به نام خدا

تکلیف اول- یادگیری ماشین کاربردی

استاد مربوطه: دکتر ناظرفرد

نام: زهرا اخلاقی

شماره دانشجویی: ۴۰۱۱۳۱۰۶۴

ایمیل: zahra.akhlaghi@aut.ac.ir

زمستان ۱۴۰۱

فهرست مطالب

Q1

(a

بله، KNN را می توان برای مسائل multi-class classification استفاده کرد. در KNN، برچسب کلاس یک نمونه آزمایشی توسط برچسب اکثریت k نزدیکترین همسایه آن تعیین می شود. به عنوان مثال، اگر 3 کلاس (A، B و C) داشته باشیم و k=5 را تنظیم کنیم، وقتی یک نمونه جدید را طبقه بندی می کنیم، 5 همسایه نزدیک به نمونه را پیدا می کنیم و شمارش می کنیم که چند تا از آنها متعلق به هر کلاس است. سپس کلاسی که بیشترین تعداد را دارد به نمونه اختصاص داده می شود.

رویکردهای مختلفی برای استفاده از KNN در multi-class classification، وجود دارد، مانند "one-vs-all" و«one-vs-one».

- رویکرد "one-vs-all": برای هر کلاس، یک طبقه‌بندی باینری که آن را از بقیه کلاس ها متمایز می کند، ایجاد و برای هر نمونه آزمایشی، فاصله بین نمونه آزمایشی و تمام نمونه های آموزشی در مجموعه داده را محاسبه و کلاسی را با بیشترین تعداد به عنوان برچسب کلاس برای نمونه آزمایشی پیش بینی میکند.
- رویکرد «one-vs-one»: که در آن یک classification جداگانه KNN برای هر جفت کلاس آموزش داده می‌شود: برای هر جفت کلاس، یک طبقه بندی باینری که یک کلاس را از کلاس دیگر متمایز می کند، ایجاد و برای هر نمونه آزمایشی، فاصله بین نمونه آزمایشی و تمام نمونه های آموزشی در مجموعه داده را محاسبه و برای هر جفت کلاس، کلاسی را که بیشترین تعداد را دارد به عنوان برچسب پیش بینی میکند.

(b

مفهوم curse of dimensionality به این واقعیت اشاره دارد که با افزایش ابعاد (ویژگی ها) در یک مجموعه داده، میزان داده های مورد نیاز برای نشان دادن فضا به صورت تصاعدی رشد می کند. یکی از اصلی‌ترین تأثیرات آن بر عملکرد KNN این است که با افزایش تعداد ابعاد ، فاصله بین نزدیکترین همسایگان معنی کمتری پیدا می‌کند و فاصله بین هر دو نقطه در فضا بسیار شبیه به هم می شود که می تواند به عملکرد ضعیف KNN منجر شود ، زیرا الگوریتم ممکن است بر اساس شناسایی نادرست نزدیکترین همسایگان، برچسب های نادرست را به نقاط داده جدید اختصاص دهد. علاوه بر این ، با افزایش تعداد ابعاد ، میزان داده های مورد نیاز برای دستیابی به عملکرد خوب نیز به صورت نمایی افزایش می یابد. که ممکن است در بسیاری از برنامه های دنیای واقعی امکان پذیر یا عملی نباشد.

(c

الگوریتم KNN به عنوان یک الگوریتم یادگیری تحت نظارت برای طبقه بندی و رگرسیون شناخته می شود. با این حال می تواند در کارهای یادگیری بدون نظارت مورد استفاده قرار گیرد ، جایی که هدف شناسایی الگوهای و ساختار در داده ها بدون اطلاع قبلی از برچسب ها یا اهداف است.

یک روش متداول برای استفاده از KNN در یادگیری بدون نظارت ، استفاده از آن به عنوان یک الگوریتم خوشه بندی

است. در این روش ، از الگوریتم KNN برای گروه بندی نقاط داده به خوشه ها بر اساس شباهت یا نزدیکی آنها به یکدیگر استفاده می شود. این الگوریتم به هیچ دانش قبلی در مورد تعداد خوشه ها یا ویژگی های آنها احتیاج ندارد و آن را به ابزاری مفید برای تجزیه و تحلیل داده های اکتشافی تبدیل می کند.
راه دیگر برای استفاده از KNN در یادگیری بدون نظارت ، استفاده از آن به عنوان الگوریتم تخمین چگالی است. در این روش ، الگوریتم از فاصله بین نقاط داده برای برآورد عملکرد چگالی احتمال داده ها استفاده می کند. این می تواند برای شناسایی مناطقی با چگالی بالا مفید باشد ، که ممکن است مربوط به خوشه ها یا زیر گروه های نقاط داده باشد. به طور کلی ، KNN می تواند ابزاری مفید برای کارهای یادگیری بدون نظارت باشد ، ولی مهم است که K و متریک فاصله مورد استفاده را با دقت انتخاب شود ، زیرا این پارامترها می توانند تأثیر قابل توجهی در عملکرد الگوریتم داشته باشند.

(d

الگوریتم KNN از الگوریتم های طبقه بندی است که در یادگیری ماشین مورد استفاده قرار می گیرند. از دیگر الگوریتم های طبقه بندی میتوان به Naive Bayes, Decision Trees اشاره کرد، در حالی که همه آنها هدف خود را برای حل مشکلات مشابه هدف قرار می دهند اما تفاوتهای اساسی بین آنها وجود دارد.

Approach: الگوریتم KNN غیر پارامتری است که پیش بینی ها را بر اساس فاصله بین نقطه داده جدید و نزدیکترین همسایگان آن در مجموعه آموزش انجام می دهد. این هیچ فرضیه ای در مورد توزیع اساسی داده ها ایجاد نمی کند. در حالی که الگوریتم های طبقه بندی دیگر مانند Bayes Naive یک الگوریتم احتمالی است که پیش بینی ها را بر اساس احتمال نقطه داده جدید متعلق به هر کلاس انجام می دهد و درختان تصمیم گیری یک الگوریتم مبتنی بر قانون هستند که با تقسیم بازگشتی داده ها بر اساس ویژگی ها پیش بینی می کنند تا زمانی که معیار توقف برآورده شود.

Training time: الگوریتم KNN زمان آموزش ندارد زیرا به سادگی داده های آموزش را در حافظه ذخیره می کند، از طرفی الگوریتم های طبقه بندی دیگر، برای ساختن یک مدل نیاز به آموزش داده های دارای برچسب دارند.

Scalability: الگوریتم KNN برای مجموعه داده های بزرگ مقیاس پذیر نیست زیرا نیاز به ذخیره کل مجموعه آموزش در حافظه و محاسبه مسافت بین نقطه داده جدید و هر مثال آموزشی دارد. الگوریتم های طبقه بندی دیگر مانند BayesNaive Bayes, Decision Trees هر دو مقیاس پذیر هستند و می توانند مجموعه داده های بزرگ را به طور مؤثر اداره کنند.

Interpretability: الگوریتم KNN بسیار قابل تفسیر نیست زیرا هیچ بینشی در مورد رابطه بین ویژگی ها و متغیر هدف ارائه نمی دهد. از طرف دیگر ، Naive Bayes, Decision Trees، مدل های واضح و قابل تفسیر را ارائه می دهند که می توانند برای درک روند تصمیم گیری مورد استفاده قرار گیرند.

(e

انتخاب متریک فاصله تأثیر قابل توجهی بر عملکرد الگوریتم KNN دارد. متریک فاصله تعیین می کند که الگوریتم چگونه فاصله بین نقاط داده را در فضای ویژگی محاسبه می کند و این فاصله برای شناسایی k نزدیکترین همسایه به یک نقطه استفاده می شود. انتخاب متریک فاصله باید بر اساس مقیاس بندی داده ها (برخی از معیارهای فاصله، مانند فاصله اقلیدسی، به مقیاس ویژگی ها حساس هستند)، همبستگی ویژگی ها(اگر ویژگی ها بسیار همبسته باشند، فاصله اقلیدسی ممکن است انتخاب بهینه ای نباشد زیرا فرض می کند که ویژگی ها مستقل هستند)و نوع داده ها (معیارهای فاصله متفاوت برای انواع مختلف داده ها مناسب تر است) باشد.

