مبانی یادگیری ماشین

گزارش تمرین ۳ و ۴

عرشیا حسینمردی

شماره دانشجویی : 98222030

قسمت اول سوالات :

بخش 1)

در بخش پیاده سازی backward selection و forward selection ، تابع LR مربوط به اجرای logistic regression است که معیار AUC را بسنجیم و دو تابع forward و backward وجود دارد که از تابع LR استفاده میکند تا فیچر ها را انتخاب کند (در قسمت کد مشخص است)

بخش 2)

نتایج f1-score ، recall ، precision برای 3 حالت backward selection و forward selection و پکیج لجستیک به شکل زیر است:

```
Accuracy of sklearn's Logistic Regression Classifier with forward selectopn: 0.9725, acc_auc: 0.9725465090005272
              precision    recall  f1-score   support

           0       0.97      0.97      0.97       187
           1       0.98      0.97      0.97       213

    accuracy                           0.97       400
   macro avg       0.97      0.97      0.97       400
weighted avg       0.97      0.97      0.97       400
```

```
Accuracy of sklearn's Logistic Regression Classifier with backward selectopn: 0.9875, acc_auc: 0.986957394993849
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       187
           1       0.98      1.00      0.99       213

    accuracy                           0.99       400
   macro avg       0.99      0.99      0.99       400
weighted avg       0.99      0.99      0.99       400
```

```
0.893
              precision    recall  f1-score   support

           0       0.91      0.88      0.89      1000
           1       0.88      0.91      0.89      1000

    accuracy                           0.89      2000
   macro avg       0.89      0.89      0.89      2000
weighted avg       0.89      0.89      0.89      2000
```

بخش 3 و 4)

الگوریتم PCA در قسمت کد پیاده سازی شده و نتایج f1-score و ... به صورت زیر است :

```
score of the model is :
0.6325
              precision    recall  f1-score   support

           0       0.55      1.00      0.71        99
           1       0.89      0.16      0.28        97
           2       1.00      0.25      0.40        88
           3       0.64      1.00      0.78       116

    accuracy                           0.63       400
   macro avg       0.77      0.60      0.54       400
weighted avg       0.76      0.63      0.56       400
```

بخش 6)

مهندسی ویژگی در قسمت کد این بخش به صورت کامل نوشته شده است و ترجیح داده شده که در این قسمت چیزی نیاید.

بخش 7)

در قسمت آ روش train score SVM برابر 0.80 میشود.

در قسمت ب چون one hot encoding از قبل انجام شده و داده های categorical به صورت کامل هندل شده است حالت ساده را تست میکنیم که میشود 0.95 میدهد .

در قسمت ج به ما 0.958 میدهد.

در قیمت د هم به ما 0.95 میدهد .

و اگر از همه استفاده کنیم به ما امتیاز 0.96 میدهد

بخش 8)

به طور خلاصه cross validation مجموعه داده موجود را برای ایجاد مجموعه داده های متعدد تقسیم می کند و روش bootstrapping از مجموعه داده اصلی برای ایجاد مجموعه داده های متعدد پس از نمونه برداری مجدد با جایگزینی استفاده می کند.به طور خلاصه cross validation مثل بر زدن داده ها میماند ، ما داده را k بخش تقسیم میکنیم و هر بار یکی از بخش ها مثل داده test عمل میکند . bootstrapping مثل این میماند که ما n فیچر داریم که در کیسه قرار دارد و ما هر بار دست در این کیسه میکنیم و یک فیچر انتخاب میکنیم و دوباره به کیسه برمیگردانیم و از این طریق b مدل میسازیم.

هدف از bootstrapping بیشتر در مورد ساخت مدل مجموعه یا تخمین پارامترها است.

بخش 9)

منظور این است که 5 بار متوالی 2-fold را انجام میدهیم یعنی 5 بار داده ها را به دو بخش تقسیم میکنیم

از cross validation در بخش model selection از آن استفاده میکنیم و همینطور عمدتاً در یادگیری ماشینی کاربردی برای تخمین مهارت یک مدل یادگیری ماشینی بر روی داده های دیده نشده استفاده می شود.

بخش 10)

معمولا با افزایش پیچیدگی مدل بایاس کاهش و واریانس افزایش میابد و ما میخواهیم بهترین مرتبه برای پیچیدگی مدل را پیدا کنیم که اگر بخواهیم از elbow method استفاده کنیم بستگی به بزرگی یا کوچکی داده ها و فیچر ها و نویز داده ها دارد . و با توجه به این موارد میتوان مرتبه را یافت البته چون رفتار بایاس همیشه قابل مشاهده نیست و بعضی وقت به شکل های عجیبی عمل میکند ! پس همیشه نمیتوان مرتبه آنرا یافت.

سوالات امتیازی :

بخش 1)

در مورد قسمت اول در اوایل گزارش صحبت شده است.

