به نام کیمیاگر عالم

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

یادگیری ماشین کاربردی

عنوان

تمرین اول (HW01)

مدرس

دکتر احسان ناظر فرد

دانشجو

امیرحسین بابائیان

۴۰۱۱۳۱۰۰۲

ترم بهار ۰۲-۰۱

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر (پلی تکنیک تهران)

فهرست

فهرست تصاویر

سوال اول

بخش a

بله، می‌توان از الگوریتم K-Nearest Neighbors (KNN) برای مسئله دسته‌بندی چند کلاسه استفاده کرد. در این الگوریتم، با استفاده از سیستم رأی‌گیری اکثریتی بین k نزدیک‌ترین همسایه، کلاس نمونه ورودی تعیین می‌شود. به عبارت دیگر، با شناسایی k نزدیک‌ترین همسایه به نمونه ورودی و انجام رأی‌گیری برای تعیین کلاس، می‌توان برای مسائل دسته‌بندی چند کلاسه از KNN استفاده کرد.

بخش b

Curse of dimensionality به این معنی است که با افزایش تعداد ابعاد داده‌ها، فضای داده‌ها به طور چشمگیری بزرگتر می‌شود. در الگوریتمKNN ، عملکرد آن به طور مستقیم با تعداد ابعاد داده‌ها مرتبط است. افزایش تعداد ابعاد داده‌ها باعث می‌شود که نقاط به هم دور شده و در نتیجه کارایی الگوریتم KNN کاهش می‌یابد. این به این معنی است که با افزایش تعداد ابعاد، پیدا کردن همسایه‌های مناسب برای هر نقطه بسیار دشوارتر می‌شود. به عبارت دیگر، بالا رفتن تعداد ابعاد باعث کاهش دقت و کارایی الگوریتم KNN می‌شود.

بخش c

اصولاً KNN برای وظایف یادگیری با ناظر استفاده می‌شود ولی می‌توان از آن در برخی وظایف یادگیری بدون ناظر نیز استفاده کرد. در این حالت، KNNمعمولاً به عنوان یک الگوریتم خوشه‌بندی استفاده می‌شود. بدین منظور، ابتدا نزدیک‌ترین همسایه‌های هر نقطه را پیدا کرده و سپس با استفاده از این اطلاعات، نقاطی که به هم نزدیک هستند را در یک خوشه قرار می‌دهیم. به این ترتیب، می‌توانیم داده‌ها را بر اساس شباهت به یکدیگر گروه‌بندی کنیم، استفاده از KNN در وظایف یادگیری بدون ناظر به دلیل سادگی و کارایی آن، مخصوصاً برای داده‌هایی با تعداد بالا، بسیار مناسب است.

بخش d

KNN الگوریتمی برای دسته‌بندی داده‌ها است که برای پیش‌بینی برچسب داده‌های جدید از نزدیک‌ترین همسایه‌های آن‌ها استفاده می‌کند. در مقابل، الگوریتم‌های دسته‌بندی دیگری مانند Naive Bayes و درخت تصمیم‌گیری، با استفاده از الگوهای آماری و احتمالاتی برچسب داده‌های جدید را پیش‌بینی می‌کنند.

درNaive Bayes ، به دنبال پیدا کردن توزیع احتمال برای هر برچسب داده جدید هستیم، در حالی که در درخت تصمیم‌گیری، به دنبال یافتن یک راهبرد بهینه برای تقسیم داده‌ها به گروه‌های مختلف هستیم.

بنابراین، اصلی‌ترین تفاوت بین KNN و سایر الگوریتم‌های دسته‌بندی، در روشی است که برای پیش‌بینی برچسب داده‌های جدید استفاده می‌شود. در KNN از نزدیک‌ترین همسایه‌ها استفاده می‌شود، در حالی که در الگوریتم‌های دسته‌بندی دیگر از الگوهای آماری و احتمالاتی و یا روش‌های تقسیم داده‌ها استفاده می‌شود.

بخش e

انتخاب معیار فاصله بر عملکرد الگوریتم KNN تأثیر زیادی دارد. از آنجایی که KNN بر پایه‌ی معیار فاصله کار می‌کند، انتخاب معیار مناسب به اهمیت بالایی برخوردار است.

معیار فاصله اقلیدسی معمولاً برای داده‌هایی که در فضای برداری هستند، مناسب است. این معیار بر اساس فاصله اقلیدسی بین دو نقطه در فضای n-بعدی محاسبه می‌شود.

معیار فاصله منهتن برای داده‌هایی که مختصاتشان با مقادیر نامتناهی یا بسیار بزرگ هستند، مفید است. در این معیار فاصله، فاصله بین دو نقطه برابر با جمع مقدار مطلق تفاضل بین مختصاتشان است.

معیار فاصله کسینوس برای داده‌هایی که مرتبط با زوایا و جهت هستند، مناسب است. در این معیار فاصله، فاصله بین دو نقطه برابر با کسینوس زاویه بین دو بردار است.

با توجه به مشخصات هر معیار فاصله، انتخاب معیار مناسب بسته به خصوصیات داده‌ها و مسئله‌ی مورد نظر در الگوریتم KNN بسیار مهم است. به عنوان مثال، اگر داده‌ها در فضای برداری هستند، معیار اقلیدسی مناسب است. اما اگر داده‌ها مرتبط با زوایا و جهت هستند، معیار کسینوس مناسب است.

بخش f

فرایند انتخاب مقدار بهینه K در KNN به شرح زیر است:

۱. جدول داده ها را به دو بخش تقسیم کنید: بخش آموزش و بخش تست.
۲. از بخش آموزش استفاده کرده و مدل KNN را با مقادیر مختلف K آموزش دهید.
۳. سپس با استفاده از بخش تست، عملکرد مدل را بررسی کنید و مقدار K بهینه را بر اساس عملکرد مدل در بخش تست انتخاب کنید.

متد های مختلفی برای انتخاب مقدار بهینه K وجود دارد که عبارتند از:

۱. تعیین مقدار K بر اساس قوانین تجربی: در این روش، مقدار K به صورت دستی تعیین می شود و توسط کارشناسان مسئول انجام می شود. این روش ساده است اما ممکن است به دلیل عدم استفاده از داده های واقعی، بهینه نباشد.

۲. تعیین مقدار K بر اساس تقسیم داده ها: در این روش، مقدار K بر اساس تعداد داده های آموزش و تست تقسیم بر دو تعیین می شود. این روش معمولاً به خوبی عمل می کند ولی ممکن است به صورت غیر قابل پیش بینی بهینه نباشد.

۳. استفاده از روش کراس ولیدیشن: در این روش، داده ها به چند بخش تقسیم می شوند و مدل با استفاده از هر بخش به صورت مجزا آموزش داده می شود. سپس عملکرد مدل با استفاده از بخش تست بررسی می شود و مقدار K بهینه بر اساس عملکرد مدل بر روی بخش تست تعیین می شود. این روش معمولاً به خوبی عمل می کند و باعث افزایش دقت مدل می شود.

بخش g

رگرسیون وزن‌دار محلی در الگوریتم KNN از رگرسیون KNN استاندارد به دو دلیل متمایز است. اولاً، تنها یک زیرمجموعه کوچکتر از همسایگان با وزن بیشتر برای پیش‌بینی استفاده می‌شود. دوماً، وزن‌ها بر اساس فاصله نسبی از نقطه داده به همسایگان آن محاسبه می‌شوند. از طرفی، رگرسیون KNN استاندارد تمام همسایگان را برای پیش‌بینی استفاده می‌کند و هیچ وزنی برای آن‌ها تعیین نمی‌شود.

مزیت استفاده از رگرسیون وزن‌دار محلی در الگوریتم KNN این است که دقت پیش‌بینی بهبود می‌یابد و الگوریتم قادر است به نواحی تخصصی نزدیک‌تر به هر نقطه توجه کند.

همچنین استفاده از وزن‌ها به این معنی است که نقاط داده‌ای که در فاصله نزدیک‌تری به نقطه تحت بررسی هستند، بیشترین تأثیر را در پیش‌بینی نقطه تحت بررسی دارند.