سه معیار فاصله که معمولاً در الگوریتم KNN مورد استفاده قرار می‌گیرند عبارتند از فاصله اقلیدسی، فاصله منهتن و فاصله کسینوس. هر یک از این معیارها ویژگی های متفاوتی دارند که آنها را برای انواع خاصی از داده ها یا مسایل مناسب تر می کند.

فاصله اقلیدسی: فاصله اقلیدسی رایج ترین متریک فاصله مورد استفاده در KNN است. فاصله خط مستقیم بین دو نقطه در فضای ویژگی را محاسبه می کند. فاصله اقلیدسی فرض می کند که ویژگی ها پیوسته هستند و فاصله بین دو نقطه معیاری برای تشابه آنها است. فاصله اقلیدسی می تواند به مقیاس ویژگی ها حساس باشد، بنابراین اغلب با داده های استاندارد یا نرمال استفاده می شود. هنگامی که ویژگی ها پیوسته هستند و داده ها مقیاس یا عادی شده اند و یا فاصله بین دو نقطه معیار مهمی برای تشابه آنها باشد، استفاده میشود.

فاصله منهتن: فاصله بین دو نقطه را با جمع کردن تفاوت مطلق بین مختصات آنها محاسبه می کند. هنگامی که ویژگی ها گسسته یا دسته بندی هستند و فاصله بین دو نقطه در طول یک مسیر شبکه مانند اندازه گیری می شود استفاده میشود.

فاصله کسینوس: زاویه بین دو بردار در فضای ویژگی را اندازه گیری می کند. فاصله کسینوس معمولاً در طبقه‌بندی متن یا پردازش زبان طبیعی استفاده می‌شود، جایی که ویژگی‌ها اغلب به عنوان بردار تعداد کلمات یا فرکانس نشان داده می‌شوند. زمانی که بزرگی مقادیر ویژگی مهم نیست، اما جهت بردارها مهم است هنگامی که ویژگی ها به صورت بردار نشان داده می شوند، مانند طبقه بندی متن یا پردازش زبان طبیعی استفاده میشود.

(f

انتخاب مقدار بهینه K در KNN گام مهمی در ساخت یک مدل دقیق است. انتخاب K بر بایاس-واریانس تأثیر میگذارد، فرآیند انتخاب مقدار بهینه K در KNN شامل ارزیابی عملکرد مدل برای مقادیر مختلف K و انتخاب مقداری است که بهترین نتایج را می دهد. چندین روش برای انتخاب مقدار بهینه K در KNN وجود دارد.

- جستجوی شبکه ای: شامل آموزش و ارزیابی مدل KNN برای طیفی از مقادیر K و انتخاب مدلی است که بهترین عملکرد را ارائه می دهد. پیاده سازی این روش آسان است و برای مجموعه داده های کوچک به خوبی کار می کند، اما برای مجموعه داده های بزرگ می تواند از نظر محاسباتی گران باشد.

- جستجوی تصادفی: جستجوی تصادفی شامل انتخاب تصادفی مقادیر K و ارزیابی مدل KNN برای هر مقدار است. این روش سریعتر از جستجوی شبکه ای است و می تواند برای مجموعه داده های بزرگ به خوبی کار کند، اما ممکن است کل محدوده مقادیر K را کاوش نکند.
- اعتبار سنجی متقابل: اعتبارسنجی متقابل شامل تقسیم داده ها به مجموعه های آموزشی و اعتبار سنجی و ارزیابی مدل KNN برای مقادیر مختلف K در مجموعه اعتبار سنجی است. این روش تخمین دقیق تری از عملکرد مدل ارائه می دهد، اما می تواند از نظر محاسباتی گران باشد و ممکن است برای مجموعه داده های کوچک به خوبی کار نکند.
- دانش دامنه: دانش دامنه را می توان برای انتخاب مقدار بهینه K بر اساس دانش قبلی از مشکل یا داده ها استفاده کرد. به عنوان مثال، اگر مشکل شامل طبقه بندی تصاویر باشد، یک متخصص حوزه ممکن است بداند که محدوده خاصی از مقادیر K نتایج خوبی ارائه می دهد.

در KNN، انتخاب تعداد همسایگان، K، بر بایاس واریانس تاثیر می‌گذارد. مقدار کمتر K منجر به مدلی با بایاس کم اما واریانس بالا می شود، در حالی که مقدار بزرگتر K منجر به مدلی با بایاس زیاد اما واریانس کم می شود. وقتی K کوچک است، مرز تصمیم پیچیده تر است و مدل می تواند تغییرات محلی بیشتری را در داده ها ثبت کند. با این حال، همچنین می تواند مدل را نسبت به نویز در داده ها حساس تر کند و منجر به بیش از حد برازش شود. از طرف دیگر، وقتی K بزرگ باشد، مرز تصمیم هموارتر است و مدل نسبت به نویز در داده ها حساسیت کمتری دارد. این می تواند به تعمیم بهتر به داده های جدید و واریانس کمتر منجر شود، اما همچنین می تواند منجر به عدم تناسب شود. به عبارت دیگر، مدل ممکن است خیلی ساده باشد و الگوی زیربنایی واقعی را در داده ها نشان ندهد. بنابراین، انتخاب K به مشکل خاص و ویژگی های داده ها بستگی دارد. به طور کلی، مقادیر بزرگتر K زمانی که داده ها دارای نویز هستند ترجیح داده می شود، در حالی که مقادیر کوچکتر K زمانی که داده ها نویز کمتری دارند یا مرز تصمیم پیچیده باشد ترجیح داده می شود.

(g

عبارت Locally weighted regression یک اصلاح KNN است که هدف آن بهبود دقت پیش‌بینی‌های رگرسیون است. بر خلاف رگرسیون استاندارد KNN، که مقدار یک نمونه جدید را با در نظر گرفتن K نزدیکترین همسایه‌ها پیش‌بینی می‌کند، LWR میانگین وزنی مقادیر K نزدیک‌ترین همسایه‌هاا محاسبه می‌کند. در LWR وزن بیشتری به نزدیک ترین همسایگان و وزن کمتری به همسایگان دورتر اختصاص میدهد، به طوری که پیش بینی بیشتر بر اساس مقادیر همسایه هایی است که مشابه نمونه جدید هستند.

مزایای استفاده از LWR در KNN عبارتند از: LWR

- می‌تواند روابط پیچیده‌تر و غیرخطی‌تری را بین ویژگی‌ها و متغیر هدف ثبت کند، زیرا وزن بیشتری به همسایه‌های شبیه تر می‌دهد.
- استفاده از LWR می‌تواند داده‌های نویز دار و پرت را بهتر از رگرسیون KNN استاندارد مدیریت کند، زیرا می‌تواند تأثیر اقلام پرت را کاهش دهد.

معایب استفاده از LWR در KNN عبارتند از:

- استفاده از LWR می تواند از نظر محاسباتی گران باشد، زیرا نیاز به محاسبه فاصله بین نمونه جدید و تمام نمونه های آموزشی برای هر پیش بینی دارد.
- استفاده از LWR میتواند با افزایش ابعاد مشکل داشته باشد، زیرا عملکرد هسته در فضاهای با ابعاد بالا کمتر اطلاعاتی می دهد.

