گزارش پروژه دوم یادگیری ماشین

بخش 1:

در این بخش در حالت 1 خواسته شده از همان مجموعه داده پروژه قبلی استفاده شود و رگرسیون با k-fold cross validation روی داده ها اعمال شود.
در اول کار طبق همان پروژه اول داده های خالی ، پر شدند و normalize شدند و... .

سوال 1:

اما برای بخش بندی داده ها برای 5fold چون حدود 250000 داده داشتیم برای هر fold ما 50000 داده در نظر گرفتیم سپس مدل رگرسیون تک فیچری که در پروژه قبل پیاده سازی شده بود را روی 4 fold یادگیری انجام میداد و روی fold دیگر باقی مانده تست میشد که چه مقدار درستی دارد و MSE که میزان خطا است چقدر است و باز با جایگشتی متفاوت نسبت به اولی روی 4 fold دیگر یادگیری انجام شد و تست روی fold باقی مانده که درکل 5 بار تکرار میشود .در این مدل تک فیچر ما livingSpaceRange بود که correlation بالایی با تارگت داشت.سپس دقیقا همین مراتب با 10fold انجام شد که تعداد داده هر fold هم 25000 داده بود.در 5fold درستی حدود 0.90 و MSE حدود 98.3 در 10fold هم درستی حدود 0.90 و MSE حدود 98.3 داشتیم.البته این هم شاید لازم به ذکر باشد که در مدل رگرسیون ما نرخ یادگیری 0.01 و تعداد epoch = 5000 بود

حالت 2:

دقیقا همین حالت قبلی بود با این تفاوت که مدل رگرسیون را با پکیج sklearn.linear_model پیاده سازی شد و cross validation هم از پکیج sklearn.model_selection متود cross_val_score استفاده شد که در 5fold درستی حدود 0.90 و MSE حدود 92.8 در 10fold هم درستی حدود 0.90 و MSE حدود 92.8 داشتیم.

حالت 3 :

باز دقیقا مثل حالت 2 است با این تفاوت که 2 فیچر با بیشترین corr و 2 فیچر با کمترین corr به مدل دادیم و این دفعه در 5fold درستی حدود 0.91 و MSE حدود 88.4 در 10fold هم درستی حدود 0.91 و MSE حدود 88.4 داشتیم.

حالت 4 :

رگرسیون با استفاده از پکیج رو برای فیچر های دلخواه پیاده سازی کردیم که در 5fold درستی حدود 0.93 و MSE حدود 68.2 در 10fold هم درستی حدود 0.93 و MSE حدود 67.7 داشتیم.

حالت 5 :

رگرسیون Ridge با استفاده از پکیج sklearn.linear_model رو برای فیچر های دلخواه پیاده سازی کردیم که در 5fold درستی حدود 0.93 و MSE حدود 68.1 در 10fold هم درستی حدود 0.93 و MSE حدود 67.7 داشتیم.

حالت 6 :

رگرسیون Lasso با استفاده از پکیج sklearn.linear_model رو برای فیچر های دلخواه پیاده سازی کردیم که در 5fold درستی حدود 0.91 و MSE حدود 86.2 در 10fold هم درستی حدود 0.91 و MSE حدود 86.2 داشتیم.

سوال 2 :

در جواب باید گفت که بر روی این داده ها رگرسیون Ridge هم درستی بهتری داشت و هم MSE کمتری داشت هم در 5fold و هم 10fold.

بخش 2:

سوال 1:

همونطور که میدونیم در رگرسیون خطی ما میخوایم تابع خطا رو کمینه بکنیم و ضرایبی رو باید انتخاب کنیم که خطی بر اطلاعات fit کند که خطا کمینه باشد و از طرفی باید مدل ما پیچیدگی هم نداشته باشد از راه های کم کردن پیچیدگی مدل

regularization regression است که می آید یک جمله به تابع هزینه ما اضافه میکنه که اصطلاحا میگن پنالتی میده که درواقع تابع هزینه ما را جریمه میکند.
اما تفاوت ridge و lasso در اون ترمی هست که اضافه میشه به تابع هزینه

$$CostFunction = OLS + regularization\ term(Penalty)$$

$$Penalizing\ large\ coefficients\ =\ Penalizing\ Overfitting$$

$$ridge\ regression\ lost\ function = OLS_{(ordinary\ Least\ of\ square)} + \alpha * \sum_{i=1}^{n} {a_i}^2$$

$$Lasso\ regression\ lost\ function = OLS_{(ordinary\ Least\ of\ square)} + \alpha * \sum_{i=1}^{n} |a_i|$$

