده‌سازی واقعی `pipeline` ظاهر می‌شوند و معمولاً در طراحی مفهومی اولیه دیده نمی‌شوند.

4-9 بررسی cardinality ویژگی‌های categorical

بعد از encoding، cardinality ویژگی‌های دسته‌ای نیز بررسی شد.

cat2_slug فقط یک مقدار در داده فروش مسکونی داشت و cat3_slug سه نوع اصلی را مشخص می‌کرد:

apartment-sell

house-villa-sell

plot-old

city_slug و neighborhood_slug تعداد category بسیار بیشتری داشتند و طبیعتاً بخش عمده‌ای از افزایش ابعاد One-Hot Encoding از همین دو ویژگی ایجاد شده بود.

این بررسی بعداً در انتخاب ویژگی‌ها نیز اهمیت پیدا کرد. زیرا هدف این نبود که صرفاً بیشترین تعداد feature ممکن ساخته شود؛ باید مشخص می‌شد کدام متغیر واقعاً اطلاعات قابل استفاده برای مدل دارد.

4-10 بررسی Outlier ها

در این مرحله یک audit آماری برای outlier های مجموعه train انجام شد.

برای مثال در building_size با استفاده از روش IQR حدود 10 درصد رکوردها در محدوده outlier قرار گرفتند. برای land_size این مقدار حدود 2.7 درصد بود.

اما مهم‌ترین نتیجه این بررسی این بود که همه outlier ها الزاماً داده اشتباه نیستند.

در مورد building_size، وجود زمین‌ها و املاک بسیار بزرگ می‌توانست دلیل واقعی برای دم سنگین توزیع باشد.

در rooms_count_numeric نیز روش IQR نتیجه‌ای حدود 35.6 درصد outlier نشان می‌داد، اما مشخص بود که استفاده مکانیکی از IQR برای یک متغیر گسسته و محدود مناسب نیست.

بنابراین نتیجه این audit حذف outlier ها نبود.

در عوض، این تحلیل به عنوان یک کنترل کیفیت ثبت شد و تصمیم گرفته شد داده‌های شدید بدون دلیل معنایی حذف نشوند.

این تصمیم بعداً اهمیت خود را در مدل نهایی نشان داد، زیرا حذف گسترده چنین رکوردهایی می‌توانست مدل را از بخشی از ساختار واقعی بازار جدا کند.

فصل پنجم

ساخت مدل‌های پایه و مدل نهایی

5-1 مقدمه

پس از آماده شدن pipeline، پروژه برای اولین بار وارد مرحله واقعی مدل‌سازی شد.

در این نقطه دو سؤال اصلی وجود داشت:

اول اینکه آیا ویژگی‌های آماده‌شده اصلاً توانایی پیش‌بینی قیمت را دارند؟

دوم اینکه اگر مدل بتواند الگو را یاد بگیرد، کدام روش می‌تواند بین سادگی، عملکرد و قابلیت تفسیر تعادل مناسبی ایجاد کند؟

برای پاسخ به سؤال اول، یک baseline ساده لازم بود. برای سؤال دوم نیز یک مدل خطی و سپس CatBoost مورد استفاده قرار گرفت.

Baseline 5-2 با پیش‌بینی میانه

اولین baseline بسیار ساده بود: برای تمام نمونه‌ها مقدار میانه target آموزشی پیش‌بینی شد.

نتیجه این baseline :

MAE روی log-price: 1.118531

RMSE روی log-price: 2.159330

R^2 : -0.006105

این مدل طبیعتاً انتظار زیادی ایجاد نمی‌کرد، اما وجود آن برای مقایسه ضروری بود.

مدل واقعی باید بتواند به شکل مشخصی بهتر از این baseline عمل کند؛ در غیر این صورت پیچیدگی مدل توجیهی نداشت.

Ridge Regression 5-3

مدل بعدی Ridge Regression بود.