برخی از کاربردهای LWR عبارتند از: پیش‌بینی ۱- سری‌های زمانی(LWR می‌تواند برای پیش‌بینی مقادیر آینده یک سری زمانی بر اساس مقادیر گذشته آن، با استفاده از یک تابع که وزن بیشتری به مشاهدات اخیر می‌دهد، استفاده شود)، ۲- پردازش تصویر(LWR می‌تواند برای درون‌یابی پیکسل‌های از دست رفته در یک تصویر، با استفاده از یک تابع که وزن بیشتری به پیکسل‌های مجاور با رنگ یا شدت مشابه می‌دهد، استفاده شود)، ۳- رباتیک، ۴- مدل‌سازی مالی(LWR می‌تواند برای مدل‌سازی قیمت سهام یک سبد، با استفاده از یک تابع که وزن بیشتری به سهام با حرکات قیمت مشابه می‌دهد، استفاده شود) و ۵-پردازش زبان طبیعی اشاره کرد، البته استفاده از LWR به موارد فوق محدود نمیشود.

(h

$$d = \sum_{i=1}^{n} |x_i - y_i|$$

برای test instance 7 فاصله‌ی آن را با تمام نمونه‌ها محاسبه کرده و بر اساس ۳ نزدیک‌ترین همسایه کلاس آن مشخص می‌شود:

برای محاسبه فاصله داریم:

Instance 7
Feature 1 = 3
Feature 2 = 4

Instance	distance
1	$\|3-3\| + \|4-4\| = 1$
2	$\|3-3\| + \|5-4\| = 1$
3	$\|4-3\| + \|6-4\| = 3$
4	$\|5-3\| + \|7-4\| = 5$
5	$\|4-3\| + \|8-4\| = 7$
6	$\|7-3\| + \|9-4\| = 9$

۳ نزدیک‌ترین همسایه

براساس کلاس ۳ نزدیک‌ترین همسایه Instance 7 ، کلاس آن برای A می‌باشد

برای Instance 8 داریم:

Instance 8
Feature 1 = 6
Feature 2 = 5

Instance	distance
1	$\|2-6\| + \|4-5\| = 5$
2	$\|3-6\| + \|5-5\| = 3$
3	$\|4-6\| + \|6-5\| = 3$
4	$\|5-6\| + \|7-5\| = 3$
5	$\|6-6\| + \|8-5\| = 3$
6	$\|7-6\| + \|9-5\| = 5$

با توجه به اینکه ۴ نمونه‌داده نامتعادل هستند و ۲ نمونه متعلق به کلاس A و ۲ نمونه دیگر متعلق به کلاس B هستند. به این معنی است که الگوریتم KNN با $k=3$ نمی‌تواند کلاس را مشخص کند و راه حل بهتر افزایش مقدار k تا دستیابی به نزدیک‌ترین همسایه‌ها بیشتر و یا استفاده از وزن نزدیکی همسایه‌ها، برای مشخص کردن این کلاس می‌باشد.

(i

در یادگیری ماشین، مدل یک نمایش ریاضی از یک سیستم یا فرآیندی است که ورودی ها را به خروجی ها ترسیم می کند. پارامتر مدل متغیری است که از داده ها آموخته می شود و رفتار مدل را تعریف می کند. به عنوان مثال، در رگرسیون خطی، پارامتر مدل، شیب خطی است که بهترین تناسب را با داده ها دارد که از داده ها در طول آموزش یاد می شود. از سوی دیگر، فراپارامتر، توسط کاربر بر اساس دانش قبلی یا آزمون و خطا برای کنترل رفتار الگوریتم یادگیری تنظیم می شوند. نمونه‌هایی از فراپارامترهای الگوریتم یادگیری شامل تعداد لایه‌های پنهان در یک شبکه عصبی، نرخ یادگیری یک الگوریتم بهینه‌سازی یا مقدار K در KNN است. پارامترهای مدل برای یک مجموعه داده

آموزشی خاص هستند و می توانند از یک مجموعه داده به مجموعه دیگر تغییر کنند، در حالی که هایپرپارامترها مستقل از مجموعه داده هستند و در مجموعه داده های مختلف یکسان می مانند.

(j

از چالش های یادگیری ماشین میتوان به موارد زیر اشاره کرد:

کیفیت و کمیت داده ها: الگوریتم های یادگیری ماشینی به مقادیر زیادی داده با کیفیت بالا برای یادگیری الگو و پیش بینی دقیق نیاز دارند. با این حال، داده ها می توانند دارای نویز، ناقص، مغرضانه یا پرت باشند که می تواند بر عملکرد مدل تأثیر بگذارد.

اضافه برازش و عدم تناسب: برازش بیش از حد زمانی اتفاق می‌افتد که یک مدل داده‌های آموزشی را خیلی خوب یاد می‌گیرد و نمی‌تواند به داده‌های جدید تعمیم یابد، در حالی که عدم برازش زمانی رخ می‌دهد که یک مدل برای دریافت پیچیدگی داده‌ها بسیار ساده باشد. یافتن تعادل مناسب از حد یک چالش رایج در یادگیری ماشین است.

انتخاب الگوریتم‌ها و فراپارامترهای مناسب: الگوریتم‌ها و فراپارامترهای مختلف می‌توانند نتایج قابل‌توجهی متفاوتی را برای یک مجموعه داده ایجاد کنند و انتخاب مناسب نیاز به بررسی دقیق مسئله، داده‌ها و معیارهای عملکرد دارد.

تفسیرپذیری و توضیح پذیری: از آنجایی که یادگیری ماشین به طور فزاینده ای در برنامه هایی استفاده می شود که بر زندگی افراد تأثیر می گذارد، تقاضای فزاینده ای برای مدل هایی وجود دارد که شفاف، قابل تفسیر هستند و می توانند توضیحاتی را برای پیش بینی های آنها ارائه دهند.

(k

اگر یک مدل در داده‌های آموزشی عملکرد عالی داشته باشد اما به نمونه‌های جدید تعمیم ضعیفی داشته باشد، احتمالاً مدل بیش از حد با داده‌های آموزشی مطابقت دارد. تطبیق بیش از حد زمانی اتفاق می‌افتد که یک مدل ویژگی‌های خاص داده‌های آموزشی را خیلی خوب یاد بگیرد و نتواند به داده‌های جدیدی که قبلاً دیده نشده است تعمیم دهد. هنگامی که یک مدل بیش از حد برازش می‌کند، نویز و نقاط پرت داده های آموزشی را به خاطر می‌سپارد. در نتیجه، مدل بیش از حد پیچیده و در نمونه های جدید و دیده نشده عملکرد خوبی ندارد، این مشکل می تواند ناشی از زمان زیاد آموزش نیز باشد. راه حل های ممکن برای رفع این مشکل عبارتند از:

منظم‌سازی: منظم‌سازی تکنیکی است که شامل اضافه کردن یک عبارت جریمه به تابع ضرر مدل برای جلوگیری از وزن‌های بزرگ یا مدل‌های پیچیده است.

اعتبار سنجی: روشی است که شامل تقسیم داده ها به چند فولد و آموزش مدل بر روی زیر مجموعه های مختلف داده و در عین حال اعتبارسنجی عملکرد در قسمت باقی مانده است.

توقف زودهنگام: توقف زودهنگام تکنیکی است که شامل نظارت بر عملکرد مدل در یک مجموعه اعتبار سنجی جداگانه در طول آموزش و توقف آموزش زمانی که عملکرد مجموعه اعتبارسنجی شروع به بدتر شدن می کند، می شود.

(l

تنظیم هایپر پارامترها با استفاده از مجموعه تست می تواند منجر به تخمین های عملکرد بیش از حد برازش و مغرضانه شود. هدف از تست، ارزیابی عملکرد نهایی مدل پس از تنظیم تمام فرا پارامترها است، نه تنظیم فرا پارامترها.

