بسم الله الرحمن الرحیم

دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

فاز اول پروژه در مبانی هوش و کاربردها

پک‌من و رگرسیون خطی

استاد درس: دکتر حسین کارشناس

دستیار استاد: پوریا صامتی

دانیال شفیعی
مهدی مهدیه
سید امیررضا نجفی

آبان ۱۴۰۳

فهرست مطالب

فصل ۱

Pacman

فصل ۲

رگرسیون خطی

در این بخش از پروژه ما قصد داشتیم با داده‌های آموز مدلی را بسازیم که با آن داده‌های جدید را پیش‌بینی کنیم.

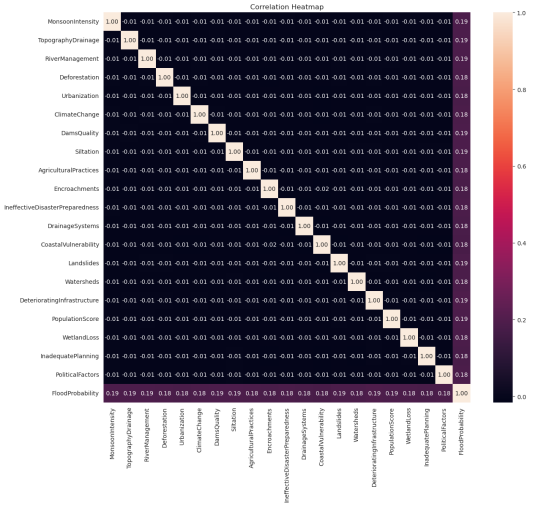
۱.۲ اکتشاف درون داده‌ها

با توجه به اینکه همه‌ی ستون‌های ما به جز id متغیرهای پیوسته بودند، ما می‌توانستیم به خوبی خود احتمال را که یک چیز پیوسته است پیش‌بینی کنیم. صرفا یک شک درمورد این وجود داشت که هاid در میان سطرهای یکی باشند. برای همین این را بررسی کردیم و چنین نبود. بنابرین متغیر id را که یک متغیر categorical بود حذف کردیم.

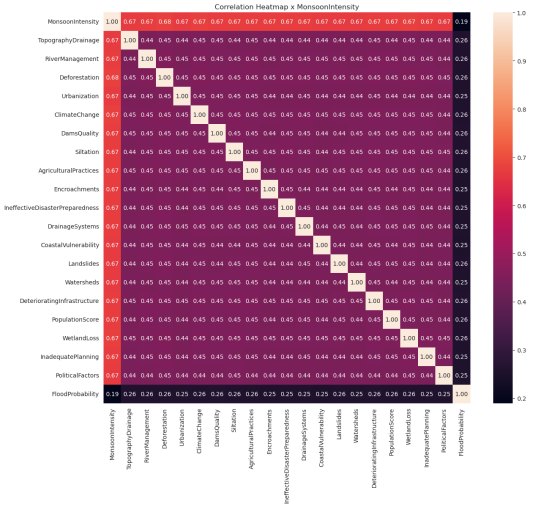
۱.۱.۲ پر کردن مقادیر خالی

داده‌ها هیچ مقادیر خالی نداشتند اما ما یک تابع knn imputer نوشتیم که به‌جای میانگین گرفتن از کل داده‌ها از k همسایه نزدیک (با توجه به بقیه‌ی ویژگی‌ها) میانگین می‌گیرد و مقادیر را پر می‌کند. فاصله در این تابع به صورت فاصله‌ی اقلیدسی تعریف شده است. در نهایت اما به علت پیچیدگی ما از یک تابع میانگین ساده روی کل مقادیر برای پر کردن داده‌ها استفاده کردیم.

۲.۱.۲ رسم نقشه گرمایی همبستگی



تقریبا هیچ همبستگی بین داده‌ها وجود ندارد تا آن را حذف کنیم. به عنوان یک پیشنهاد ما تست کردیم که اگر دو ستون را در هم ضرب کنیم چه اتفاقی برای همبستگی و همچنین میانگین مربع خطا می‌افتد برای همین در نوتبوک feature.ipynb این قضیه را تست کردیم. این کار را برای همه‌ی ستون‌ها تکرار کردیم و نتیجه همبستگی مقداری بهتر شد. بنابرین ما یک تابع جدید با ضرب بالاترین میانگین همبستگی ایجاد کردیم که البته در ادامه از این کار پشیمان شدیم! (چون در همه‌ی شرایط بدتر عمل می‌کرد)



۳.۱.۲ اسکیل کردن

در این قسمت ما از اسکیلر استاندارد و اسکیلر کمینه بیشینه استفاده کردیم.