نتیجه Ridge :

MAE_log: 1.062685

RMSE_log: 2.090108

R²: 0.057367

بنابر این نسبت به baseline پیشرفت ایجاد شد، اما هنوز مقدار زیادی از تغییرات قیمت توسط مدل توضیح داده نمی‌شد. این نتیجه با ماهیت مسئله نیز سازگار بود. قیمت ملک فقط یک رابطه خطی ساده میان چند متغیر عددی نیست. نوع ملک، شهر، محله، اندازه، سال ساخت و سایر عوامل می‌توانند روابط غیرخطی و تعاملی ایجاد کنند. در این مرحله انتظار شکل‌گرفته این بود که یک مدل درختی مبتنی بر boosting بتواند ساختار پیچیده‌تری را استخراج کند.

5-4 ورود CatBoost

برای مدل اصلی از CatBoost استفاده شد؛ دلیل اصلی انتخاب آن توانایی مناسب در کار با ویژگی‌های categorical با cardinality بالا، مانند شهر و محله، و همچنین توانایی یادگیری روابط غیر خطی بود.

نسخه استفاده‌شده CatBoost برابر 1.2.10 بود.

مدل اولیه با 1000 iteration آموزش داده شد.

نتیجه validation :

MAE_log: 0.855118

RMSE_log: 1.939262

R²: 0.188519

این نتیجه نسبت به Ridge بهبود محسوسی داشت.

به‌خصوص کاهش MAE از حدود 1.06 به حدود 0.855 نشان داد که ساختار غیرخطی داده واقعاً اهمیت دارد.

5-5 افزایش تعداد iteration

در مرحله بعد تعداد iteration به 1500 افزایش داده شد.

مدل در iteration شماره 1374 به بهترین validation RMSE رسید.

نتیجه:

MAE_log: 0.852131

RMSE_log: 1.938742

R²: 0.188954

بهبود نسبت به مدل iteration 1000 بسیار بزرگ نبود، اما نشان داد که مدل هنوز کمی ظرفیت برای یادگیری بیشتر دارد.

در اینجا مشخص شد که صرفاً افزایش تعداد درخت‌ها قرار نیست جهش بزرگی در عملکرد ایجاد کند. بنابراین تمرکز از «بیشتر کردن مدل» به «بررسی کیفیت ویژگی‌ها» منتقل شد.

5-6 اهمیت ویژگی‌ها

تحلیل feature importance مدل CatBoost نشان داد که مهم‌ترین متغیرها عمدتاً مربوط به موقعیت جغرافیایی و اندازه ملک هستند.

در مدل اولیه، city_slug مهم‌ترین ویژگی بود و پس از آن:

log_building_size

building_size

user_type

rooms_count_numeric

city_listing_frequency

land_size

neighborhood_slug

cat3_slug

قرار داشتند.

این نتیجه با یافته‌های مرحله EDA نیز همخوانی داشت.

از طرف دیگر، اهمیت پایین is_land و flag های مربوط به وجود اطلاعات نشان داد که وجود یا نبود این اطلاعات به تنهایی نقش کمتری نسبت به ویژگی‌های اصلی ملک و موقعیت جغرافیایی دارد.

5-7 بررسی user_type و city_listing_frequency

در اینجا یک نکته مهم ظاهر شد.

user_type در train مقدارهای بیشتری داشت، اما missingness آن در validation و test به شکل بسیار شدید افزایش پیدا می‌کرد. در train حدود 51.7 درصد این متغیر missing بود، در حالی که در validation حدود 96.5 درصد و در test حدود 99.9 درصد missing بود.

این اختلاف باعث شد نسبت به قابلیت تعمیم این ویژگی تردید ایجاد شود.

city_listing_frequency نیز به دلیل حساسیت آن به نحوه محاسبه و زمان، نیاز به بررسی داشت.

بنابراین یک آزمایش ablation انجام شد: هر دو ویژگی از مدل حذف شدند تا مشخص شود آیا واقعاً به عملکرد کمک می‌کنند یا ممکن است باعث وابستگی نامطلوب به ساختار داده شوند.

Ablation Study 5-8

پس از حذف user_type و city_listing_frequency، مدل CatBoost دوباره آموزش داده شد.

نتیجه validation :

MAE_log: 0.787236

RMSE_log: 1.933024

R^2 : 0.193732

نکته جالب این بود که حذف دو ویژگی نه تنها باعث افت مدل نشد، بلکه MAE و R^2 بهتر شدند.