اگر فرا پارامترها روی مجموعه تست تنظیم شوند، ممکن است مدل در نهایت مجموعه تست را بیش از حد برازش دهد و روی داده‌های جدید و دیده نشده ضعیف عمل کند، علت آن این است که مدل به‌طور خاص برای عملکرد خوب در مجموعه آزمایشی به‌جای تعمیم به نمونه‌های جدید بهینه شده است. استفاده از مجموعه تست برای تنظیم فرا پارامتر می‌تواند تخمین‌های عملکرد را سوگیری کند و آنها را بیش از حد خوش بینانه کند. مجموعه آزمایشی برای هدایت انتخاب فرا پارامترهای استفاده شده است و نمی تواند یک تخمین بی طرفانه از عملکرد واقعی مدل در داده های جدید ارائه دهد. برای جلوگیری از این مسائل، مهم است که داده ها را به سه مجموعه جداگانه تقسیم کنیم: یک مجموعه آموزشی برای آموزش مدل، یک مجموعه اعتبار سنجی برای تنظیم فرا پارامتر، و یک مجموعه تست برای ارزیابی نهایی.

(m

نشان‌دهنده آن است که مدل بیش از حد با داده‌های آموزشی مطابقت دارد. برازش بیش از حد زمانی اتفاق می‌افتد که مدل بیش از حد پیچیده باشد و به جای الگوهای زیربنایی، نویز را در داده‌های آموزشی تطبیق دهد. این می تواند باعث شود که مدل در داده های آموزشی عملکرد خوبی داشته باشد، اما در داده های جدید و دیده نشده ضعیف عمل کند، که با خطای اعتبار سنجی نشان داده می شود.

در رگرسیون چند جمله ای، پیچیدگی مدل با درجه چند جمله ای مورد استفاده برای برازش داده ها تعیین می شود. اگر درجه چند جمله ای خیلی زیاد باشد، مدل پارامترهای زیادی خواهد داشت و شروع به تطبیق نویز در داده ها می کند. این باعث می شود که خطای آموزش به سطوح بسیار پایین کاهش یابد، در حالی که خطای اعتبارسنجی به دلیل نویز در مجموعه اعتبار سنجی بالاتر باقی می ماند.

برای حل این شکل می توانیم، ۱- پیچیدگی مدل را با کاهش درجه چند جمله ای کاهش دهیم، ۲- استفاده از تکنیک های منظم سازی برای محدود کردن مقادیر پارامترهای مدل و ۳- از افزایش اندازه داده های آموزشی و استفاده از تکنیک های تقویت داده ها برای تولید نمونه های آموزشی بیشتر استفاده کنیم.

(n

Ridge Regression instead of plain Linear Regression

- در رگرسیون خطی، زمانی که تعداد ویژگی‌ها نسبت به تعداد نمونه‌های آموزشی زیاد باشد، مدل می‌تواند به راحتی داده‌های آموزشی را اضافه کند. رگرسیون ریج یک عبارت جریمه به تابع هزینه اضافه می کند که تخمین پارامترها را به سمت صفر کوچک می کند و کمتر مستعد بیش از حد برازش است.
- با جلوگیری از برازش بیش از حد، رگرسیون ریج می تواند عملکرد تعمیم مدل را بر روی داده های جدید و دیده نشده بهبود بخشد. این کار زمانی که مدل برای دنیای واقعی اعمال شود، جایی که داده های ورودی ممکن است حاوی نویز یا سایر منابع تغییرپذیری باشند، مفید است.

- در رگرسیون خطی، تغییرات کوچک در داده های ورودی یا نویز در داده ها می تواند منجر به تغییرات بزرگ در پارامترهای مدل شود که منجر به راه حل های ناپایدار می شود. در حالی که رگرسیون ریج به تثبیت پارامترهای مدل کمک می کند و منجر به پیش بینی های قوی تر و قابل اعتمادتر می شود.

رگرسیون ریج می تواند در هنگام برخورد با مجموعه داده هایی با تعداد زیادی ویژگی، داده های آموزشی محدود، زمانی که نگرانی در مورد تطبیق بیش از حد وجود دارد و یا زمانی که هدف دستیابی به عملکرد تعمیم بهتر است، مفید باشد.

Lasso instead of Ridge Regression

- رگرسیون Lasso می‌تواند با صفر کردن برخی از برآوردهای پارامترها، انتخاب خودکار ویژگی را انجام دهد. یعنی بسیاری از پارامترها دقیقاً برابر با صفر هستند و در نتیجه مدلی ساده‌تر و قابل تفسیرتر ایجاد می‌شود. در مقابل، رگرسیون ریج معمولاً تمام پارامترها را به سمت صفر کوچک می کند بدون اینکه لزوماً هیچ یک از آنها را صفر کند.
- رگرسیون Lasso می‌تواند مدل ساده‌تری با متغیرهای کمتر تولید کند، که تفسیر آن آسان‌تر از مدلی با متغیرهای زیاد است.
- در برخی موارد، Lasso می‌تواند منجر به بهبود دقت پیش‌بینی در مقایسه با رگرسیون Ridge شود، به‌ویژه زمانی که ویژگی‌های ورودی بسیار همبسته هستند. این به این دلیل است که Lasso می تواند به طور موثر یکی از متغیرهای همبسته را انتخاب کند و بقیه را به صفر برساند.

رگرسیون Lasso زمانی می‌تواند مفید باشد که نیاز به انتخاب ویژگی یا مدلی ساده‌تر و قابل تفسیرتر باشد، یا زمانی که ویژگی‌های ورودی بسیار همبسته هستند. با این حال، ممکن است برای مجموعه‌های داده که در آن همه ویژگی‌های ورودی مهم هستند یا زمانی که تعداد ویژگی‌ها بسیار بیشتر از تعداد نمونه‌ها است، مناسب نباشد.

Elastic Net instead of Lasso

- در Lasso با مجموعه داده‌هایی که ویژگی‌های همبسته زیادی وجود دارد، مبارزه کند، زیرا تمایل دارد تنها یک ویژگی را از بین ویژگی‌های مرتبط انتخاب کند و بقیه را روی صفر قرار دهد. Elastic Net می تواند این محدودیت را با افزودن عبارت جریمه L2، که مدل را تشویق می کند تا چندین ویژگی مرتبط را انتخاب کند، غلبه کند.
- Elastic Net می‌تواند عملکرد بهتری نسبت به Lasso در داده هایی با ابعاد زیاد داشته باشد. در چنین مواردی، Lasso ممکن است به طور تصادفی برخی از ویژگی ها را انتخاب کند، در حالی که Elastic Net می تواند با حفظ همه ویژگی های مرتبط و کاهش ضرایب آنها، عملکرد بهتری داشته باشد.
- شبکه Elastic Net انعطاف پذیری بیشتری را در تنظیم قدرت منظم سازی امکان پذیر می کند.

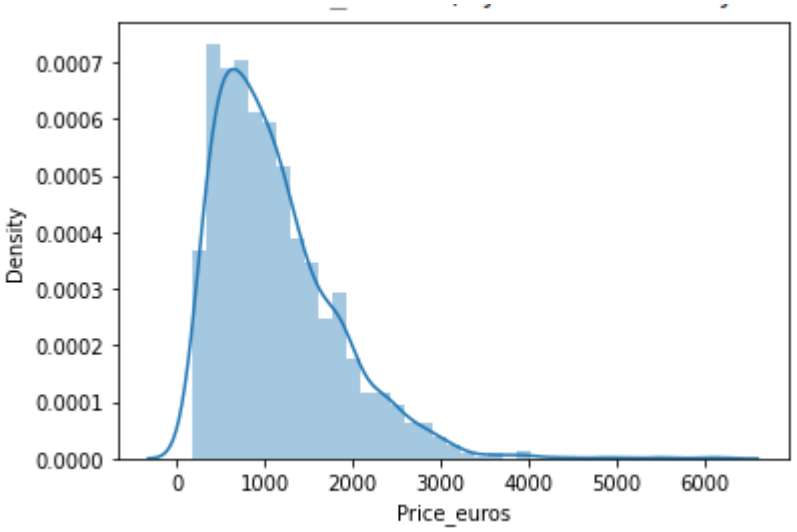
Elastic Net می تواند در شرایطی مفید باشد که Lasso با تعداد ویژگی ها یا همبستگی بین ویژگی ها محدود است. همچنین Elastic Net انعطاف پذیری بیشتری را فراهم می کند و ممکن است در مجموعه داده های با ابعاد بالا عملکرد بهتری داشته باشد. با این حال، ممکن است، زمانی که همه ویژگی‌های مجموعه داده بسیار مرتبط هستند بهترین انتخاب نباشد.