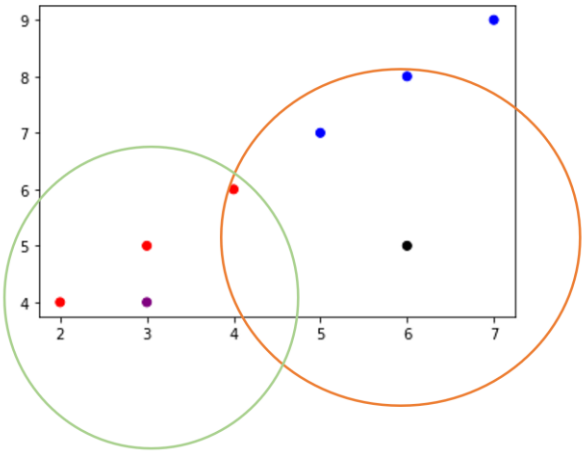
از معایب رگرسیون وزن‌دار محلی می‌توان به این نکته اشاره کرد که محاسبه وزن‌ها برای هر نقطه داده محاسبات بیشتری نیاز دارد و می‌تواند زمان پردازش را افزایش دهد. علاوه بر این، در برخی موارد، الگوریتم KNN با استفاده از تمام همسایگان نسبت به رگرسیون وزن‌دار محلی دقت بیشتری در پیش‌بینی داشته باشد.

برخی از کاربردهای رگرسیون وزن‌دار محلی در یادگیری ماشین شامل پیش‌بینی قیمت مسکن بر اساس ویژگی‌های مختلف، پیش‌بینی میزان آلودگی هوا بر اساس عوامل مختلف مانند دما و فشار هوا، تخمین بازدهی سهام بر اساس عوامل مختلف اقتصادی و تاریخی، پیش‌بینی خطر بیماری بر اساس عوامل مختلفی مانند سن، جنسیت و سابقه بیماری و ... است.

در کل، رگرسیون وزن‌دار محلی یک روش کارآمد برای پیش‌بینی دقیق‌تر در مواردی است که اطلاعات نزدیک به نقطه داده در نظر گرفته شده است و در مقایسه با رگرسیون KNN استاندارد، دارای دقت بالاتری می‌باشد.

بخش h

برای نمایش نقاط روی صفحه یک قطعه کد نوشتم و تصویر مقابل را تولید کردم.



تصویر ۱) خروجی قطعه کد پاسخ بخش h سوال ۱

آبی رنگ ها دسته بندی B و قرمز رنگ ها دسته بندی A را نمایش می دهند هر کدام از نمونه های ۷ و ۸ به ترتیب بنفش و مشکی نمایش داده شده اند.

با در نظر گرفتن K=3 داریم که فاصله برای بنفش از کلاس A کمتر است پس در این دسته قرار می گیرید و همچنین برای مشکی نیز کلاس B فاصله کمتری دارد. (به ترتیب دایره های سبز و نارنجی گویا هستند.)

بخش i

در یادگیری ماشین، پارامترهای مدل و هایپرپارامترهای الگوریتم یادگیری دو نوع پارامتر هستند که به طور کلی متفاوت عمل می‌کنند.

پارامترهای مدل مقادیری هستند که توسط الگوریتم یادگیری به‌صورت خودکار تنظیم می‌شوند و ارزش آن‌ها توسط داده‌های آموزش تعیین می‌شود. به‌عنوان مثال، در یک مدل خطی، ضرایب جمع و ضرب به‌صورت خودکار توسط الگوریتم تعیین می‌شوند و مقدار آن‌ها به‌صورت مستقیم توسط داده‌های آموزش تعیین می‌گردد.

هایپرپارامترها به‌عنوان پارامترهایی که در انتخاب الگوریتم یادگیری و تنظیم آن استفاده می‌شوند، تعریف می‌شوند. این پارامترها توسط کاربر مشخص می‌شوند و نمی‌توانند به‌صورت خودکار توسط الگوریتم تنظیم شوند. به‌عنوان مثال، در الگوریتم KNN ، هایپرپارامترهایی مانند تعداد همسایگان و معیار فاصله مورد استفاده قرار می‌گیرند که باید توسط کاربر مشخص شوند.

از مزیت‌های پارامترهای مدل این است که مقادیر آن‌ها به‌صورت خودکار توسط الگوریتم یادگیری تنظیم می‌شوند و به‌صورت مستقیم به داده‌های آموزش بستگی دارند. از طرفی، هایپرپارامترها به‌عنوان پارامترهای کنترلی الگوریتم یادگیری عمل می‌کنند و به کاربر اجازه می‌دهند که بهترین مدل را از بین الگوریتم‌های یادگیری مختلف انتخاب کند.

بنابراین، پارامترهای مدل و هایپرپارامترها با یکدیگر تفاوت دیگری هم دارند که این تفاوت در تعیین مقدار آن‌هاست. پارامترهای مدل توسط الگوریتم یادگیری به‌صورت خودکار تنظیم می‌شوند و مقدار آن‌ها به‌صورت مستقیم توسط داده‌های آموزش تعیین می‌شود. اما هایپرپارامترها مقادیر ثابتی دارند و باید توسط کاربر به‌صورت دستی تعیین شوند.

استفاده از پارامترهای مدل باعث می‌شود که مدل به‌صورت خودکار به داده‌های آموزش تطبیق پیدا کند و دقت مدل در پیش‌بینی داده‌های تست بهبود یابد. اما استفاده از هایپرپارامترها به کاربر اجازه می‌دهد تا الگوریتم یادگیری را بهترین شکل ممکن تنظیم کند و از مدلی استفاده کند که دقت بالاتری دارد.

در کل، هایپرپارامترها می‌توانند تأثیر قابل‌توجهی بر روی دقت مدل داشته باشند، اما تعیین بهترین مقدار برای آن‌ها به دلیل نیاز به تست مجدد بر روی داده‌ها، می‌تواند زمان‌بر و پر هزینه باشد. در نتیجه، انتخاب هایپرپارامترها باید با دقت و بررسی دقیق انجام شود

بخش j

۱ .کیفیت و کمیت داده‌ها: این چالش بیانگر این موضوع است که داده‌ها باید کیفیت و کمیت خوبی داشته باشند تا مدل‌های یادگیری ماشین به‌صورت دقیق و قابل‌اعتماد کار کنند.

۲ .انتخاب ویژگی‌ها: انتخاب ویژگی‌های مناسب از داده‌ها می‌تواند به‌عنوان یک چالش مهم در یادگیری ماشین باشد. این موضوع به دلیل این است که تعداد بسیار زیادی ویژگی‌ها می‌توانند برای داده‌ها وجود داشته باشند و لازم است که از بین آن‌ها ویژگی‌هایی که مفیدتر هستند انتخاب شوند.

۳ .پیچیدگی مدل: پیچیدگی مدل یک چالش مهم در یادگیری ماشین است. بعضی از مدل‌های یادگیری ماشین بسیار پیچیده هستند و به‌صورت خودکار قابل‌تنظیم نیستند. به‌عنوان نمونه، مدل‌های شبکه‌های عصبی عمیق چالش‌هایی مانند آموزش، تنظیم پارامتر و غیره دارند.

۴ .پیچیدگی محیط: پیچیدگی محیط نیز یک چالش در یادگیری ماشین است. بعضی از محیط‌ها، مانند محیط‌هایی که دارای شرایط آموزش بسیار گسترده هستند، بسیار پیچیده هستند و برای مدل‌های یادگیری ماشین مشکل‌ساز هستند.

بخش k

اگر مدل شما در داده‌های آموزشی عملکرد عالی داشته باشد، اما به نمونه‌های جدید به خوبی عمومی سازی نشود، ممکن است مشکلی وجود داشته باشد.

این مشکل به عنوان بیش‌برازش (Overfitting) شناخته می‌شود، به این معنی که مدل برای داده‌های آموزشی بسیار پیچیده شده و اطلاعات جزئی و نویز در آن داده‌ها را به یاد گرفته است، اما نتوانسته است الگوهای کلی و عمومی را درک کند.