این نتیجه یکی از مهمترین یافته‌های مدل‌سازی پروژه بود.

در واقع مشخص شد که بیشتر بودن تعداد featureها لزوماً به معنی مدل بهتر نیست. بعضی ویژگی‌ها ممکن است در ظاهر اطلاعات مفیدی داشته باشند، اما در شرایط زمانی واقعی یا به دلیل missingness نامتوازن، به تعمیم مدل کمک نکنند.

در مدل ablation، اهمیت city_slug حتی بیشتر شد و به حدود 25.4 درصد رسید. پس از آن log_building_size، rooms_count_numeric و neighborhood_slug قرار گرفتند.

این نتیجه تصویر واضح‌تری از عوامل اصلی مدل ایجاد کرد.

5-9 حذف ویژگی ثابت cat2_slug

در مجموعه فروش مسکونی، cat2_slug عملاً مقدار ثابتی داشت؛ زیرا تمام نمونه‌ها در شاخه Residential Sale قرار گرفته بودند.

در نتیجه این ویژگی نیز از candidate نهایی حذف شد.

مجموعه نهایی ویژگی‌های CatBoost شامل اطلاعات مربوط به:

اندازه ملک

اندازه لگاریتمی

تعداد اتاق

سال ساخت

سن ملک

اندازه زمین

زمان

شهر

محله

نوع ملک

flag های اطلاعاتی

شد.

ویژگی‌های categorical اصلی نیز به city_slug، neighborhood_slug و cat3_slug محدود شدند.

5-10 مقایسه عمق‌های مختلف CatBoost

برای انتخاب عمق مناسب، سه مدل با depth های 6، 8 و 10 مقایسه شدند.

نتایج validation :

Depth	MAE_log	RMSE_log	R ²
6	0.786920	1.932452	0.194208
8	0.787523	1.932495	0.194173
10	0.788560	1.934104	0.192831

اختلاف depth های 6 و 8 بسیار کم بود، اما 6 depth کمی عملکرد بهتری داشت و مدل ساده‌تری نیز ایجاد می‌کرد.

بنابراین depth=6 انتخاب شد.

این انتخاب بر اساس یک تفاوت بزرگ در metric نبود؛ بلکه به دلیل عملکرد بسیار نزدیک، سادگی بیشتر و پرهیز از افزایش غیرضروری پیچیدگی مدل انجام شد.

مدل نهایی candidate در validation به این نتایج رسید:

MAE_log: 0.786920

RMSE_log: 1.932452

R²: 0.194208

5-11 آموزش مدل نهایی روی Train + Validation

پس از انتخاب مدل و hyperparameter های نهایی، دیگر validation برای انتخاب مدل استفاده نمی‌شد.

مدل با ترکیب train و validation مجدداً آموزش داده شد.

تعداد داده‌های این مجموعه:

464,615 رکورد

بود.

مدل نهایی با:

depth = 6

learning rate = 0.05

iterations = 1488

random seed = 42

آموزش داده شد.

این مدل سپس بدون هیچ آموزش مجددی روی test نهایی ارزیابی شد.

در این نقطه برای اولین بار مشخص می‌شد که مدل انتخاب‌شده واقعاً روی داده‌ای که در فرآیند انتخاب مدل دیده نشده است چگونه عمل می‌کند.

فصل ششم

ارزیابی نهایی مدل و تحلیل خطا

6-1 نتیجه نهایی روی Test

پس از آموزش نهایی، مدل روی test شامل آگهی‌های دسامبر 2024 تا مارس 2025 اجرا شد. در بررسی اولیه مشخص شد که بخش کوچکی از قیمت‌ها شامل مقادیر غیرواقعی از نوع اعداد تکراری مانند: 1، 11، 111، 1111، 111111 و موارد مشابه است.