Q2

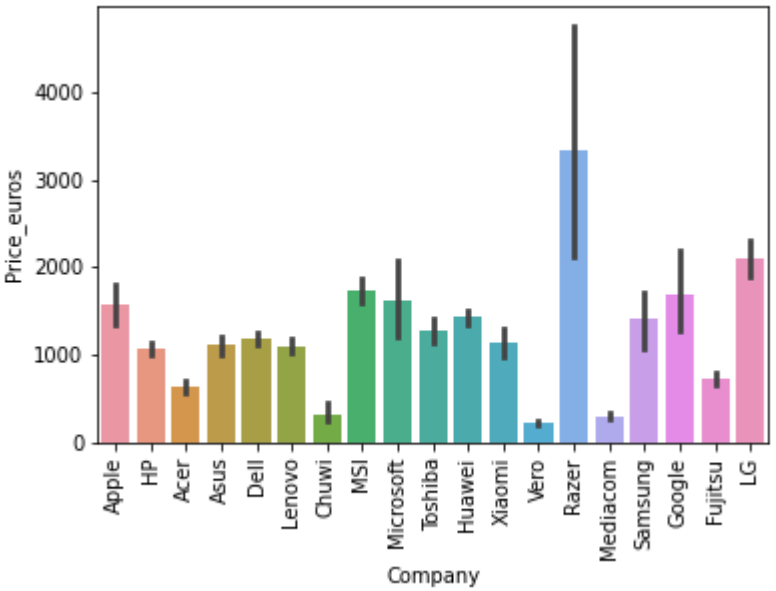
(a

	laptop_ID	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price_euros
829	838	Acer	Swift 3	Notebook	14.0	IPS Panel Full HD 1920x1080	Intel Core i3 7100U 2.4GHz	8GB	128GB SSD	Intel HD Graphics 620	Windows 10	1.5kg	619.0
472	479	Google	Pixelbook (Core	Ultrabook	12.3	Touchscreen 2400x1600	Intel Core i5 7Y57 1.2GHz	8GB	128GB SSD	Intel HD Graphics 615	Chrome OS	1.1kg	1275.0
244	249	Lenovo	Yoga 910-13IKB	2 in 1 Convertible	13.9	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 7500U 2.7GHz	8GB	256GB SSD	Intel HD Graphics 620	Windows 10	1.38kg	1079.0
639	647	Dell	XPS 15	Notebook	15.6	4K Ultra HD / Touchscreen 3840x2160	Intel Core i7 7700HQ 2.8GHz	16GB	1TB SSD	Nvidia GeForce GTX 1050	Windows 10	2.06kg	2399.0
76	78	Lenovo	IdeaPad 320-15IKBN	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	2TB HDD	Intel HD Graphics 620	No OS	2.2kg	519.0
1134	1149	Acer	Aspire E5-576G	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	4GB	1TB HDD	Nvidia GeForce 940MX	Windows 10	2.23kg	616.0
1228	1246	Lenovo	IdeaPad Y700-15ISK	Gaming	15.6	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6700HQ 2.6GHz	16GB	128GB SSD + 1TB HDD	Nvidia GeForce GTX 960M	Windows 10	2.6kg	1029.0
263	268	Dell	Vostro 3568	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	4GB	1TB HDD	Intel HD Graphics 620	Windows 10	2.18kg	657.0
912	925	Lenovo	IdeaPad 510-15IKB	Notebook	15.6	Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	6GB	256GB SSD	Nvidia GeForce 940MX	Windows 10	2.2kg	789.0
385	391	Lenovo	Thinkpad X1	Ultrabook	14.0	IPS Panel 2560x1440	Intel Core i7 7500U 2.7GHz	8GB	512GB SSD	Intel HD Graphics 620	Windows 10	1.13kg	2282.0

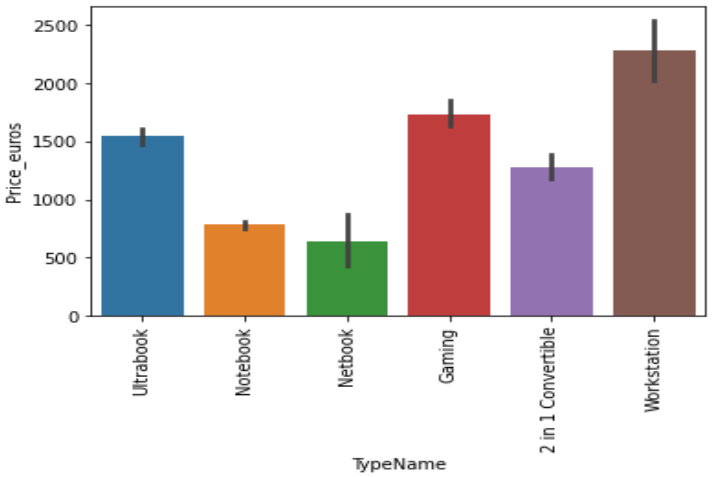
(b



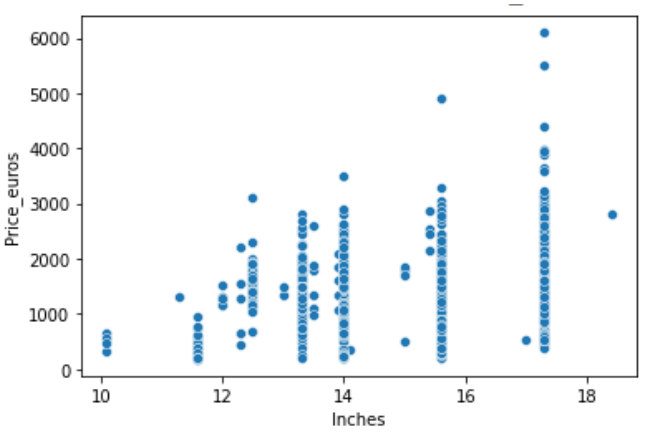
تفسیر: نمودار توزیع قیمت به سمت چپ تمایل دارد که نشان دهنده این است، لپ تاپ هایی با قیمت پایین بیشتر از مارک دار خرید و فروش می شوند.



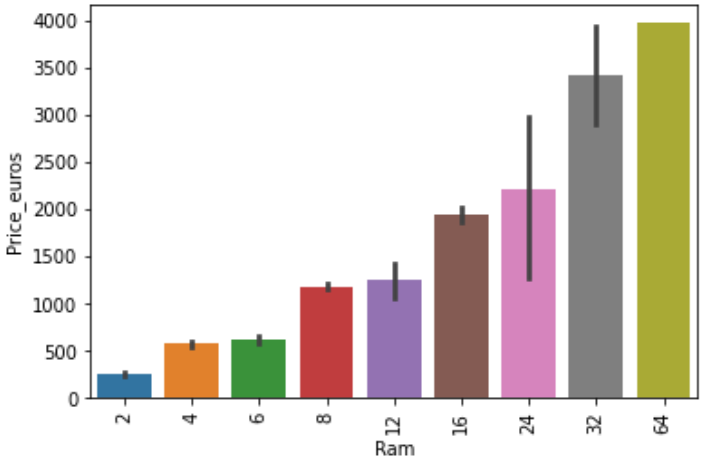
تفسیر: نمودار بالا نشان دهنده تاثیر نام تجاری بر قیمت لپ تاپ میباشد، که لپ تاپ های Razer, Apple, LG, Microsoft, Google, MSI گران هستند.



تفسیر: نمودار بالا تاثیر نوع لپ تاپ بر قیمت آن را نمایش میدهد، که workstation دارای بیشترین قیمت و notebook دارای کمترین قیمت است، بنابراین طبیعی است که تمایل انسان ها بیشتر در خرید notebook باشد.