ترجیحاً باید از روش‌های زیر استفاده کنید تا این مشکل را حل کنید:

۱ .استفاده از داده‌های بیشتر: جمع‌آوری داده‌های بیشتری که شامل اطلاعات متنوعی از شرایط مختلف است، می‌تواند بهبودی در عملکرد عمومی مدل برای داده‌های جدید داشته باشد.

۲ .استفاده از تکنیک‌های کاهش اندازه مدل: با کاهش اندازه مدل و از بین بردن بعضی از پارامترهای آن می‌توان از بیش‌برازش جلوگیری کرد.

۳ .استفاده از روش‌های مانع‌سازی بیش‌برازش: مثل Dropout که با حذف برخی ویژگی‌ها در هر مرحله آموزش، از بیش‌برازش جلوگیری می‌کند.

بخش l

اگر شما با استفاده از مجموعه داده آزمون، هایپرپارامترهای مدل خود را تنظیم کنید، ممکن است مشکلاتی بوجود آید. اگر هایپرپارامترها برای بهتر شدن عملکرد در داده‌های آزمون تنظیم شوند، احتمال اینکه مدل برای داده‌های دیگر به درستی کار نکند، بسیار بالا است. این مشکل به عنوان بیش‌برازش به داده آزمون شناخته می‌شود و ممکن است باعث شود که مدل شما برای داده‌های جدید به درستی کار نکند. بنابراین برای تنظیم هایپرپارامترها باید از یک مجموعه داده جداگانه به نام مجموعه اعتبارسنجی (Validation set)استفاده کرد.

بخش m

در رگرسیون چند جمله ای، اگر بررسی منحنی‌های یادگیری، تفاوت بزرگی بین خطاهای آموزش و اعتبارسنجی وجود داشته باشد، این به این معناست که مدل در برابر بررسی داده‌های جدید (اعتبارسنجی) کمترین کارایی را از خود به نمایش گذاشته است. این امر به دلیل بیش‌برازش (overfitting) مدل رخ می‌دهد .برای رفع این مشکل، سه روش می‌توان به کار برد:

۱. استفاده از روش‌های کاهش اندازه ویژگی (Feature Selection) و کاهش پیچیدگی یا ساده سازی مدل (Model Simplification)
۲. استفاده از روش‌های جلوگیری از بیش‌برازش، مانندRegularization
۳. افزایش تعداد داده‌های آموزش(Training Data)

بخش n

۱. رگرسیون ریج به جای رگرسیون خطی ساده (بدون هیچ گونه تنظیم) به دلیل جلوگیری از بیش‌برازش (overfitting) و کاهش وابستگی متغیرهای ورودی استفاده می‌شود. این روش از جمله تکنیک‌های رگولاریزاسیون است که به دلیل اضافه کردن جمله شامل مقدار مطلق همواره مثبت از پارامترهای مدل، می‌تواند باعث کاهش بیش‌برازش و بهبود کارایی مدل شود.

۲. لاسو به جای رگرسیون ریج به دلیل اینکه می‌تواند متغیرهای ورودی را انتخاب کند و از بین بردارد که موجب کاهش ابعاد داده و افزایش کارایی مدل می‌شود. به عبارت دیگر، لاسو به عنوان یک الگوریتم انتخاب ویژگی (feature selection) مفید است.

۳. الستیک نت به دلیل داشتن مزیت‌های رگرسیون ریج و لاسو به عنوان ترکیبی از دو روش، بهتر از هر دوی آن‌ها استفاده می‌شود. به این صورت که جمله رگولاریزاسیون در الستیک نت شامل یک ترم لاسو و یک ترم ریج می‌شود، به این ترتیب می‌تواند بهتر از رگرسیون ریج و لاسو با هم ترکیب شوند و کارایی مدل را بهبود بخشند.

سوال دوم

فایل کد در پوشه ی Source Codes با عنوان P2.ipynb قرارداده شده است.

بخش a

اندازه دیتا ست برابر است با ۱۳۸۳ در ۱۳ یعنی ۱۳۸۳ سطر و ۱۳ ستون که هر ستون یک فیچر است.

ده داده ی تصادفی نیز به شرح ذیل است:

(1303, 13)

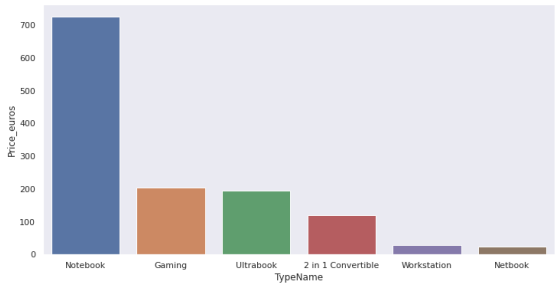
	laptop_ID	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price_euros
238	243	Asus	ROG G703VI-E5062T	Gaming	17.3	Full HD 1920x1080	Intel Core i7 7820HK 2.9GHz	32GB	512GB SSD + 1TB HDD	Nvidia GeForce GTX 1080	Windows 10	4.7kg	3890.0
1149	1167	Lenovo	ThinkPad X1	2 in 1 Convertible	14.0	IPS Panel Touchscreen 2560x1440	Intel Core i7 6500U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 520	Windows 10	1.27kg	2339.0
227	232	Asus	Vivobook X541UV-DM1217T	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Nvidia GeForce 920MX	Windows 10	2kg	769.0
8	9	Asus	ZenBook UX430UN	Ultrabook	14.0	Full HD 1920x1080	Intel Core i7 8550U 1.8GHz	16GB	512GB SSD	Nvidia GeForce MX150	Windows 10	1.3kg	1495.0
1015	1029	Toshiba	Portege A30-C-1CZ	Notebook	13.3	1366x768	Intel Core i5 6200U 2.3GHz	8GB	256GB SSD	Intel HD Graphics 520	Windows 10	1.5kg	1210.0
251	256	Asus	ROG G752VSK-GC493T	Gaming	17.3	Full HD 1920x1080	Intel Core i7 7700HQ 2.8GHz	16GB	256GB SSD + 1TB HDD	Nvidia GeForce GTX 980M	Windows 10	4.3kg	1799.0
327	333	Asus	VivoBook S15	Ultrabook	15.6	Full HD 1920x1080	Intel Core i7 7500U 2.7GHz	8GB	256GB SSD	Nvidia GeForce 940MX	Windows 10	1.7kg	1049.9
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
935	949	HP	EliteBook 820	Netbook	12.5	Full HD 1920x1080	Intel Core i5 6200U 2.3GHz	8GB	256GB SSD	Intel HD Graphics 520	Windows 10	1.26kg	1669.0
937	951	MSI	GP62M Leopard	Gaming	15.6	Full HD 1920x1080	Intel Core i7 7700HQ 2.8GHz	8GB	128GB SSD + 1TB HDD	Nvidia GeForce GTX 1050	Windows 10	2.2kg	1199.0

تصویر ۲) خروجی ۱۰ داده تصادفی

بخش b

نمودار barplot

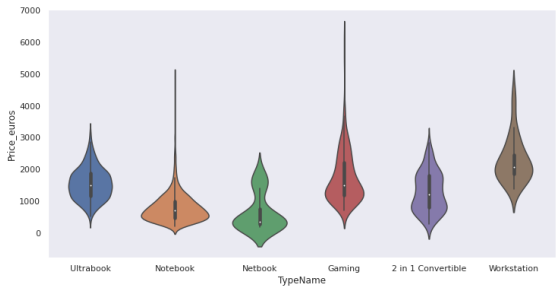
خب این نمودار تعداد بر حسب نوع لپ تاپ آورده شده است. (به Price_euros توجه نشود.)