بخش 2)

مدل‌ها معمولاً با استفاده از روش‌های نمونه‌گیری مجدد مانند اعتبارسنجی متقاطع k-fold که از آن میانگین نمرات مهارت محاسبه و به طور مستقیم مقایسه می‌شوند، ارزیابی می‌شوند.ولی ممکن است این روش گمراه کننده باشد چون نمیتوان فهمید تفاوت در mean skill scores واقعی است یا نتیجه محاسبات اشتباه آماری است.
Statistical significance tests برای این طراحی شده تا ما به این مشکل برنخوریم.آنها برای این طراحی شده اند که برای تعیین کمیت احتمال مشاهده نمونه‌های نمرات مهارت با این فرض طراحی شده‌اند که از توزیع یکسان گرفته شده‌اند. اگر این فرض رد شود، نشان می دهد که تفاوت در نمرات مهارت از نظر آماری معنی دار است.

ما مدلی را انتخاب میکنیم که بهترین دقت و کمترین خطا را داشته باشد. چالش انتخاب مدل با بهترین مهارت این است که چقدر می توانید به مهارت تخمینی هر مدل اعتماد کرد.برای همین از آزمون های فرض استفاده میکنیم.که میتوان گفت یک نمونه برای Statistical significance tests است . (بحث Statistical significance tests پیچیده تر از توضیحات بالا است ولی به همین بسنده میکنیم)

بخش 3)

mcc یک ابزار آماری است که برای ارزیابی مدل استفاده می شود. وظیفه آن اندازه گیری یا اندازه گیری تفاوت بین مقادیر پیش بینی شده و مقادیر واقعی است و معادل chi-square statistics برای contingency 2*2 table است.

Chi-square در واقع یک آزمون آماری است که برای بررسی تفاوت‌های بین متغیرهای categorical از یک نمونه تصادفی به منظور بررسی برازش مناسب بین نتایج مورد انتظار و مشاهده‌شده استفاده می‌شود.

MCC بهترین معیار طبقه بندی برای خلاصه کردن confusion matrix است .

confusion matrix حاوی 4 موجودیت True positives، True negatives، False positives ، False negatives است.که طبق فرمول زیر محاسبه میشوند.

$$\text{true positive rate (TPR)} = \frac{TP}{TP + FN}$$

(worst value =0; best value =1)

$$\text{true negative rate (TNR)} = \frac{TN}{TN + FP}$$

(worst value =0; best value =1)

$$\text{positive predictive value (PPV)} = \frac{TP}{TP + FP}$$

(worst value =0; best value =1)

$$\text{negative predictive value (NPV)} = \frac{TN}{TN + FN}$$

(worst value =0; best value =1)

اگر این پیش‌بینی نرخ‌های خوبی را برای هر چهار مورد از این نهادها به دست آورد، گفته می‌شود که معیار قابل اعتمادی است که نمرات بالایی ایجاد می‌کند.

(MCC)، در واقع، همبستگی کلاس های واقعی c را با برچسب های پیش بینی شده l اندازه گیری می کند:

$$\text{MCC} = \frac{Cov(c,l)}{\sigma_c \cdot \sigma_l} = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}}$$

نرخ MCC بین -1 و +1 متغیر است. همچنین در فرمول Cov(c,l) کوواریانس کلاس های واقعی c و برچسب های پیش بینی شده l است در حالی که σc و σl به ترتیب انحرافات استاندارد هستند.

+1 بهترین توافق بین مقادیر پیش بینی شده و واقعی است.

0 توافقی نیست به این معنی که پیش بینی با توجه به واقعیات تصادفی است

قسمت دوم سوالات :

در بخش اول به پاکسازی داده ها و مهندسی ویژگی و ... میپردازیم که مانند تمرین سری اول انجام شده و توضیحات آن درگزارش تمرین سری اول است ، در بخش بعدی ؛ الگوریتم linear regression را با خطای MSE خودمان پیاده سازی میکنیم .

دو تابع CostFunction و GradientDesent پیاده سازی linear regression است که R^2 score ما برابر با 0.47018 میشود.(توجه شود که فقط از 3 فیجر serviceCharge ، heatingType ، telekomUploadSpeed برای train استفاده شده)

سپس با همان فیچر های بالا و از طریق پکیج های موجود در sklearn استفاده میکنیم که در این حالت score ما 0.4790 میشود که به جوابی که در پیاده سازی خودمان انجام داده ایم نزدیک است.(توجه شود که در پیاده سازی خودم تعداد دوره ها 15000 و learning rate برابر با 0.002 است که البته میتوانیم آنها را تغییر دهیم ، همچنین برای درک بهتر ، خوب است که کد مطالعه شود.)

حالا سراغ رگرسیون ridge و lasso میرویم و برای اینکه مقایسه بهتری با نتایج مرحله قبل داشته باشیم فیچر ها را همان 3 فیچر مراحل قبل در نظر میگیریم.

در رگرسیون ridge ، train score برابر 0.4777 و R^2 score برابر 0.4790 شده است.

قسمت پیاده سازی مدل رگرسیون با absolute error در انتهای قسمت کد مربوط به این مسیله قرار دارد که پیاده سازی آن با تابع های GradientDescentAbsolute و CostFunctionAbsolute است که R2 score نهایی آن برابر 0.4626 است.(توضیحات مربوط به آن در قسمت کد قابل مشاهده است.)