تفسیر: نمودار بالا تاثیر اندازه بر قیمت لپ تاپ را نشان میدهد، که میتوان نتیجه گرفت میان قیمت و اندازه رابطه وجود دارد اما رابطه قوی وجود ندارد.



تفسیر: نمودار بالا تاثیر Ram بر قیمت لپ تاپ را نشان میدهد، که با افزایش آن قیمت لپ تاپ افزایش میابد.

(c

مهندسی ویژگی فرآیندی برای تبدیل داده های خام به اطلاعات معنی دار است. ستون‌هایی در جدول دارای نویز هستند و یا میتوان اطلاعات مفیدی از آنها به دست آورد و یا طبقه بندی کرد، بنابراین باید مهندسی ویژگی را روی آنها اعمال کرد.

برای استخراج دقیق تر و فهم بهتر اطلاعات، ستون های Weight , Ram که در آنها واحد اطلاعات نیز ذخیره شده است (GB,kg) را حذف و نوع آنها را به ترتیب به integer و float تغییر دادم.

ستون Screen Resolution حاوی اطلاعات زیادی است، با دیدن مقادیر آن میتوان متوجه شد که حاوی اطلاعات مربوط به وجود یک پنل IPS، وجود صفحه لمسی لپ‌تاپ و وضوح صفحه نمایش محور X و محور Y می باشد.

بنابراین،آن را میتوانیم به 3 ستون جدید در مجموعه داده استخراج کرد.

Touchscreen	Ips	ppi
0	0	100.454670
0	1	141.211998
0	0	141.211998
0	0	111.935204
0	0	100.454670

با مشاهده ستون CPU و مقادیر آن، می توان متوجه شد که عمده داده ها را میتوان در ۵ دسته Intel Core i7, Intel Core i5 ,Intel Core i3, Other Intel Processor, AMD Processor قرار دارند و ستون CpuBrand براین اساس اضافه شده که حاوی این ۵ دسته بر اساس مقادیر در ستوان cpu میباشد.

ستون memory، دارای اطلاعات پرت می باشد که بدون تحلیل و بررسی اطلاعات مفیدی را در اختیار ما نمیگذارد، میتوان آن را به ۴ ستون hybrid، Flash storage ،HHD, SSD تقسیم کرد، که بیانگر عنوان حافظه ها میباشد و در هر ستون مقدار آن ذخیره شده است(برای محاسبه آنها ستون هایی برای کمک اضافه و سپس حذف میشوند).

از ستون Gpu برند آن را در Gpu brand ذخیره و ستون Gpu را حذف کردم.

با مشاهده مقادیر ستون سیستم عامل متوجه میشویم که دسته های زیادی از سیستم عامل ها وجود دارد. با مشاهده مقدار تکرار هر کدام آن را با سه دسته windows, Mac, others ذخیره کردم.

با مشاهده تاثیر هر ستون بر روی قیمت لپ تاپ، فیلد هایی که تاثیر کمی در قیمت لپ تاپ داشتند را حذف کردم. جدول نهایی به صورت زیر است:

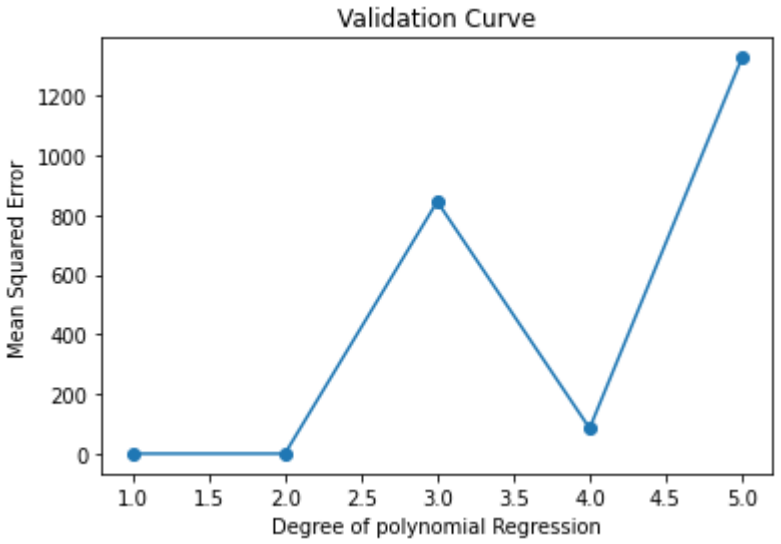
	Company	TypeName	Ram	Weight	Price_euros	Touchscreen	Ips	ppi	CpuBrand	HDD	SSD	GpuBrand	os
1200	Dell	Notebook	4	2.30	499.0	0	0	100.454670	Intel Core i3	0	128	AMD	Windows
218	Dell	Ultrabook	8	1.60	1149.0	0	0	157.350512	Intel Core i5	0	256	Intel	Windows
546	Lenovo	Notebook	4	1.87	785.0	0	0	157.350512	Intel Core i5	500	0	Intel	Windows
845	Dell	Notebook	4	2.18	739.0	0	0	141.211998	Intel Core i7	0	256	AMD	Windows
1060	HP	Notebook	8	2.04	1070.0	0	0	141.211998	Intel Core i7	0	256	Intel	Windows

(d

ستون های SSD , HDD, ppi, Weight, Ram برای scaling و ستون های os, GpuBrand, CpuBrand, TypeName , Company برای encoding اعمال شده است

داده ها به سه دسته test , validation , train تقسیم شده اند ولی با توجه به اینکه در بررسی مدل از kfold استفاده شده در آن نیازی به جداسازی داده validation نیست.

(e



تفسیر: نمودار بالا رابطه افزایش درجه و mse را نشان میدهد. کمترین مقدار mse مربوط به درجه ۱ و ۲ می باشد , برای درجه ۴ به طور ناگهانی کاهش یافته است. بهترین درجه انتخاب شده ۱ میباشد.

(f

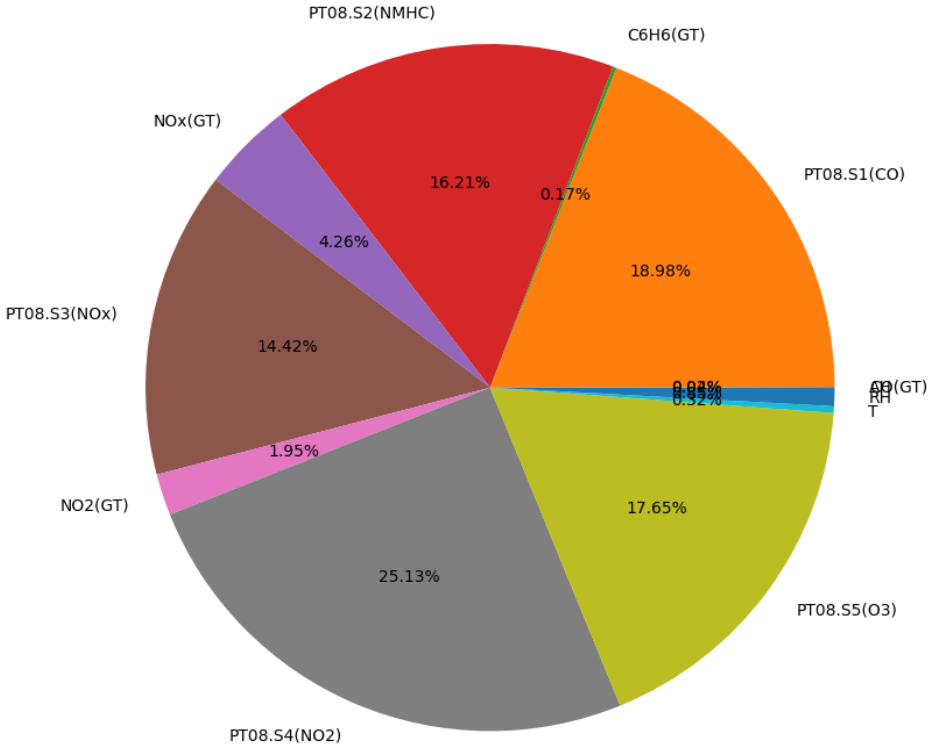
```
MSE on test data: 0.061380194187830764
R2 on test data: 0.8268167810990916
```