تصویر ۳) نمودار barplot

نمودار violinplot

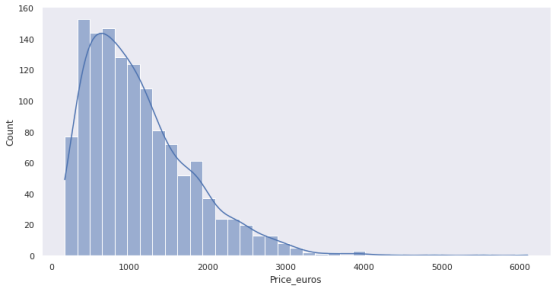
این نمودار مشابه نمودار جعبه ای عمل می کند با این تفاوت که می توان چندین توزیع را با یکدیگر مقایسه کرد.



تصویر ۴) نمودار violonplot

نمودار histplot

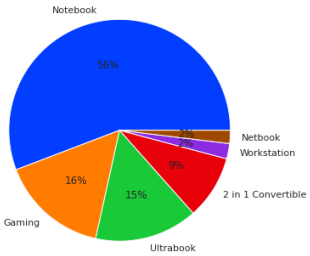
نمودار هیستوگرام داده ها تعداد بر مبنای قیمت را در این بخش داریم.



تصویر ۵) نمودار histogram

نمودار pieplot

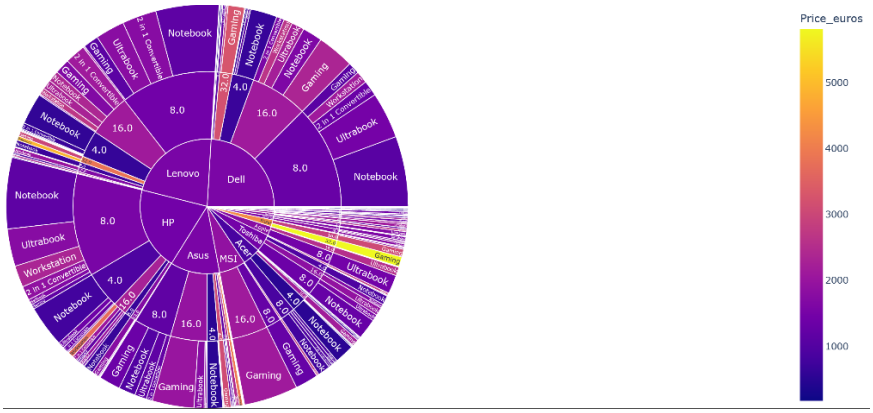
نمودار دایره ای تعداد هر نوع را در شکل ذیل مشاهده می کنید.



تصویر ۶) نمودار pieplot

نمودار sunburst

یکی از جذاب ترین نمودار ها که می توان رسم کرد نمودار sunburst است در این نمودار هر شرکت بر حسب میزان حضورش در بازار در گام اول تقسیم بندی شده و در قدم بعدی نیز بر حسب تولیدات متفاوتی که ارائه می کند.



تصویر ۷) نمودار sunburst

دیگر نمودار ها

نمودار های دیگری نیز رسم شده است که در بخش کد قابل مشاهده می باشد.

بخش c و d

قدم اول – حذف ستون و بدست آوردن اطلاعات مفید

در این قدم ستون laptop_ID که بدون کار برد بود را حذف نمودیم.

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price_euros
0	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8GB	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37kg	1339.69
1	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8GB	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34kg	898.94
2	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8GB	256GB SSD	Intel HD Graphics 620	No OS	1.86kg	575.00
3	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16GB	512GB SSD	AMD Radeon Pro 455	macOS	1.83kg	2537.45
4	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8GB	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37kg	1803.60

تصویر ۸) خروجی پس از حذف ستون laptop_ID

سپس داده ها را از نظر نوع بررسی نمودیم تا اطلاعات مربوط به هریک را بدست آوریم و بتوانیم تغییرات مد نظر را اعمال کنیم.

	column	dtypes	nunique	sum_null
0	Company	object	19	0
1	Product	object	618	0
2	TypeName	object	6	0
3	Inches	float64	18	0
4	ScreenResolution	object	40	0
5	Cpu	object	118	0
6	Ram	object	9	0
7	Memory	object	39	0
8	Gpu	object	110	0
9	OpSys	object	9	0
10	Weight	object	179	0
11	Price_euros	float64	791	0

تصویر ۹) خروجی اطلاعات ویژگی ها

قدم دوم – تبدیل رشته های KG و GB

در این قدم داده های ستون هایی که عددی هستن اما قابل کار عددی رویشان نیست را تغییر می دهیم، مانند ستون weight و RAM.

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price_euros
0	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 2.3GHz	8.0	128GB SSD	Intel Iris Plus Graphics 640	macOS	1.37	1339.69
1	Apple	Macbook Air	Ultrabook	13.3	1440x900	Intel Core i5 1.8GHz	8.0	128GB Flash Storage	Intel HD Graphics 6000	macOS	1.34	898.94
2	HP	250 G6	Notebook	15.6	Full HD 1920x1080	Intel Core i5 7200U 2.5GHz	8.0	256GB SSD	Intel HD Graphics 620	No OS	1.86	575.00
3	Apple	MacBook Pro	Ultrabook	15.4	IPS Panel Retina Display 2880x1800	Intel Core i7 2.7GHz	16.0	512GB SSD	AMD Radeon Pro 455	macOS	1.83	2537.45
4	Apple	MacBook Pro	Ultrabook	13.3	IPS Panel Retina Display 2560x1600	Intel Core i5 3.1GHz	8.0	256GB SSD	Intel Iris Plus Graphics 650	macOS	1.37	1803.60
...	...	...	...	...	...	...	...	...	...	...	...	...
1298	Lenovo	Yoga 500-14ISK	2 in 1 Convertible	14.0	IPS Panel Full HD / Touchscreen 1920x1080	Intel Core i7 6500U 2.5GHz	4.0	128GB SSD	Intel HD Graphics 520	Windows 10	1.80	638.00
1299	Lenovo	Yoga 900-13ISK	2 in 1 Convertible	13.3	IPS Panel Quad HD+ / Touchscreen 3200x1800	Intel Core i7 6500U 2.5GHz	16.0	512GB SSD	Intel HD Graphics 520	Windows 10	1.30	1499.00
1300	Lenovo	IdeaPad 100S-14IBR	Notebook	14.0	1366x768	Intel Celeron Dual Core N3050 1.6GHz	2.0	64GB Flash Storage	Intel HD Graphics	Windows 10	1.50	229.00
1301	HP	15-AC110nv (i7-6500U/6GB /1TB/Radeon	Notebook	15.6	1366x768	Intel Core i7 6500U 2.5GHz	6.0	1TB HDD	AMD Radeon R5 M330	Windows 10	2.19	764.00
1302	Asus	X553SA-XX031T (N3050/4GB /500GB/W10)	Notebook	15.6	1366x768	Intel Celeron Dual Core N3050 1.6GHz	4.0	500GB HDD	Intel HD Graphics	Windows 10	2.20	369.00

1303 rows × 12 columns

تصویر ۱۰) حذف رشته های زائد

قدم سوم – labelencoder

ستون هایی که محتوای آن ها شامل توضیحات متنی بود را به عدد تبدیل کردیم.