۴.۱.۲ آماده‌سازی داده‌ها برای آموزش مدل

داده‌ها را به صورت ۴ مجموعه داده در حالت‌های مختلف با اسکیلرهای مختلف آماده کردیم و متغیرهای وابسته‌ی آن را از متغیر مستقل آن جدا کردیم. سپس یک بردار تصادفی از وزن‌ها (slope) و یک بردار تصادفی از مقادیر ثابت (intercept) آماده کردیم تا نتایج حاصل از مدل را در آن بریزیم.

۲.۲ مدل‌سازی داده‌ها و آموزش

در این قسمت ما به مدل‌سازی داده‌ها و پیش‌بینی می‌پردازیم.

۱.۲.۲ تابع رگرسیون خطی

تابع هزینه این تابع در اول کدنویسی بسیار ساده بود اما با اضافه شدن سایر ویژگی‌ها کمی به پیچیدگی آن افزوده شد. در این تابع تعدادی ایپاک از کاربر دریافت می‌شود. سپس مشتق خطای هر سطر نسبت به مقدار واقعی محاسبه می‌شود. تابع خطای ما در اینجا مشتق تابع میانگین مربع خطاست که ما را به نقطه‌ی بهینه هدایت می‌کند.

تکانه ما در اینجا یک velocity تعریف کردیم که برای اینکه تکانه واقعا تغییر کند، در اینجا به‌جای اینکه خطا مستقیما روی slope اثر بگذارد (با learning rate مشخص)، از سرعت قبلی هم کاملا اثر می‌پذیرد و این کمک می‌کند از داده‌های outlier کمتر اثربپذیریم و سریع‌تر به چیزی که می‌خواهیم میل کنیم.برای مثال با تکانه کم، ما حدود ۶۰۰ تکرار نیاز داشتیم تا زودهنگام به نتیجه برسیم اما با پیاده‌سازی تکانه این عدد به ۳۴۰ هزار رسید و زمان را به نصف کاهش داد!

توقف زودهنگام مسئله‌ی بعدی که قبل‌تر هم به آن اشاره شد خروج زود هنگام است. قبل از پیاده‌سازی این عملکرد، زمان اجرای یک دور تابع جبرخطی برای ۴ مجموعه داده حتی با وجود اینکه نمودارها جدا رسم می‌شدند برای ۱۰ ایپاک چیزی حدود ۶ دقیقه و نیم بود! اما با پیاده‌سازی این تابع، با حفظ عملکرد، تابع آموزش در کمتر از ۱۰ ثانیه هر ۴ مدل را آموزش می‌داد!

عادی‌سازی هرچند تکنیک عادی‌سازی L۲ در کد پیاده‌سازی شد که باعث می‌شود خودگردیان با مقادیر قبلی slope جمع شود ولی نتیجه‌های این اتفاق باعث می‌شد توابع underfit شود و اصلا به چیزی که می‌خواهیم شبیه نشود. یعنی با پیاده‌سازی این مکانیسم، امتیاز R۲ روی داده‌های تست به کمتر از ۰٫۸ می‌رسید که این از نظر ما مطلوب نبود. هرچند در کارکرد فلسفه‌ی آن هم همین است.

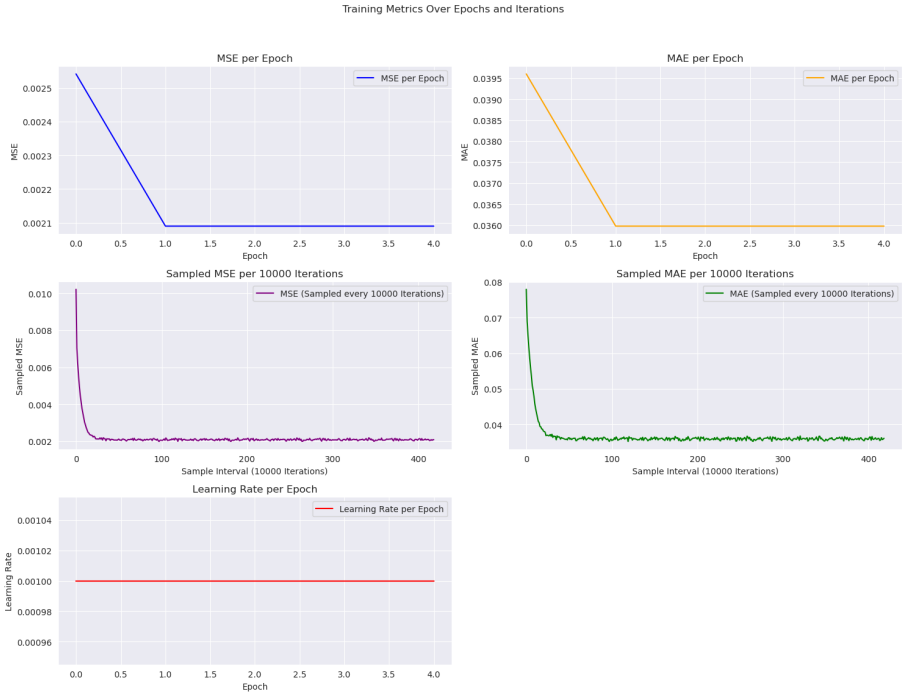
تغییر سرعت یادگیری ما از یک تکنیک ساده استفاده کردیم که در آن نرخ یادگیری بعد از هر ایپاک در یک مقدار کمتر از ۱ ضرب می‌شود. این با این فرض درنظر گرفته شده که مدل از جایی به بعد همگرا می‌شود. بیش از اندازه کوچک بودن این مقدار باعث می‌شود مدل underfit شود.