(g

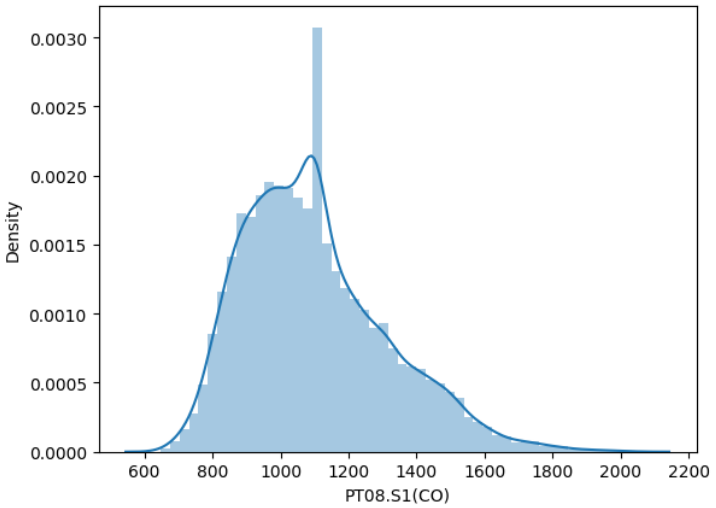
مهمترین ویژگی انتخاب درجه مناسب برای چند جمله ای است. زیرا درجه چند جمله ای به طور مستقیم بر پیچیدگی مدل و توانایی آن در برازش داده های آموزشی تأثیر می گذارد. اگر درجه چند جمله‌ای خیلی کم باشد، مدل ممکن است با داده‌ها کمتر مطابقت داشته باشد و بایاس بالایی داشته باشد، در حالی که اگر درجه خیلی زیاد باشد، ممکن است مدل بیش از حد برازش داده و واریانس بالایی داشته باشد. بنابراین، انتخاب درجه بهینه چند جمله ای که برازش و واریانس را متعادل می کند برای دستیابی به عملکرد خوب مدل بسیار مهم است.

Q3

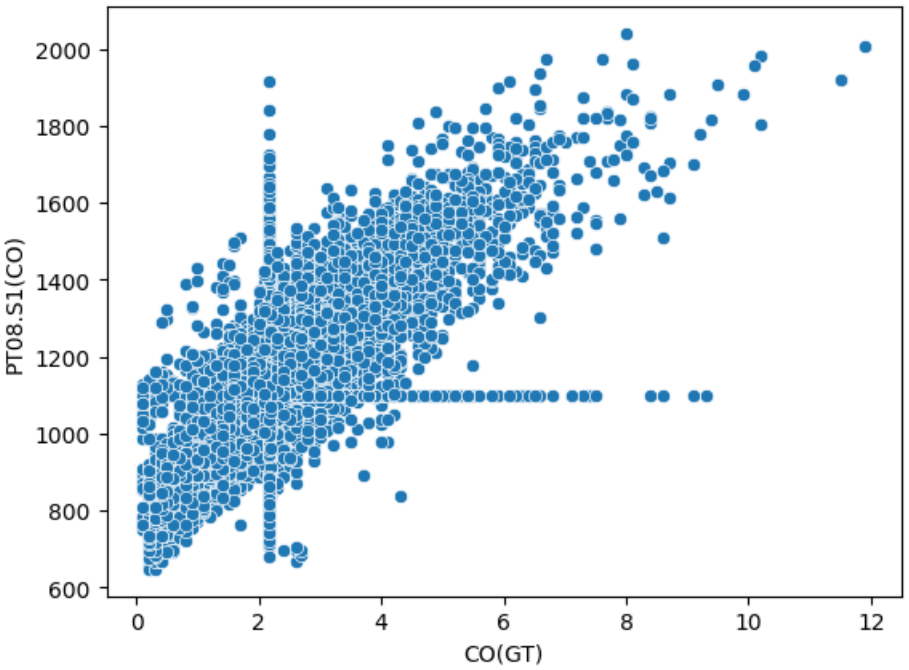
(a



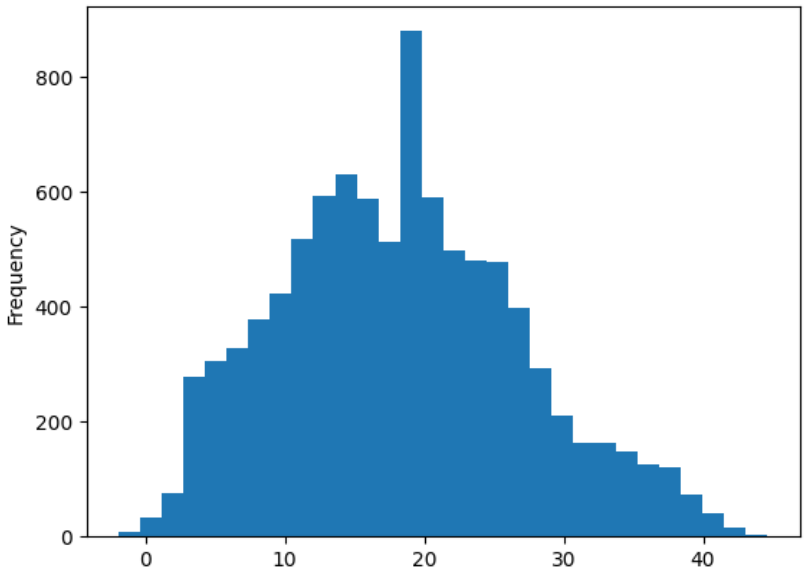
تفسیر: نمودار بالا میانگین آلاینده های مختلف را نشان میدهد، که PT08.S4(NO2)دارای بیشترین مقدار میباشد.



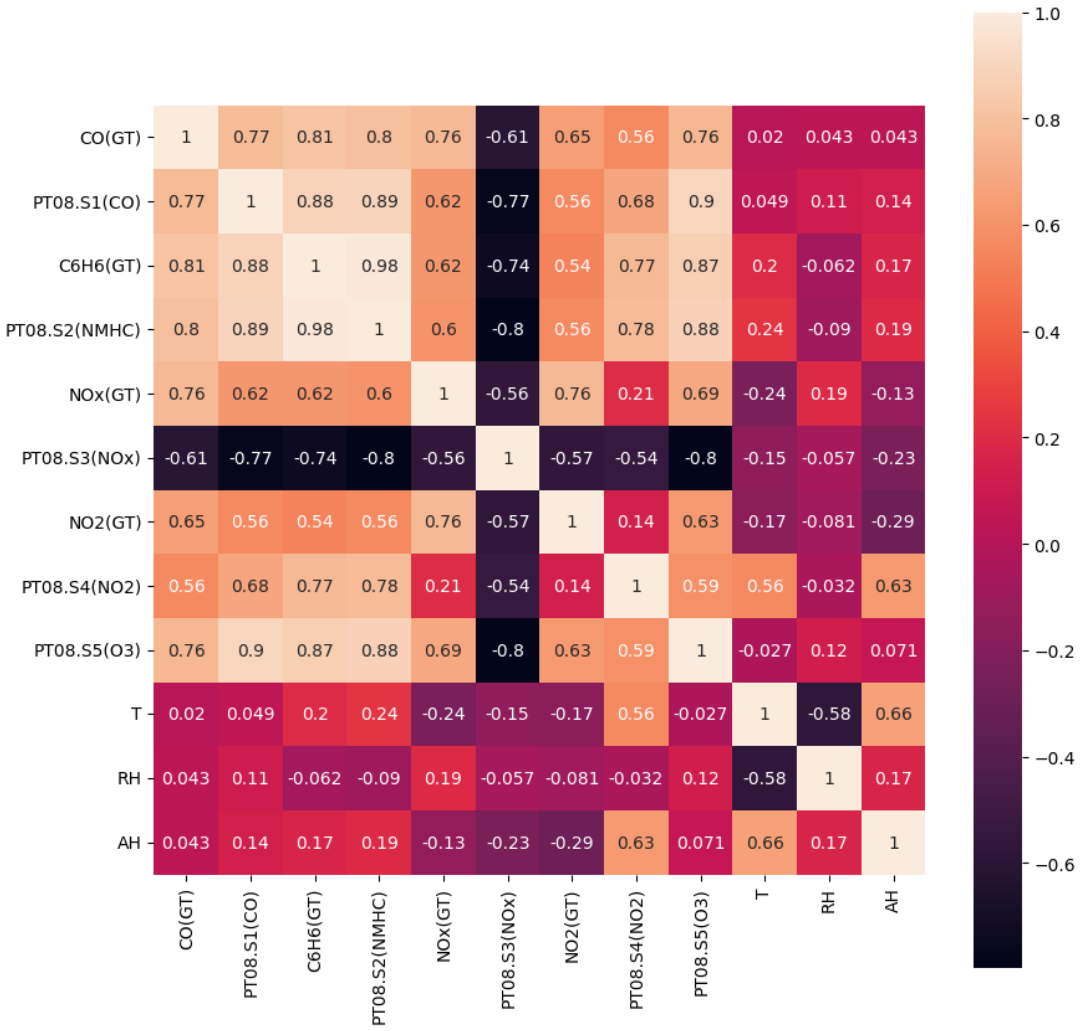
تفسیر: شکل بالا نمودار توزیع گاز PT08.s1(CO) را نمایش میدهد،نمودار به سمت چپ تمایل دارد به این معنی که در بیشتر روز های گزارش شده، مقدار PT08.s1(CO) کمتر از میانگین است.