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight	Price_euros
0	1	300	4	13.3	23	65	8.0	4	58	8	1.37	1339.69
1	1	301	4	13.3	1	63	8.0	2	51	8	1.34	898.94
2	7	50	3	15.6	8	74	8.0	16	53	4	1.86	575.00
3	1	300	4	15.4	25	85	16.0	29	9	8	1.83	2537.45
4	1	300	4	13.3	23	67	8.0	16	59	8	1.37	1803.60
...	...	...	...	...	...	...	...	...	...	...	...	...
1298	10	580	0	14.0	13	89	4.0	4	47	5	1.80	638.00
1299	10	588	0	13.3	19	89	16.0	29	47	5	1.30	1499.00
1300	10	196	3	14.0	0	34	2.0	35	40	5	1.50	229.00
1301	7	2	3	15.6	0	89	6.0	10	21	5	2.19	764.00
1302	2	568	3	15.6	0	34	4.0	26	40	5	2.20	369.00

1303 rows × 12 columns

تصویر ۱۱) خروجی پس از labelEncoding

قدم چهارم – تقسیم کردن داده ها

داده ها را به بخش های مختلف train و test و validation تقسیم میکنیم، در تصویر ذیل می بینیم که صرفا داده های تست که ۲۵ درصد کل داده ها هستند آورده شده است(تعداد داده های تست ۳۲۶ می باشد.):

X_test

	Company	Product	TypeName	Inches	ScreenResolution	Cpu	Ram	Memory	Gpu	OpSys	Weight
644	0	127	3	15.6	0	56	4.0	26	47	5	2.40
1275	2	602	4	13.3	15	45	8.0	29	46	5	1.20
163	10	291	1	15.6	15	102	16.0	16	76	5	2.50
1008	7	362	3	14.0	8	74	8.0	16	69	5	1.64
909	9	193	4	14.0	13	97	8.0	29	53	5	0.98
...	...	...	...	...	...	...	...	...	...	...	...
34	1	299	4	13.3	1	62	8.0	15	51	3	1.35
85	4	258	1	15.6	15	102	16.0	5	76	5	2.65
979	2	406	1	17.3	8	102	16.0	18	76	5	2.99
156	10	583	0	14.0	13	59	4.0	16	53	5	1.74
1130	7	27	3	15.6	0	97	8.0	22	53	5	2.04

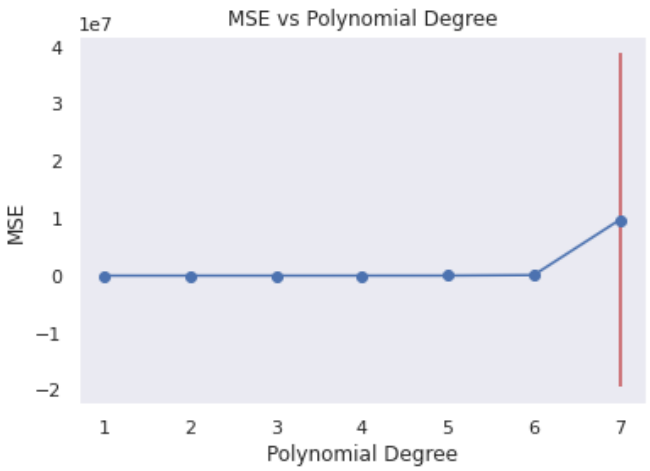
326 rows × 11 columns

تصویر ۱۲) داده های تست

بخش e

در این بخش ما با قرار دادن k=10 برای دسته بندی داده ها به دسته در kfold اقدام کردیم و برای رنج ۱ تا ۷ همسایه کار را اجرا نمودیم و براساس محاسبات انجام شده نمودار ذیل را به عنوان خروجی نمایش دادیم.

(این فرایند بر روی google colab نزدیک به ۵ دقیقه زمان برد، در ابتدا برای ۱ تا ۱۰ تست کردیم که پس از ۳۰ دقیقه با Execution Error مواجه شدیم.)



تصویر ۱۳) خروجی MSE با درجه های مختلف

مقیاس نمودار ۱۰ به توان ۷ می باشد از این رو بایستی داده ها را به صورت ذیل مشاهده کرد.

```
R = 1 ==> MSE_MEAN: 0.3428438422414889 , MSE_STD: 0.10892880838762807
R = 2 ==> MSE_MEAN: 0.25914132147079844 , MSE_STD: 0.07722952240504972
R = 3 ==> MSE_MEAN: 0.4976313703954771 , MSE_STD: 0.43206002900712503
R = 4 ==> MSE_MEAN: 4267.655507821013 , MSE_STD: 4035.2358671806664
R = 5 ==> MSE_MEAN: 9830.384439699625 , MSE_STD: 29210.280121261232
R = 6 ==> MSE_MEAN: 101522.64027319077 , MSE_STD: 303864.72559281
R = 7 ==> MSE_MEAN: 9749465.352966739 , MSE_STD: 29245337.604114253
```

تصویر ۱۴) میانگین و انحراف از معیار MSE بر اساس تعداد همسایه ها

بر این اساس K=2 را به عنوان عدد بهتر بر میگزینیم.

بخش f

براساس داده های قبلی با استفاده از رگرسیون چندجمله ای داده ها را برای آموزش و تست دسته بندی نموده و در نهایت R2_Score و MSE را محاسبه می کنیم که به شرح ذیل می باشد.

```
model.score(X_test,y_test)
```

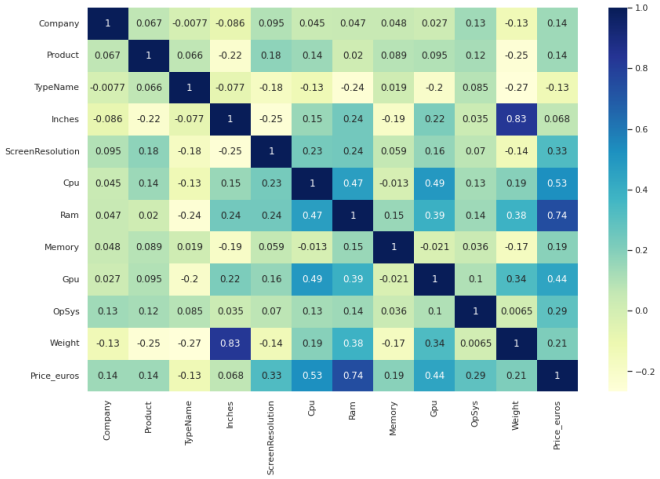
```
0.7874859220051293
```

```
R2: 0.7874859220051293
MSE: 0.1993097967180412
```

تصویر ۱۵) خروجی MSE، R2 و model score

بخش g

بر اساس هیت مپ تصویر مقابل در میابیم که بیشترین ارتباط با قیمت بین ویژگی ها مربوط به Ram می شود.



تصویر ۱۶) جدول Hitmap

سوال سوم

فایل کد در پوشه ی Source Codes با عنوان P3.ipynb برای بخش مصوری سازی و P3v2.ipynb برای بخش رگرسیون قرارداده شده است.

توضیحات دیتاست

مجموعه داده کیفیت هوایی، داده‌هایی است که در مورد کیفیت هوای یک منطقه خاص و در یک بازه زمانی مشخص، جمع‌آوری شده‌اند. این داده‌ها ممکن است شامل اطلاعاتی درباره آلاینده‌های مختلف هوا باشند، مانند ذرات معلق، ازون، نیتروژن اکسید، دی اکسید کربن و سایر آلاینده‌های هوایی.

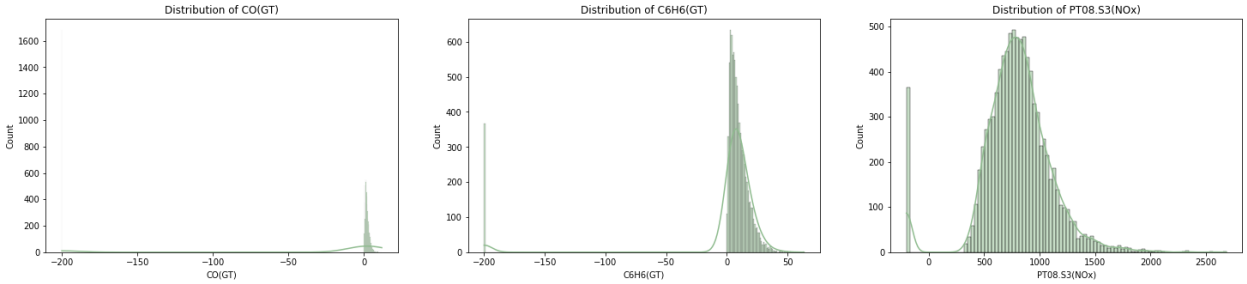
علاوه بر آلاینده‌های هوایی، این داده‌ها می‌توانند شامل اطلاعات هواشناسی مانند دما، رطوبت، سرعت و جهت باد، فشار هوا و تعداد ساعت آفتابی باشند. این داده‌ها به عنوان یک منبع ارزشمند برای بررسی روندهای آلاینده‌های هوایی و تأثیر آن‌ها بر سلامتی و محیط زیست استفاده می‌شوند.