نمونه‌برداری سیاست ما این بود به‌جای مقایسه‌ی ایپاک‌ها برای توقف زودهنگام و همچنین به‌جای رسم همه‌ی تکرارها از نمونه‌برداری بین تعدادی سطر استفاده کنیم. بدین منظور میانگین خطای هر مثلا ۱۰۰۰ خط را اندازه‌گیری می‌کردیم و سپس بعد هر هزار بار آن را صفر می‌کردیم.

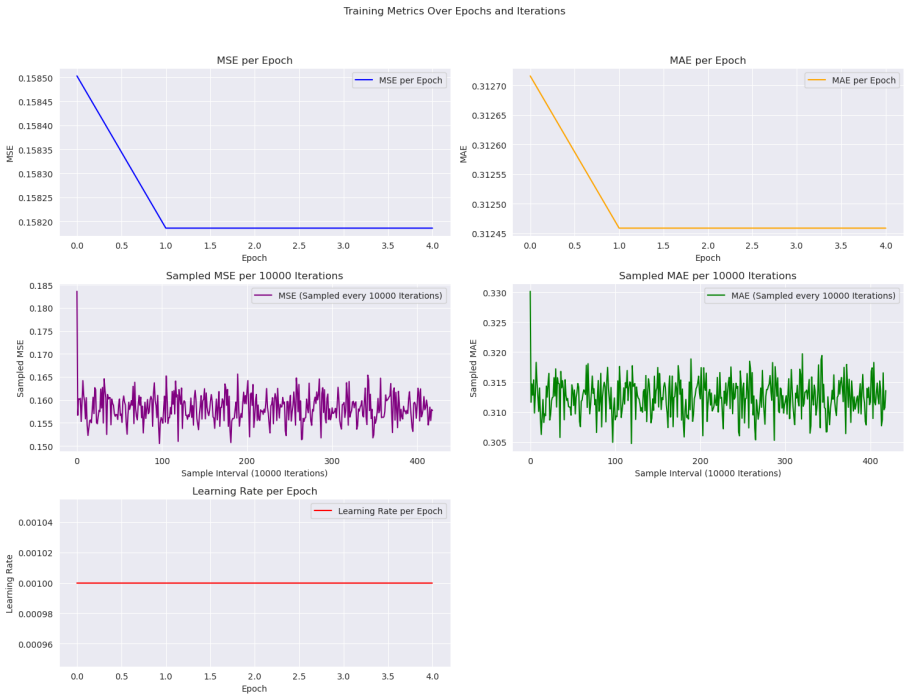
رسم تغییرات در اینجا ما کلیه تغییرات را بر حسب ایپاک و تکرار همچنین تغییرات سرعت یادگیری و خطا را نشان داده‌ایم.

۳.۲ پیش‌بینی

در این بخش، داده‌های آموزش و آزمون را به مدل می‌دهیم. تبعا به دنبال مدلی هستیم که روی هر دو داده‌ی آموزش و آزمون مقدار R^2 کافی برای ما فراهم کند.