از تفاوت هایی که میشود گفت باز این است که رگرسیون lasso تمایل دارد ضرایب را به 0 مطلق برساند برخلاف ridge که هرگز ضریب را به 0 نمیرساند.پس در کل lasso میتواند پیچیدگی کمتری داشته باشد.
سوال 2:
از راه هایی که میشود بهترین ضریب رو پیدا کرد و کلا برای پیدا کردن هایپر پارامتر ها میتوان استفاده کرد روش greed search است که میاییم ضرایب مختلفی را امتحان می کنیم و جداگانه اطلاعات را به مدل fit میکنیم تا بررسی کنیم روی کدام ضریب بهترین عملکرد را مدل ما دارد و بهترین ضریب رو به این شکل انتخاب میکنیم
سوال 3:
در کل افزایش تعداد fold نه خوب می تواند باشد و نه بد بلکه بستگی به تعداد داده هایی که ما داریم دارد معمولا k = 10 رو همه جا استفاده میکنن و نتایج بهتری ازش بدست می آید و اگر کمتر از این مقدار باشد ممکن است بهترین عملکرد مدل رو بدست نیاوریم و از طرفی اگر بیشتر از این مقدار باشد ممکن بار محاسباتی و هزینش زیادتر خواهد شد و احتمال overfit شدن هم ممکن است بوجود بیاید.
سوال 4:

روشی است که مثل kfold cross validation عمل میکند فقط تعداد fold هاش با تعداد کلاس های y برابر است و عملکردش روی fold ها مثل همان kfold است که روی k-1 fold یادگیری انجام میشود و روی fold باقی مانده ارزیابی میشود و به تعداد k این روش تکرار میشود با جایگشت های متفاوت.
سوال 6:
در این روش ما 5 بار 2fold cross validation رو انجام میدیم

بخش 3:
برای این بخش اما مجموعه داده متفاوتی نسبت به بخش 1 داشتیم که مجموعه مشخصات موبایل های مختلف بود.دوباره شروع به مرتب سازی داده ها کردیم ولی در کل مجموعه داده ی بسیار تمیزی بود و داده null نداشت و حتی outlier نداشتیم و همه داده ها هم داده های عددی بودند که این هم خودش راحتی کار با این مجموعه داده بود برخلاف مجموعه داده بخش 1.
سوال 1 :
از پکیج sklearn.linear_model رگرسیون لجستیک را فراخوانی کردیم و روی داده های train که با train_test_split از داده های test جدا کرده بودیم یادگیری رو انجام دادیم سپس مدل را روی داده های xtest پیشبینی کردیم تا با مقایسه این جواب با ytest واقعی دقت مدل را ارزیابی کنیم .در ادامه با استفاده از پکیج sklearn.metrics و متد confusion_matrix این ماتریس را دریافت کردیم و با متد classification_report به سوالات مربوط به این بخش درباره precision و recall و f1-score جواب داده شد که برای هر کلاس متفاوت ولی به طور کلی حدود 0.91 برای همه بود.
سوال 2 :
جواب این سوال رو هم با کد پیاده سازی کردیم که جوابش بله است یعنی تعداد داده های 4 کلاس متوازن است و هر کلاس 500 داده دارد
سوال 3 :

داده هایی که لیبل 2 یا 3 داشتند رو به 1 تغییر دادیم و برچسب های unique داده های ما شد 0 و 1 یعنی 2 کلاس فقط داریم.

سوال 4:

روی کلاس بندی جدید داده هایمان با پکیج رگرسیون لاجستیک زدیم و روی xtest پیش بینی انجام دادیم و باز با classification_report اطلاعات مهم رو بدست اوردیم precision حدود 0.98 و recall حدود 0.96 و f1-score حدود 0.98 داشتیم

سوال 5 :

جواب بله است داده های ما نامتوازن است در کلاس 0 ،500 داده داریم و در کلاس 1 ، 1500 داده که برای متوازن کردن داده ها 3 راهی که میشود انجام داد اولی این است که به شکل random بیایم داده هایی که برچسب 1 دارند رو حذف کنیم تا جایی که تعداد داده های کلاس ها یکی بشه که ما در کد هم همین روش رو پیاده سازی کردیم.روش دوم این است که تعداد داده های کلاس 0 را به شکل random زیاد کنیم تا با کلاس 1 یکی شوند.روش سوم هم این است که از شبکه های عصبی GAN استفاده کنیم برای تولید داده های جدید برای کلاس 0 که تعداد داده های آن با کلاس 1 یکی شود.اما بعد استفاده از روش اول برای متوازن سازی کلاس ها رگرسیون لاجستیک رو روی داده های train اجرا کردیم و سپس روی test ارزیابی کردیم و باز با classification_report اطلاعات رو بدست اوردیم precision حدود 0.98 و recall حدود 0.98 و f1-score حدود 0.97 داشتیم

سوال 6:

انتخاب ویژگی forward selection رو از پکیج mlxtend.feature_selection و متد SequentialFeatureSelector استفاده کردیم و با انتخاب 10 ویژگی بر اساس معیار auc که مساحت زیر نمودار roc curve است.