از این داده‌ها برای تحلیل و پیش‌بینی کیفیت هوا، تعیین راهکارهای کاهش آلودگی هوا و تحلیل تأثیر آلودگی هوا بر سلامتی انسان و محیط زیست استفاده می‌شود. به عنوان مثال، می‌توان از این داده‌ها برای تحلیل تأثیر آلاینده‌های هوایی بر بیماری‌های تنفسی و قلبی، تعیین اثرات زیست محیطی سیاست‌های کاهش آلودگی هوا، و تعیین ارزش‌های اقتصادی آلودگی هوا استفاده کرد.

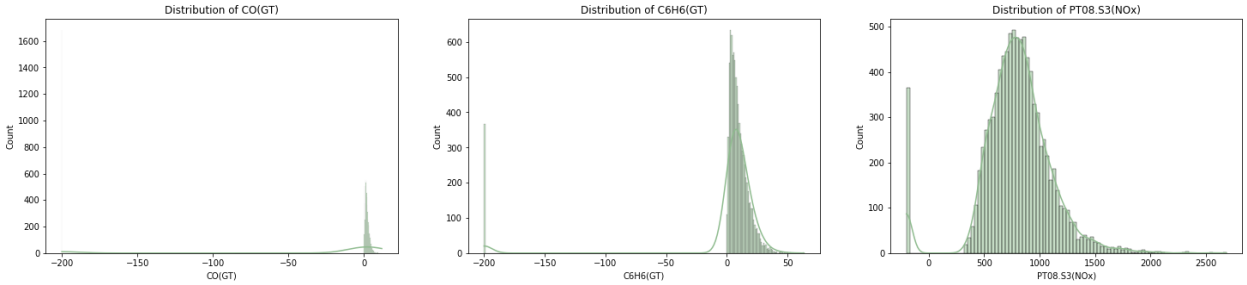
بخش a

رسم نمودار ها در این بخش انجام شده است، توضیح کوتاه هر یک به عنوان کپشن جدول اضافه می شود که گزارش کار بیش از این طولانی نشود.

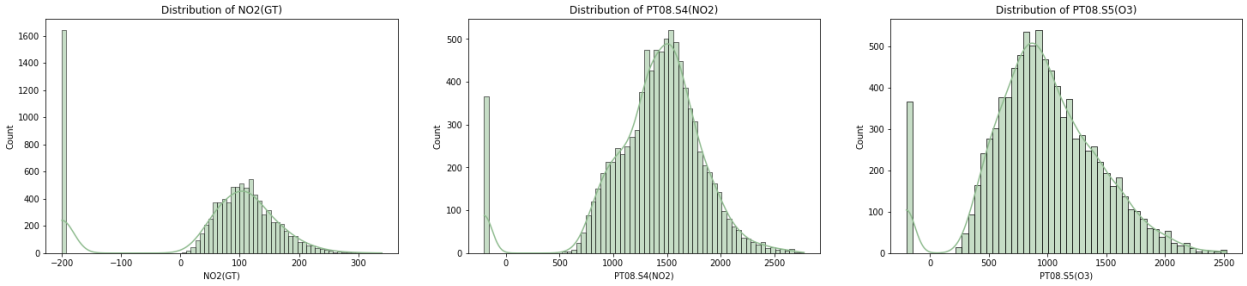
این نمودار جهت بررسی میزان نرمال بودن نسبت به یکدیگر گرفته شده اند.



تصویر ۱۷) نمودار توزیع بر حسب سه ویژگی اول

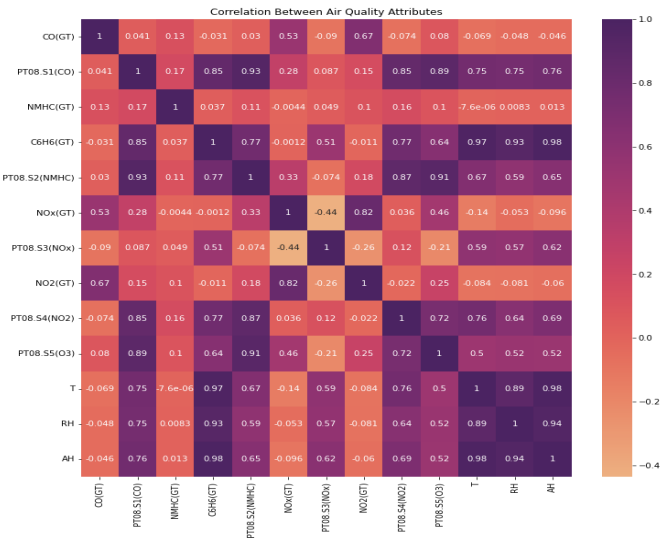


تصویر ۱۸) نمودار توزیع بر حسب سه ویژگی دوم



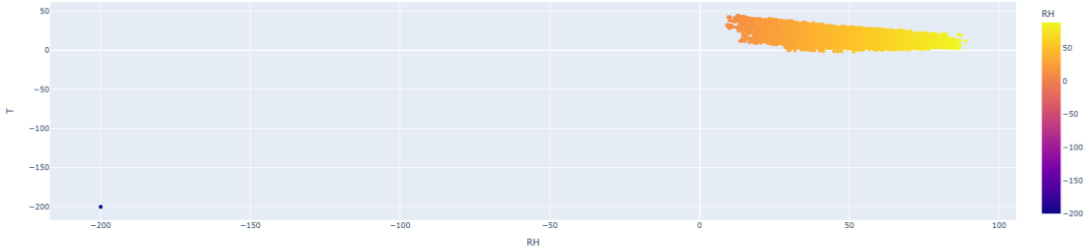
تصویر ۱۹) نمودار توزیع بر حسب سه ویژگی سوم

از هیت مپ برای دریافت نزدیکی و ارتباط بین داده ها استفاده میکنیم:

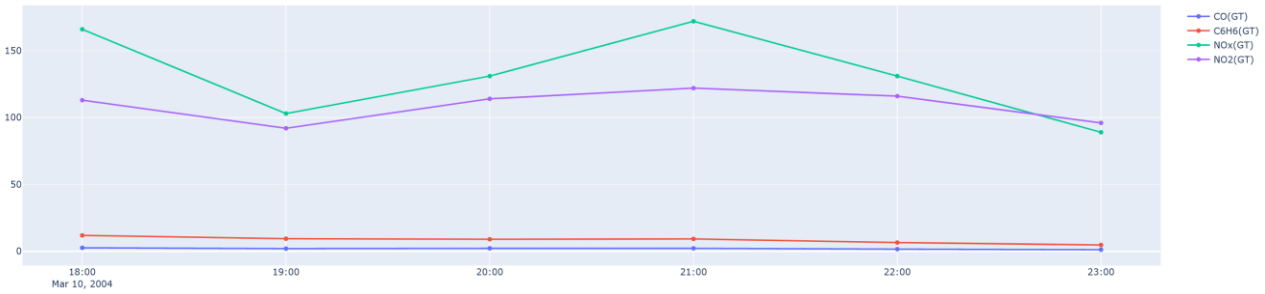


تصویر ۲۰) نمودار هیت مپ تمامی ویژگی ها

این نمودار برای بررسی ارتباط میان رطوب و دما استفاده شده است ک ه یک داده پرت را مشاهده میکنیم.



تصویر ۲۱) نمودار ارتباط بین دما و رطوبت



تصویر ۲۲) نمودار میزان تغییر ۴ ویژگی از ساعت ۱۸ الی ۲۳

بخش b

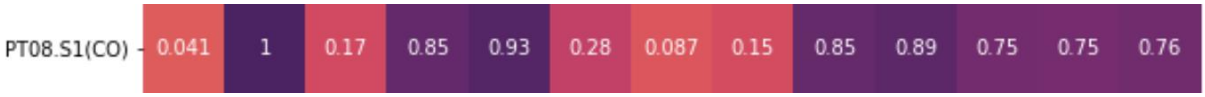
دستور dropna اجرا شد برای حذف داده های دارای null اما هیچ سطری بابت این مورد حذف نشد، در این بخش ستون time و date با یکدیگر ترکیب شد و به عنوان یک datetimeObject ذخیره سازی شد.