شکل ۱.۲: آموزش داده‌های اسکیل شده با کمینه بیشینه در حالت پایه با ۵ ایپاک و سرعت آموزش ۰٫۰۱



شکل ۲.۲: آموزش داده‌های اسکیل شده با z در حالت پایه با ۵ ایپاک

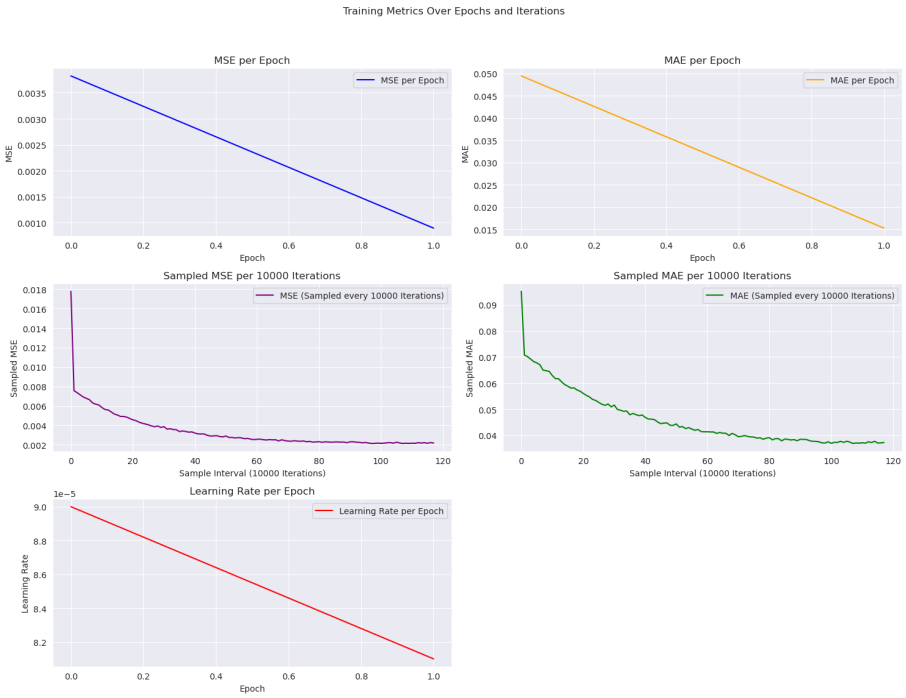
۱.۳.۲ پیش‌بینی روی داده‌های تست

برای داده‌های تست فرآیند مشابهی در زمینه‌ی اسکیل کردن طی می‌کنیم. در اینجا باید اشاره کنیم از همان اسکیلرهای آموزش استفاده می‌کنیم. سپس این داده‌ها را به مدل‌ها می‌دهیم و پیش‌بینی آن‌ها را می‌سنجیم. خروجی:

Mean Absolute Error: ۰.۳۱۴۲۹۴۳۰۷۸۰۴۶۲۱۸۶
Mean Squared Error: ۰.۱۵۹۹۰۹۹۷۲۴۴۵۹۶۵۹۷
R۲ Score: ۰.۸۳۹۷۱۸۵۸۹۸۸۷۸۴۹
Mean Absolute Error: ۰.۰۳۶۰۳۴۱۶۱۲۷۵۲۰۸۲۲۶
Mean Squared Error: ۰.۰۰۲۰۹۳۹۶۷۵۴۹۵۵۰۲۰۴۴
R۲ Score: ۰.۸۴۴۰۳۹۲۲۵۹۶۶۰۶۲۲
Mean Absolute Error: ۰.۳۱۲۲۹۳۵۷۷۰۴۲۷۶۰۸
Mean Squared Error: ۰.۱۵۷۸۴۸۹۴۲۸۱۰۸۱۹۵۴
R۲ Score: ۰.۸۴۱۷۸۴۴۰۶۸۶۶۹۳۵۹
Mean Absolute Error: ۰.۰۳۵۹۵۰۲۲۴۸۷۱۶۵۲۷۵
Mean Squared Error: ۰.۰۰۲۰۸۹۵۰۴۷۱۹۵۹۷۳۷۱
R۲ Score: ۰.۸۴۴۳۷۱۶۲۱۹۵۴۷۰۸۴

۲.۳.۲ بازی کردن با پارامترها

در نهایت ما می‌توانیم با بازی کردن با پارمترهای تابع رگرسیون در زمان کوتاه به توابع بسیار خوبی برسیم. نمونه‌ی این تابع:



شکل ۳.۲: Early stopping at iteration 1180000 with MSE: 0.0021721710452656936

که در آن پارامترها به نحو زیر تعیین شده بودند:

```
epochs_number=10, initial_learning_rate=0001.0,
momentum=5.0, patience=20, regularization_param=0.0,
lr_decrease=90.0, iteration_sample=10000)
```

که درظرف ۵ ثانیه آموزش مدل به نتایج زیر رسیدیم:

Mean Absolute Error: ۰.۰۳۷۰۹۵۱۷۱۴۵۷۰۳۲۴
Mean Squared Error: ۰.۰۰۲۱۵۴۵۹۴۱۴۰۳۷۵۴۶۳۶
R۲ Score: ۰.۸۳۹۵۲۳۶۹۷۵۲۱۴۲۲۶