تفسیر: نمودار بالا نشان دهنده ارتباط دو گاز CO GT و PT08.S1 CO را نشان میدهد، تقریبا این دو داده با یکدیگر ارتباط مستقیم دارند و با افزایش یکی دیگری نیز افزایش میابد.



تفسیر: نمودار بالا هیستوگرام دما را نمایش میدهد، بیشترین تعداد دما حدود ۲۰ درجه بوده است.



تفسیر: نمودار بالا تاثیر فیلد های مختلف را بر یکدیگر نمایش میدهد.

(b

با مشاهده مقادیر ستونNMHC(GT) مشخص است که ۸۴۴۳ از ۹۳۵۷ دارای مقدار ۲۰۰-است. بنابراین این ستون مورد نیاز نیست و باید حذف می شود.

در داده ها مقدار زیادی ۲۰۰- وجود دارد که به نظر میرسد خطاست راه حل پیشنهادی جایگزینی آنها با null و سپس مقدار تخمینی میباشد.

در جدول مقادیر Date , Time با فرمت object ذخیره شده اند و اطلاعات مفیدی از آنها در ارتباط با تاثیر آنها به روی گار نمی توان استخراج کرد، نوع این دو ستون را به float تغییر دادم.

تاثیر هر یک از ستون ها بر PT08.S1 CO را بررسی کردم.

(d

```
{'Ridge': {'alpha': 0.001}, 'Lasso': {'alpha': 0.001}, 'Elastic Net': {'alpha': 0.001, 'l1_ratio': 0.1}}
{'Ridge': 0.8803291540701933, 'Lasso': 0.8652401192958463, 'Elastic Net': 0.8772320081747195}
```

(e

```
Ridge Regression: MSE= 0.004 ,R_Squared= 0.879
```

```
Lasso Regression: MSE= 0.005 ,R_Squared= 0.863
```

```
Elastic Net Regression: MSE= 0.004 ,R_Squared= 0.876
```

(f

برای مقایسه عملکرد سه مدل (Ridge، Lasso و Elastic Net)، می‌توانیم از معیارهای MSE وR2 روی داده‌های آزمون که در قسمت قبل انجام شده، استفاده کنیم. به طور کلی، هرچه MSE کمتر و R2 بالاتر باشد، مدل بهتر است. دو الگوریتم Ridge , Elastic Net Regression دارای mse یکسانی میباشند و بین این الگوریتم ها Ridge با اختلاف کمی دارای بهترین عملکرد میباشد. زیرا MSE کمتر و R2 بالاتر دارد.

Q4

(a

```
test: X: (540, 64)  Y: (540,)
train: X: (1257, 64)  Y: (1257,)
```

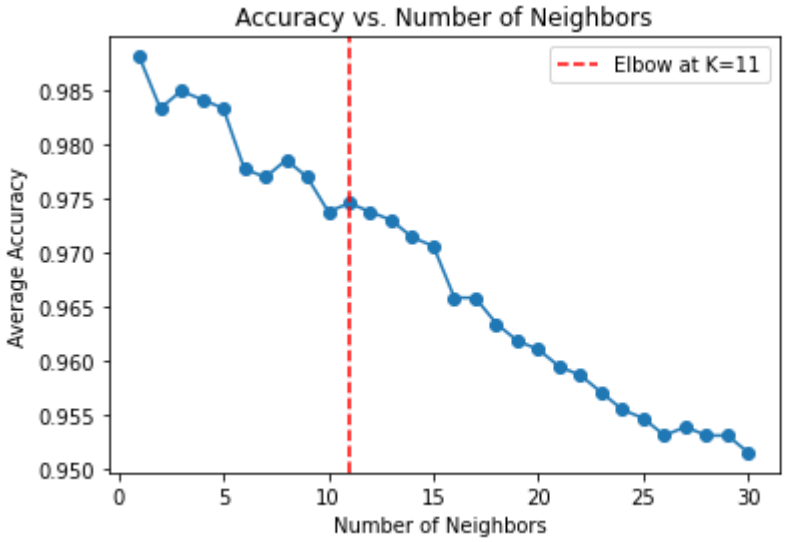
(c

	precision	recall	f1-score	support
0	1.00	1.00	1.00	53
1	1.00	1.00	1.00	50
2	1.00	1.00	1.00	47
3	0.98	1.00	0.99	54
4	0.98	1.00	0.99	60
5	0.98	0.98	0.98	66
6	1.00	1.00	1.00	53
7	1.00	1.00	1.00	55
8	1.00	1.00	1.00	43
9	0.98	0.95	0.97	59
accuracy			0.99	540
macro avg	0.99	0.99	0.99	540
weighted avg	0.99	0.99	0.99	540

(d

```
Average Accuracy: 0.98 (+/- 0.01)
Average Precision: 0.98 (+/- 0.01)
Average Recall: 0.98 (+/- 0.01)
Average F1-score: 0.98 (+/- 0.01)
```

(e



در این حالت، مقدار بهینه K حدود K=11 به نظر می رسد، جایی که دقت شروع به کاهش می کند. این به این معنی است که استفاده از K=11 احتمالاً بالاترین دقت را برای این طبقه‌بندی‌کننده خواهد داشت.

Q5

(a

```
test: X: (30, 4)  Y: (30,)
train: X: (120, 4)  Y: (120,)
```

(c

```
K = 1, accuracy = 1.00
K = 2, accuracy = 1.00
K = 3, accuracy = 1.00
K = 4, accuracy = 1.00
K = 5, accuracy = 1.00
K = 6, accuracy = 1.00
K = 7, accuracy = 1.00
K = 8, accuracy = 1.00
K = 9, accuracy = 1.00
K = 10, accuracy = 1.00
K = 11, accuracy = 1.00
K = 12, accuracy = 1.00
K = 13, accuracy = 1.00
K = 14, accuracy = 1.00
K = 15, accuracy = 1.00
K = 16, accuracy = 1.00
K = 17, accuracy = 1.00
K = 18, accuracy = 1.00
K = 19, accuracy = 1.00
K = 20, accuracy = 1.00

Optimal value of K = 1 with accuracy = 1.00
```