	Date	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	AH
0	2004-03-10 18:00:00	2.6	1360.00	150	11.881723	1045.50	166.0	1056.25	113.0	1692.00	1267.50	13.60	48.875001	0.757754
1	2004-03-10 19:00:00	2.0	1292.25	112	9.397165	954.75	103.0	1173.75	92.0	1558.75	972.25	13.30	47.700000	0.725487
2	2004-03-10 20:00:00	2.2	1402.00	88	8.997817	939.25	131.0	1140.00	114.0	1554.50	1074.00	11.90	53.975000	0.750239
3	2004-03-10 21:00:00	2.2	1375.50	80	9.228796	948.25	172.0	1092.00	122.0	1583.75	1203.25	11.00	60.000000	0.786713
4	2004-03-10 22:00:00	1.6	1272.25	51	6.518224	835.50	131.0	1205.00	116.0	1490.00	1110.00	11.15	59.575001	0.788794

تصویر ۲۳) مجموعه داده ها پس از تغییرات اعمال شده

برای مهندسی ویژگی ها را برای بررسی می توانیم از هیت مپی که در بخش قبلی رسم کرده ایم استفاده کنیم.

بیشترین Correlation میان هدف ما با PT۰۸.S۲(NMHC) و PT۰۸.S۵(O۳) با اعداد ۰,۹۳ و ۰,۸۹ ووجود دارد.



تصویر ۲۴) میزان Correlation

بخش c

```
print(X_test)
```

```
[[ 0.45725876 -0.29267014  0.10989759 ...  0.37437449 -0.0904774
   0.20496752]
 [-2.1350422  -0.29267014  0.0743467  ...  0.30666793  0.20290566
   0.21097125]
 [ 0.4739999  -0.29267014  0.30895945 ...  0.35469908 -0.14124586
   0.20143072]
 ...
 [ 0.45339542 -0.29267014  0.08087125 ...  0.40215153  0.17019906
   0.21924787]
 [ 0.47142434 -0.29267014  0.23842855 ...  0.30840399 -0.32674597
   0.19171879]
 [-2.1350422  -0.29267014  0.17935534 ...  0.22275807  0.59782563
   0.21552985]]
```

تصویر ۲۵) داده های تست پس از اعمال StandardScaler

```
print(X_train[0])
```

```
[ 0.48430214 -0.29267014 -4.87857933 -3.19743969 -1.43196864 -3.09005116
 -2.03377286 -3.40640895 -2.57158213 -4.85581291 -4.67623536 -4.95611139]
```

تصویر ۲۶) خروجی نمونه پس از جداسازی داده آموزش و تست

بخش d

برای همه مقدار آلفا به شرح ذیل قرار داده شده است:

```
testAlpha = [0.001, 0.002, 0.005, 0.008, 0.01, 0.1, 0.2, 0.3, 0.5, 0.8, 1, 10, 100, 500, 1000]
```

تصویر ۲۷) مقدار لیست آلفا

برای رگرسیون الاستیک مقدار l1_ratio به صورت ذیل قرار داده شده است:

```
l1_ratios = [0.1, 0.3, 0.5, 0.7, 0.9]
```

تصویر ۲۸) مقدار لیست l1_ratios

Ridge

```
best_rmse_ridge: 0.21749328148119 61, best_alpha_ridge = 0.001
```

تصویر ۲۹) خروجی پس از آموزش و تست در Ridge

Lasso

```
best_rmse_lasso: 0.219029781019688 42, best_alpha_lasso = 0.001
```

تصویر ۳۰) خروجی پس از آموزش و تست در Lasso

Elastic

```
best_rmse_elastic: 0.218114784614622 08, best_alpha_elastic = 0.001, best_l1_ratio_elastic = 0.1
```

تصویر ۳۱) خروجی پس از آموزش و تست در Elastic

بخش e

با بهترین پارامتر های انتخاب شده هر سه مدل را آموزش دادیم و تست کردیم و معیار های خواسته شده را اعمال نمودیم:

```
Ridge Regression: RMSE = 0.22, R2 = 0.95
Lasso Regression: RMSE = 0.22, R2 = 0.95
Elastic Net Regression: RMSE = 0.22, R2 = 0.95
```

تصویر ۳۲) نتیجه اعمال معیار های R2 و RMSE

بخش f

سه مدل رگرسیون Ridge ، Lassoو Elastic Net با هدف پیش‌بینی دقیقتر یک متغیر وابسته در دسترس قرار گرفت. نتایج بررسی سه مدل به شرح زیر است:

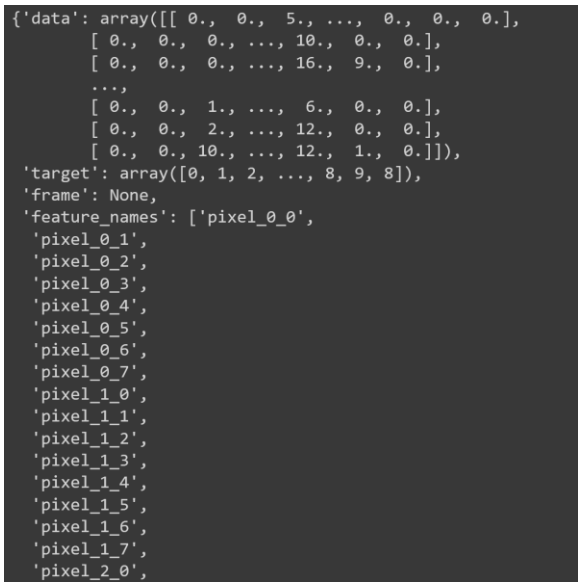
```
Ridge Regression: RMSE = 0.22, R2 = 0.95
Lasso Regression: RMSE = 0.22, R2 = 0.95
Elastic Net Regression: RMSE = 0.22, R2 = 0.95
```

با توجه به نتایج بدست آمده، مدل رگرسیون Ridge عملکرد بهتری نسبت به دو مدل دیگر از خود نشان داده است. در واقع، مقادیر RMSEو R2 برای این مدل به ترتیب کمتر و بیشتر از دو مدل دیگر است، که نشان دهنده دقت بیشتر و توانایی بیشتر مدل Ridge در پیش‌بینی متغیر وابسته است.

سوال چهارم

فایل کد در پوشه ی Source Codes با عنوان P4.ipynb قرارداده شده است.

بخش a



تصویر ۳۳) اضافه کردن دیتاست digits

```
X_train shape : (1257, 64), y_train shape : (1257,).
X_test shape : (540, 64), y_test shape : (540,).
```

تصویر ۳۴) تقسیم بندی داده ها به داده آموزش و تست

بخش b

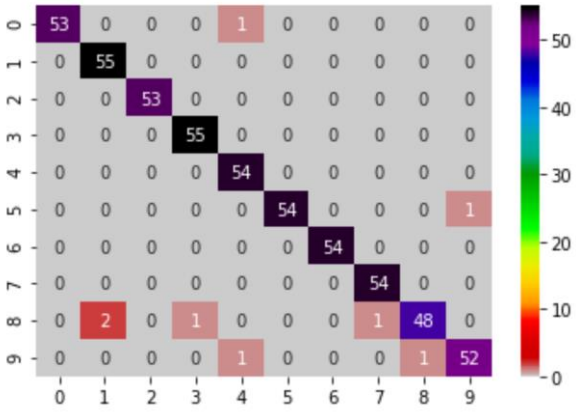
```
[[0.     0.     0.     ... 0.5    0.     0.    ]
 [0.     0.     0.4375 ... 0.     0.     0.    ]
 [0.     0.     0.0625 ... 0.4375 0.     0.    ]
 ...
 [0.     0.     0.     ... 0.625  0.     0.    ]
 [0.     0.     0.125  ... 0.5625 0.     0.    ]
 [0.     0.     0.0625 ... 0.75   0.0625 0.    ]]
```

