سوال ۱_قسمت ج) مدل ساخته شده قیمت را حدودا ۱۰۵۶۷۴۳۶۹ تومان پیش بینی میکند که خیلی بیشتر از قیمت ارائه شده توسط فرد است و فرد قیمت گذاری پایینی را در نظر گرفته است.

سوال ۱_قسمت د: ابتدا برای روش BGD نرخ یادگیری های مختلف ۰.۰۱ و ۰.۱ و ۰.۰۵ و ۰.۰۰۱ را امتحان کردیم و میزان خطای نهایی هر کدام را بررسی کردیم و بنظر رسید نرخ یادگیری ۰.۰۱ برای این روش مناسبتر باشد. برای روش sgdنیز با همین مقادیر الفا امتحان کرده و بنظر میرسد ۰.۰۱ مقدار بهتری باشد. از طرف دیگر تعداد دفعات ایپاک را ۱۰۰ و ۱۰۰۰ امتحان کردیم و بنظر میرسد تعداد ۱۰۰ تا بهتر باشد. برای روش minibatch نیز با نرخ های متفاوتی امتحان کرده و بنظر میرسد الفا ۰.۷ مقدار مناسبی باشد و با تعداد ایپاک های ۱۰۰ و ۱۰۰۰ نیز امتحان کردیم و بنظر میرسد تعداد ایپاک ۱۰۰۰ مناسب تر باشد.

برای مقایسه بین این سه روش حالت بهینه هر کدام از روش ها با نرخ یادگیری و تعداد ایپاک مشخص را بدست آورده. با توجه به نمودار minibatch loss vs cost سریعتر از مابقی روش ها نقطه مینیمم سراسری را یافته است. و روش Gd کمترین سرعت همگرایی را داشته است چون دیرتر از مابقی به نقطه مینیمم گراییده شده است. اما روش sgd دیرتر از روش مینی بچ نقطه مینیمم را یافته است. با توجه به اینکه sgd در هر بار اجرا فقط با دیدن یک سمپل آپدیت میشود نویز زیادی دارد در نتیجه به دلیل این نویز دیرتر نقطه مینیمم سراسری را یافته است اما چون روش مینی بچ تعداد سمپل بیشتری را برای یک آپدیت در نظر میگیرد نویز کمتری داشته و سریعتر همگرا میشود. روش gd نیز چون در هر ایپاک تمامی داده را یکجا مشاهده میکند نویز ندارد و بنظر میرسد در iterationهای پایانی به نقطه مینیمم گراییده میشود. از نظر کمینه خطا minibatch در نهایت به خطای بسیار کوچکتر از مابقی گراییده شده و sgd از نظر کمینه خطا بینابین این دو روش است و gd کمینه خطای بسیار بزرگتری دارد. نویز زیاد sgd مانع از داشتن کمترین مقدار خطا میشود. چون به هر حال با دیدن هر سمپل جداگانه ای آپدیت انجام میشود که این کار میتواند خطای بسیار زیادی ایجاد کند. اما در مینی بچ نویز به این شدت نبوده پس کمینه خطای کمتری دارد.

سوال ۱_قسمت ه) نرخ یادگیری را ابتدا افزایش داده برابر با ۰.۹ قرار دادیم و مجددا در این حالت ترین را انجام داد و نمودارهای cost vs iteration هر دو را تحلیل کردیم.(نمودار های قرمز رنگ مربوط به آلفای جدید ۰.۹و نمودار های سبز رنگ آلفاهای اولیه هر روش است. از روی این نمودارها متوجه میشویم که در هر سه روش نمودار های قرمز رنگ سریعتر نقطه مینیمم را پیدا کرده اند و زودتر همگرا شده اند. پس با افزایش نرخ یادگیری تا یک محدوده مشخص همگرایی سریعتر شده و زودتر به نقطه منیمم بهینه میرسیم. اما باید توجه کرد افزایش نرخ یادگیری تا یک جایی سرعت همگرایی را افزایش میدهد و از آن به بعد

میتواند موجب رد کردن نقطه بهینه شود و کلا به بیراهه برود. (به نظر میرسد در این حالت SGD نقطه بهینه را رد کرده و cost آن بعد از مدتی رو به افزایش میرود.)