سوال 7:

در این سوال هم مدل رگرسیونی لاجستیک رو فیچر های انتخابی پیاده کردیم که چون ما ویژگی ها را کاهش دادیم ارزیابی مدل اصلا خوب نشد و precision حدود 0.3 و recall حدود 0.31 و f1-score حدود 0.32 داشتیم .

سوال 8 و 9:

هم مثل سوال قبل کاهش فیچر داشتیم اما با الگوریتم pca با همان تعداد فیچر 10 تایی که در سوال قبلی رگرسیون رو روی این داده های اعمال کردیم که این هم مدل خوبی نشد و ارزیابی مدل به این صورت شد که precision حدود 0.32 و recall حدود 0.32 و f1-score حدود 0.33 داشتیم .

سوال 10:

دقیقا مثل سوال 6 و 7 با این تفاوت که backward selection استفاده شد precision حدود 0.3 و recall حدود 0.31 و f1-score حدود 0.32 داشتیم .

سوال 11:

اما در سوال آخر بر روی تمامی فیچر ها 5fold و 10fold cross validation را با کمک پکیج زدیم که در 5fold درستی حدود 0.92 در 10fold هم درستی حدود 0.92 داشتیم.

بخش 4:

سوال 1:

در کل default رگرسیون لجستیک روی 2 کلاس باینری است ولی اگه با تعداد کلاس بیشتر هم باهاش کار کنیم عملکردش اینطوری هست که میاد اون ستون رو به چند ستون باینری تبدیل میکنه و باهاش کار میکنه.

سوال 2:

در این مدل خاص تفاوت زیادی حاصل نشد و درستی سوال 5 کمتر هم شد حتی ولی به طور کل balance کردن داده ها و کار روی این دیتاست عملکرد بهتری خواهد بود تا کار بر روی یک دیتاست imbalanced.

سوال 3:
بله در مدلی که زده شد تفاوت محسوسی و چشمگیری دیده شد در سوال 6 درستی ما به 0.32 رسید ولی در حالت اولیه درستی 0.91 بود اما از طرفی هم این feature selection فرآیندی است که موجب کاهش تعداد متغیرهای ورودی هنگام ایجاد یک مدل پیش بینی . کاهش تعداد متغیرهای ورودی برای کاهش هزینه محاسباتی مدل سازی و در بعضی موارد هم برای بهبود عملکرد مدل مطلوب است.
سوال 4:
یکی دیگر از فرایندهای انتخاب ویژگی میتواند به این شکل باشد بررسی کنیم ارتباط بین ویژگی هایمان با تارگت و مثلا اگر 5 ویژگی را میخواهیم انتخاب کنیم 5 ویژگی را انتخاب کنیم که بالا ترین ارتباط را با تارگت دارد.
یکی دیگر فرایند حذف بازگشتی ویژگی است که یک روش «حریصانه» برای انتخاب ویژگی است. در این روش، ویژگی‌ها به طور بازگشتی و با در نظر گرفتن مجموعه‌های کوچک و کوچک‌تر از ویژگی‌ها (در هر مرحله) انتخاب می‌شوند. در این روش، ویژگی‌ها بر اساس مرتبه حذف شدن آن‌ها از فضای ویژگی رتبه‌بندی می‌شوند.

سوال 5:
در کل feature selection الگوریتم خیلی قدرتمندی نیست چرا که ما بخشی از داده هایمان را ازبین می بریم و در هر مرحله که فیچری اضافه میشه چه به صورت پیشرو و چه پسرو ممکنه فیچر های خوبی انتخاب نشن و فیچر های مهم حذف شوند چرا که فیچر ها در تعامل با هم مشخص میشود که مناسب هستند یا خیر مثلا ممکن است براساس معیار دو فیچر نزدیک بهم باشند و ما فیچری از این 2 را انتخاب کنیم که اگر دیگری را انتخاب می کردیم عملکرد بهتری برای مدل ما حاصل میشد.