این مقادیر از نظر الگوی ثبت داده مشکوک بودند و در مرحله نهایی به عنوان یک مسئله کیفیت داده شناسایی شدند. تعداد کل این رکوردها در مجموعه model-ready برابر 8,712 بود که حدود 1.63 درصد داده‌ها را تشکیل می‌داد. این رکوردها بین سه نوع اصلی ملک نیز توزیع شده بودند و صرفاً محدود به یک دسته نبودند.

6-2 پاکسازی نهایی قیمت‌های تکراری

برای جلوگیری از اینکه این مقادیر مشکوک ارزیابی مدل را بیش از حد تحت تأثیر قرار دهند، رکوردهای دارای قیمت‌های تکراری مشخص از مجموعه ارزیابی حذف شدند. پس از این پاکسازی:

رکورد Train: 323,033

رکورد Validation: 134,189

رکورد Test: 67,327

باقی ماند.

مدل نهایی تمیز شده سپس روی test تمیز ارزیابی شد.

نتیجه نهایی:

MAE_log: 0.598094

RMSE_log: 1.447052

R²_log: 0.297366

و در مقیاس قیمت:

MAE_price: حدود 53.55 میلیارد

RMSE_price: حدود 1.294 تریلیون

این اعداد به عنوان نتایج نهایی پروژه ثبت شدند.

6-3 مقایسه Test و Validation

یکی از نکات مثبت نتیجه نهایی این بود که عملکرد مدل در test از نظر log-scale فاصله بسیار شدیدی با validation نداشت.

Validation پاکشده:

$$\text{MAE}_{\log} \approx 0.574$$

$$\text{RMSE}_{\log} \approx 1.378$$

$$R^2 \approx 0.310$$

Test پاکشده:

$$\text{MAE}_{\log} \approx 0.598$$

$$\text{RMSE}_{\log} \approx 1.447$$

$$R^2 \approx 0.297$$

بنابراین مدل در داده آینده کمی ضعیفتر عمل کرد، اما افت آن کنترل شده بود.

این موضوع با ماهیت ارزیابی زمانی نیز سازگار است: باید دشوارتر از validation باشد، زیرا مدل هیچکدام از آگهی‌های آن بازه را در آموزش ندیده است.

6-4 تفاوت عملکرد میان انواع ملک

وقتی خطاها بر اساس cat3_slug بررسی شدند، تفاوت مشخصی میان انواع ملک دیده شد.

در test پاکشده:

آپارتمان

تعداد: 41,424

MAE_log: 0.422

RMSE_log: 1.264

median absolute error: 0.210 حدود

خانه و ویلا

تعداد: 13,278

MAE_log: 0.814

RMSE_log: 1.981

median absolute error: 0.433 حدود

زمین

تعداد: 12,625

MAE_log: 0.949

RMSE_log: 1.342

median absolute error: 0.715 حدود

این تفاوت نشان داد که مسئله قیمت‌گذاری برای همه انواع ملک یکسان نیست. مدل در آگهی‌های آپارتمان خطای log بسیار پایین‌تری داشت، در حالی که خانه/ویلا و زمین پیچیدگی بیشتری داشتند. این نتیجه با مشاهدات قبلی نیز همخوانی داشت. برای مثال، زمین‌ها توزیع اندازه بسیار متفاوتی دارند و ویژگی‌های استاندارد مانند تعداد اتاق برای آن‌ها اغلب وجود ندارد.

6-5 بررسی خطاهای بزرگ

تحلیل خطا نشان داد که بزرگترین خطاهای مدل عمدتاً در رکوردهایی اتفاق می‌افتند که قیمت ثبت‌شده بسیار کوچک است. برای مثال، در test مواردی با قیمت‌های 2، 3، 5، 6، 7، 8، 12 و 20 واحد پولی مشاهده شدند، در حالی که ویژگی‌های ملک نشان می‌دادند این رکوردها از نظر اندازه و نوع ملک، املاک عادی با ارزش بسیار بالاتر هستند. برای نمونه، یکی از رکوردها مربوط به خانه‌ای حدود 360 مترمربع بود که قیمت ثبت‌شده آن 5 بود، در حالی که مدل قیمت بسیار بالاتری برای آن پیش‌بینی کرد. در نمونه دیگری، ملکی حدود 421 مترمربع قیمت 12 داشت و مدل حدود 19.55 میلیارد پیش‌بینی کرد.