تصویر ۳۵) داده های تست پس از اعمال min max scaler

بخش c

	precision	recall	f1-score	support
0	1.00	0.98	0.99	54
1	0.96	1.00	0.98	55
2	1.00	1.00	1.00	53
3	0.98	1.00	0.99	55
4	0.96	1.00	0.98	54
5	1.00	0.98	0.99	55
6	1.00	1.00	1.00	54
7	0.98	1.00	0.99	54
8	0.98	0.92	0.95	52
9	0.98	0.96	0.97	54
accuracy			0.99	540
macro avg	0.99	0.98	0.98	540
weighted avg	0.99	0.99	0.99	540

تصویر ۳۶) خروجی Classification_Report پس از آموزش و مدل سازی با ۵ همسایه



تصویر ۳۷) ماتریس Confusion

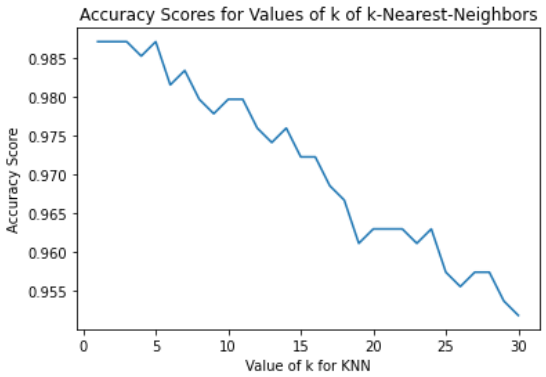
بخش d

```
Scores: [0.98888889 0.99444444 0.98333333 0.97777778 0.97777778 0.97777778
 0.98333333 1.         0.98882682 1.        ]
Mean accuracy: 0.9872160148975793
Accuracy standard deviation: 0.008257695334594247
```

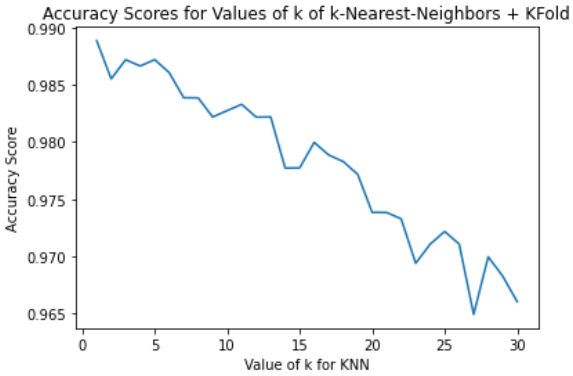
تصویر ۳۸) خروجی Score در k=10 روش KFold

بخش e

خب در این بخش قطعه کدی نسبتا طولانی نوشتیم که در آن بازه ی یک تا سی را برای k اختصاص دادیم و نتیجه صحت را در لیستی ذخیره نمودیم، الگوریتم را با این روش اجرا نمودیم و نتجه ذیل در قالب نمودار قابل دسترس است:



تصویر ۳۹) نمودار نسبت Accuracy نسبت به K بدون KFold



تصویر ۴۰) نمودار نسبت Accuracy به K با استفاده از KFold

سوال پنجم

فایل کد در پوشه ی Source Codes با عنوان P5.ipynb قرارداده شده است.

فایل کد من به دلیل آماده سازی برای آموزش در کلاس پایتون کامل تر از خواسته های این سوال می باشد و در بخش های این سوال صرفا پاسخ آن بخش ها آورده شده است.

بخش a

```
.. _iris_dataset:

Iris plants dataset
--------------------

**Data Set Characteristics:**

    :Number of Instances: 150 (50 in each of three classes)
    :Number of Attributes: 4 numeric, predictive attributes and the class
    :Attribute Information:
        - sepal length in cm
        - sepal width in cm
        - petal length in cm
        - petal width in cm
        - class:
                - Iris-Setosa
                - Iris-Versicolour
                - Iris-Virginica

    :Summary Statistics:

    ============== ==== ==== ======= ===== ====================
                    Min  Max   Mean    SD   Class Correlation
    ============== ==== ==== ======= ===== ====================
    sepal length:   4.3  7.9   5.84   0.83    0.7826
    sepal width:    2.0  4.4   3.05   0.43   -0.4194
    petal length:   1.0  6.9   3.76   1.76    0.9490  (high!)
    petal width:    0.1  2.5   1.20   0.76    0.9565  (high!)
    ============== ==== ==== ======= ===== ====================

    :Missing Attribute Values: None
    :Class Distribution: 33.3% for each of 3 classes.
    :Creator: R.A. Fisher
    :Donor: Michael Marshall (MARSHALL%PLU@io.arc.nasa.gov)
    :Date: July, 1988
```

تصویر ۴۱) بارگیری دیتاست iris از sklearn.datasets

```
X_train shape : (120, 4), y_train shape : (120,).
X_test shape : (30, 4), y_test shape : (30,).
```

تصویر ۴۲) جداسازی داده های آموزش و تست ۲۰/۸۰

بخش b

در این بخش ما داده های آموزش و تست را استاندارد میکنیم، خروجی بخشی از داده های آموزش در تصویر قابل مشاهده است.

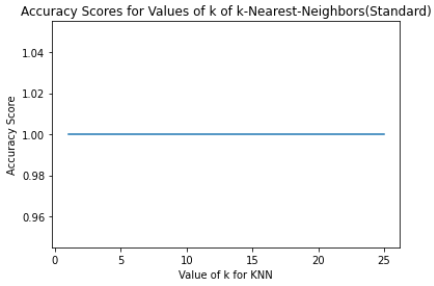
```
X_train

array([[-1.47393679,  1.20365799, -1.56253475, -1.31260282],
       [-0.13307079,  2.99237573, -1.27600637, -1.04563275],
       [ 1.08589829,  0.08570939,  0.38585821,  0.28921757],
       [-1.23014297,  0.75647855, -1.2187007 , -1.31260282],
       [-1.7177306 ,  0.30929911, -1.39061772, -1.31260282],
       [ 0.59831066, -1.25582892,  0.72969227,  0.95664273],
       [ 0.72020757,  0.30929911,  0.44316389,  0.4227026 ],
       [-0.74255534,  0.98006827, -1.27600637, -1.31260282],
       [-0.98634915,  1.20365799, -1.33331205, -1.31260282],
       [-0.74255534,  2.32160658, -1.27600637, -1.44608785],
```

تصویر ۴۳) خروجی داده های آموزش پس از استاندارد سازی

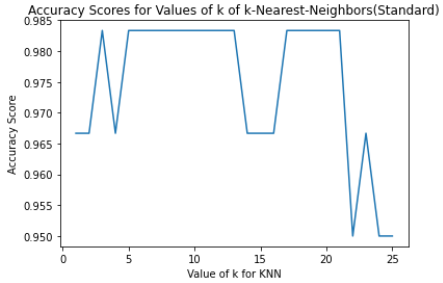
بخش C

خب آموزش و بررسی انجام شد و تماما پاسخ با حالت خواسته شده ۱ می باشد از این رو k را نمی توان یافت و مشاهده میکنیم که احتمالا ما دارای خطایی هستیم که یکی از دلایل آن کمبود تعداد داده ها می باشد، حال داده های آموزش و تست را به ۴۰/۶۰ تغییر می دهیم تا شاید تغییری حاصل شود و نتایج بهتری جهت نمایش بدست آید.



تصویر ۴۴) نمودار صحت در حالت خواسته شده

در ادامه فعالیتی انجام می گردد که خواسته نشده است.



تصویر ۴۵) نمودار صحت پس از تغییر نسبت داده های تست و آموزش

عدد ۳ را به عنوان k برگزیده انتخاب میکنیم و سپس آموزش می دهیم.

```
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X, y)
knn.predict([[6, 3, 4, 2]])

/usr/local/lib/python3.8/dist-packages/sklea
  warnings.warn(
array(['Versicolor'], dtype=object)
```

تصویر ۴۶) تصویری از کد به همراه پیش بینی