در مرحله بعدی نرخ یادگیری را خیلی کوچ کرده و برابر ۱e-۸ قرار دادیم. نمودار های قرمز رنگ مربوط به این نرخ یادگیری و نمودار های سبز رنگ مربوط به نرخ یادگیری اولیه هستند. با تحلیل متوجه میشویم نمودار های قرمز رنگ در هر سه روش هنوز به نقاط مینیمم بهینه همگرا نشدند و نیاز به iteration های خیلی بیشتری دارند تا همگرا شوند. پس شرعت همگرایی در این نمودارها شدیدا کم است و نمودار ها یا اصلا شیب ندارند یا شیب خیلی کمی دارند.

سوال ۱_ قسمت و) میزان خطا در روش پیاده سازی با کتابخانه آماده یک عدد ۸ رقمی معادل ۳۲۱۶۴۸۹۷ است در حالی که میزان خطای مدل ساخته شده خودمان عدد ۱۲ رقمی معادل ۱۶۰۸۷۸۴۴۶۲۰۴ است و میازن خطا توسط مدل کتابخانه آماده خیلی پایین تر است.

بنظر میرسد سرعت همگرایی آن نیز بالاتر باشد. سرعت همگرایی مدل آماده ۸۹۱۵۹ میلی ثانیه در حالی که سرعت مدل قسمت ب حدود ۱ ثانیه و ۲۶۲۵۴ میلی ثانیه است.

سوال ۲ _قسمت ب) بنظر میرسد خطا روی داده های آموزشی حدودا ۷۴٪ اما روی داده های تست حدودا ۶۸٪ است. هر دو خطا بالا هستند و این گونه نیست که تفاوت زیادی بین خطای ترین و تست وجود داشته باشد (درواقع این گونه نیست که خطای ترین خیلی پایین باشد و خطای تست بالا یاشد) بلکه تفاوت کمی بین هر دو خطاست و هر دو خطا بالا میباشد. در نتیجه میتوان گفت مدل underfitشده است یعنی حتی داده های ترین را نیز به خوبی یاد نگرفته است و نیاز دارد تا پیچیده تر شود. این در صورتی است که نرخ یادگیری برابر ۰.۱ در نظر گرته شود. اگر این مقدار کمتر در نظر گرفته شود(مثلا ۰.۰۰۱) مشاهده میکنیم که خط برازش شده در این حالت نیز underfit بوده و حتی وضعیت بدتری نسبت به خط برازش شده با نرخ ۰.۱ دارد. و این خط حتی در فاصله زیادی نسبت به توده داده ها قرار گرفته است.

سوال ۲_قسمت ج) بنظ میرسد وزن های بدست آمده از روش بسته نیز مطابق با راه حل BGDاما با نرخ یادگیری (۰.۱) است. و به تبع آن خطای آموزش و تست بدست آمده از مدل بسته نیز مطابق با خطای آموزش و تست مدل BGD با نرخ یادگیری ۰.۱ است. اما اگر نرخ یادگیری BGD مقادیری کمتر از ۰.۱(مثلا ۰.۰۰۰۱ یا ۰.۰۰۱ ...) انتخاب شود راه حل بسته نتایج بهتر خواهد داشت و خطای کمتری روی ترین و تست خواهد داشت. (میزان خطا در این حالت در راه حل بسته کمتر است). پس نتیجه میگیریم اگر هایپر پارامتر نرخ یادگیری به درستی و دقیق برای روش BGD تنظیم نشود خطای بیشتری نسبت به راه

حل بسته خواهد داشت ولی اگر این هایپر پارامتر به درستی تنظیم شود نتایج هر دو تقریبا یکسان است و از نظر میزان خطا نیز هر دو یکسان خواهند بود. پس در واقع روش BGD یک بار اضافه تری داشته که تنظیم هایپر پارامتر نرخ یادگیری است.

اما از نظر سرعت همگرایی بنظر میرسد راه حل بسته سرعت بیشتری داشته باشد. چون روش گرادیان یک روش تکراری است که آرا آرام و گام به گام به سمت نقطه بهینه حرکت میکند و هر گام از اردر mn است و چون ۱۰۰۰ گام در این سوال داشتیم و ۳۲ سمپل آموزشی و ۲فیچر داشتیم از اردر ۱۰۰۰ * ۳۲ *۲ خواهد شد. ولی در روش بسته چون معکوس گرفتن از ماتریس n*nاز اردر n^۳ است و n در این سوال برابر ۲ است پس از اردر ۲^۳ بوده که مقدار خیلی کوچکتری نسبت به گرادیان است و در واقع چون دیتاست کوچک است حجم محاسبات نیز کمتر بوده و سریعتر طی یک گام همگرا میشود به نقطه بهینه. از نظر زمان اجرای این دو روش نیز راه حل بسته زمان اجرای کمتری داشت. زمان اجرای روش گرادیان ۱۷۹۱۶ میلی ثانیه درحالی که زمان اجرای روش بسته ۹۵۹ میلی ثانیه بود.