از دید مدل، چنین اختلافی عجیب نیست؛ مدل بر اساس ویژگی‌های ملک و الگوهای بازار آموزش دیده و نمی‌تواند بفهمد که یک قیمت بسیار کوچک احتمالاً یک placeholder، خطای ورود اطلاعات یا نوع دیگری از ثبت غیرواقعی است. این تحلیل باعث شد مشخص شود که بخشی از خطای ظاهری مدل در واقع ناشی از کیفیت target است، نه الزاماً ناتوانی مدل در یادگیری رابطه قیمت و ویژگی‌های ملک.

6-6 بررسی عملکرد در سطوح مختلف قیمت

برای اینکه مشخص شود این مشکل تا چه اندازه metric کلی را تحت تأثیر قرار می‌دهد، ارزیابی روی چند آستانه قیمت نیز انجام شد.

برای کل test :

در نسخه اولیه $MAE_log = 0.8166$

$RMSE_log = 1.9901$

$R^2 = 0.1935$

اما وقتی فقط املاکی با قیمت بیشتر از 100 میلیون بررسی شدند:

$MAE_log \approx 0.570$

$RMSE_log \approx 0.963$

$R^2 \approx 0.410$

این تفاوت بسیار قابل توجه بود.

یعنی بخش بزرگی از دشواری مدل در metric کلی، از رکوردهای بسیار کم‌قیمت و مشکوک ناشی می‌شد.

با این حال، پروژه در همین نقطه تصمیم گرفت وارد یک فرآیند طولانی‌تر برای پاک‌سازی تک‌تک این موارد یا ساخت مدل‌های جداگانه برای هر نوع ملک نشود.

دلیل این تصمیم حفظ دامنه پروژه و جلوگیری از تبدیل شدن آن به یک پروژه بسیار سنگین و طولانی بود.

6-7 محدودیت مهم پروژه

مهمترین محدودیت باقی مانده کیفیت target بود.

حتی پس از حذف الگوهای واضح قیمت های تکراری، تعدادی از قیمت های بسیار کوچک همچنان در داده وجود داشتند و در test نیز مشاهده شدند.

این مسئله باعث می شود metric هایی که در مقیاس خام قیمت محاسبه می شوند، به شدت تحت تأثیر چند خطای بزرگ قرار بگیرند.

بنابراین برای تفسیر مدل، MAE و RMSE در فضای log-price معیار های مناسب تری برای مقایسه ساختار پیش بینی بودند.

از طرف دیگر، این نتیجه نشان داد که در یک پروژه صنعتی واقعی، مرحله data quality می تواند به اندازه انتخاب الگوریتم اهمیت داشته باشد.

6-8 جمع بندی مدل نهایی

مدل نهایی پروژه یک CatBoost Regressor بود که روی داده های فروش مسکونی Divar آموزش داده شد.

ویژگی های اصلی مدل شامل اطلاعات مربوط به:

شهر

محله

نوع ملک

اندازه ساختمان

اندازه لگاریتمی ساختمان

تعداد اتاق

سال ساخت

سن ملک

اندازه زمین

زمان ثبت آگهی

و برخی شاخص های وجود اطلاعات

بودند.

مدل با target لگاریتمی آموزش داده شد و ارزیابی آن با split زمانی انجام گرفت.

نتیجه نهایی روی 67,327 رکورد test پاکشده:

$$\text{MAE_log} = 0.598$$

$$\text{RMSE_log} = 1.447$$

$$R^2_log = 0.297$$

بود.

این اعداد، همراه با تحلیل خطا و محدودیت‌های داده، در `reports/final_model_report.md` ثبت شدند و به عنوان خروجی نهایی بخش مدل‌سازی پروژه در نظر گرفته شدند.