سوال ۲_قسمت د) با اضافه کردن توان ها نسبت به قسمت قبل(که خطای روی ترین و تست نسبتا زیاد بود) خطای روی ترین و تست کاهش شدیدی داشته و از اعداد تقریبا ۷۰٪ به ۴ الی ۵٪ رسیده است. ابتدا با درجه ۵ خطای روی ترین و تست هر دو پایین بوده و اختلافشان نیز کم است(۴.۲ و ۴.۶). با اضافه کردن درجه به ۷ خطای روی ترین کاهش یافته اما خطا روی تست افزایش میبابد که این نشانه خوبی نیست و درواقع مدل در این شرایط به سمت overfit شدن میرود. با افزایش درجه به ۹ تقریبا همان نتایج با درجه ۷ را خواهیم داشت. با افزایش درجه به ۱۱ خطی ترین مجددا نسبت به درجه ۵کاهش یاقته اما خطای تست نسبت به درجه ۵ فزایش داشته است. که باز هم نشانه خوبی نبوده و میتواند باعث overfit شدن مدل شود. با افزایش درجه به ۱۳ نیز همین اتفاق افتاده است. اما با افزایش درجه به ۱۵ اختلاف بین خطای تست و ترین خیلی بیشتر شده و خطای ترین به ۲ درصد رسیده درحالی که خطای تست حدود ۱۷٪ است و با کاهش خطای ترین و افزایش شدید خطای تست مواجه هستیم که نشانه overfitشدن مدل است. از روی منحنی مدل نیز این امر مشهود است. چون بنظر میرسد با افزایش درجه منحنی رفته رفته تمایل به چسبیدن به داده های ترین دارد و در منحنی با درجه ۱۵ این امر شدیدتر است.پس بنظر میرسد پیچیدگی زیاد مناسب نبوده و درجه ۵ نتیجه مناسبی دارد.

سوال ۲ _قسمت ه) با توجه به اینکه Ridge سعی میکند از overfit شدن مدل جلوگیری کند و در واقع در صورت نیاز وزن جملات با درجات بالا را کم کند یا حتی به صفر برساند از روی نمودارهای برازش شده درجه ۱۳ و ۱۵ و مقایسه آنها با نمودارهای نظیرشان در قسمت قبل متوجه میشویم که تا حدودی جلوی

overfitشدن این نمودارها را گرفته است. چون نمودارها در این قسمت پیچیدگی و پیچ و تاب و خمیدگی کمتری داشته و کمتر تمایل به چسبیدن به داده های ترین را دارند. از طرف دیگر خطای روی داده های تست نیز نسبت به قسمت قبلی کمتر شده است. مثلا درحالت درجه ۱۵ خطای تست از ۱۷٪ در قسمت قبلی به ۳.۸٪ و درحالت درجه ۱۳ از ۵٪ به ۳.۷٪ رسیده است. درحالت درجه ۱ نیز از ۶۸٪ به ۶۷٪ رسیده است. پس با توجه به regularization انجام شده روی داده ها از overfit شدن مدل جلوگیری شده در نتیجه خطاهای روی تست نیز کاهش یافته است.

سوال ۲_قسمت و) مشاهده میکنیم روش lasso در حالت درجه ۱۳ و ۱۵ فقط دو تا از وزن ها را در نهایت مقدار غیر صفر کرده است و مابقی وزن ها صفر شده اند. منطقی است چون lasso به دلیل شکل لوزی که پیدا میکند به سمت صفر کردن ضرایب میرود. برای همین برای انتخاب فیچر میتواند مورد استفاده قرار گیرد. روش ridge هم وزن ها را به نحوی جهت داده است که overfit شدن کاهش یابد اما ضرایب را صفر نکرده است. برخی ضرایب نزدیک به صفر رسده اند اما صفر نشده اند. منطقی است چون ridge به دلیل شکل دایره ای که پیدا میکند به سمت صفر کردن ضرایب نمیرود. در روش نرمال نیز چون نمودار با درجات ۱۳ و ۱۵ به سمت overfit شدن میرفت پس ضرایب در روش ridge که از overfit شدن جلوگیر ی میکند تغییرات زیادی داشته و برخی ضرایب خیلی بزرگتر و برخی کوچکتر شده و نزدیک صفر رسیدند تا از چسبیدن نمودار به داده های ترین جلوگیری شود.