فصل هفتم

جمع‌بندی کل پروژه

7-1 مسیر طی شده

این پروژه از یک دیتاست خام یک میلیون ردیفی آغاز شد؛ دیتاستی که در نگاه اول فقط مجموعه‌ای از آگهی‌های Divar بود، اما بررسی دقیق‌تر نشان داد که ساختار آن بسیار پیچیده‌تر است. ابتدا مشخص شد که دیتاست چند business مختلف را در کنار یکدیگر قرار داده است. بنابراین یک هدف مشترک و ساده برای همه داده‌ها تعریف نشد و مسیر اصلی مدل‌سازی روی residential-sell متمرکز شد. پس از آن، تحلیل اکتشافی نشان داد که قیمت‌ها به شدت چوله هستند، missingness ساختارمند است، اندازه املاک توزیع بسیار سنگینی دارد و ویژگی‌های جغرافیایی نقش مهمی در قیمت دارند. در مرحله پاک‌سازی، مقادیر نامعتبر قیمت شناسایی شدند، متغیرهای متنی به نمایش قابل استفاده تبدیل شدند و ویژگی‌های جدیدی مانند سن ملک و اندازه لگاریتمی ساخته شدند. سپس با توجه به ساختار زمانی داده، split زمانی طراحی شد تا ارزیابی مدل به شرایط پیش‌بینی آینده نزدیک‌تر باشد. در مرحله مدل‌سازی، ابتدا baseline ساخته شد، سپس Ridge به عنوان مدل خطی و CatBoost به عنوان مدل غیرخطی بررسی شدند. CatBoost عملکرد بهتری نشان داد و سپس با ablation و مقایسه depthها، مدل ساده‌تر و پایدارتر انتخاب شد. در نهایت، مدل روی train و validation آموزش داده شد و روی test زمانی ارزیابی شد.

7-2 مهم‌ترین یافته‌های پروژه

یکی از مهم‌ترین یافته‌ها این بود که موقعیت جغرافیایی نقش بسیار مهمی در پیش‌بینی قیمت دارد city_slug. در مدل نهایی مهم‌ترین ویژگی بود و پس از آن اندازه ملک و ویژگی‌های مرتبط با آن قرار گرفتند. یافته مهم دیگر این بود که ویژگی بیشتر الزاماً به مدل بهتر منجر نمی‌شود. حذف user_type و city_listing_frequency باعث بهبود validation شد و نشان داد که کیفیت و قابلیت تعمیم ویژگی مهم‌تر از تعداد آن‌هاست. همچنین مشخص شد که ساختار قیمت‌گذاری در انواع مختلف ملک یکسان نیست. مدل روی آپارتمان عملکرد بهتری داشت، در حالی که زمین و خانه/ویلا دشوارتر بودند. در نهایت مشخص شد که کیفیت target محدودیت مهمی برای مدل است. قیمت‌های بسیار کوچک و الگوهای تکراری مانند 111111 یا 11111111 نشان دادند که بخشی از داده‌های واقعی آگهی‌ها ممکن است حاوی placeholder یا مقادیر غیرواقعی باشند.

7-3 نتیجه نهایی پروژه

هدف پروژه صرفاً رسیدن به یک مدل با یک metric خاص نبود. مسیر پروژه نشان داد که چگونه می‌توان یک دیتاست واقعی و بزرگ را از مرحله شناخت اولیه، تحلیل اکتشافی و پاک‌سازی، به یک pipeline مدل‌سازی قابل ارزیابی رساند. خروجی نهایی پروژه شامل notebook‌های مراحل مختلف، مستندات تصمیم‌ها، گزارش مدل نهایی و نتایج ارزیابی است.

مهم‌تر از خود مدل، فرآیندی است که پشت آن قرار دارد: تعریف صحیح مسئله، شناخت ساختار داده، جلوگیری از leakage، طراحی split زمانی، کنترل کیفیت target، مقایسه baseline با مدل‌های پیچیده‌تر و در نهایت تحلیل محدودیت‌های مدل.

به این ترتیب، پروژه در نقطه‌ای به پایان رسید که یک مسیر کامل و قابل دفاع از داده خام تا مدل نهایی در اختیار دارد، بدون اینکه برای افزایش جزئی metric‌ها وارد پیچیدگی‌های غیرضروری مانند ساخت مدل جداگانه برای هر نوع ملک، tuning گسترده یا پاک‌سازی بی‌پایان داده شود.