باسمه تعالی

دانشگاه صنعتی شریف

دانشکده مهندسی برق

درس یادگیری ماشین

گزارش تمرین شماره چهار

علی محرابیان96102331

استاد: دکتر صالح کلیبر

زمستان 1398

در تمرین این سری از کتابخانه scikit برای پیاده سازی الگوریتم های مدنظر استفاده کردیم.

از dataset که در تمرین سری قبل پردازش کردیم،استفاده می کنیم.

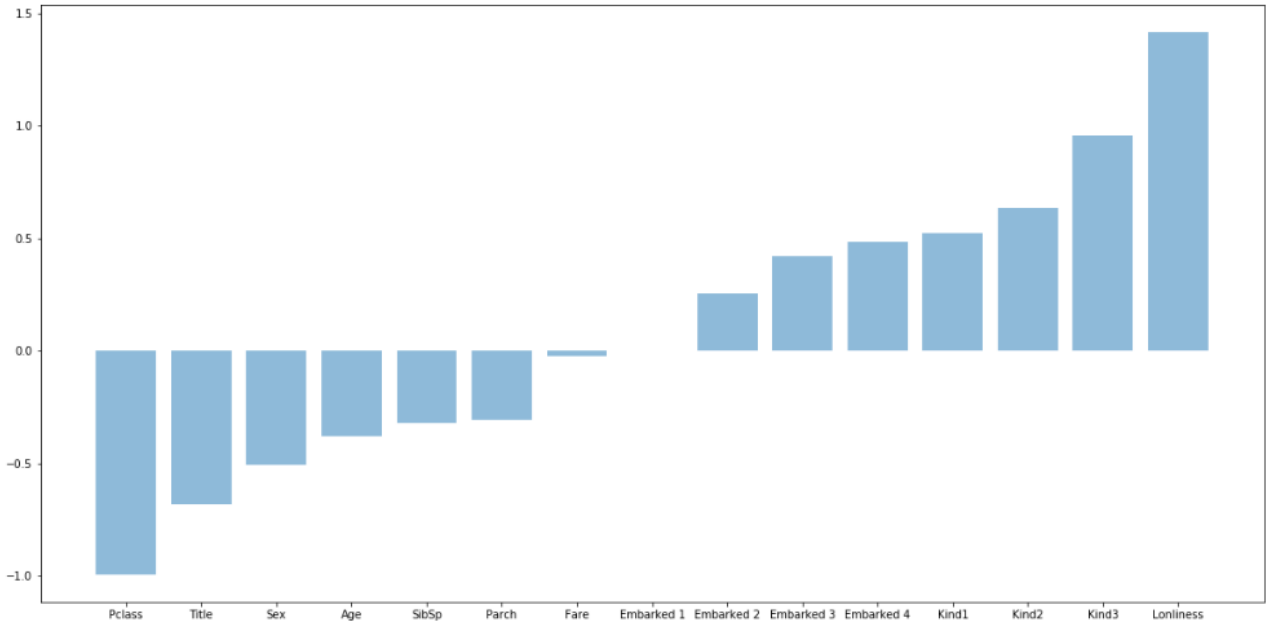
	Survived	Pclass	Title	Sex	Age	SibSp	Parch	Fare	Embarked 1	Embarked 2	Embarked 3	Embarked 4	Kind1	Kind2	Kind3	Lonliness	Rel
0	0	3	1.0	1.0	22.0	1	0	7.2500	1.0	0.0	0.0	0.0	0.0	1.0	0.0	1.0	1
1	1	1	2.0	0.0	38.0	1	0	71.2833	0.0	1.0	0.0	0.0	0.0	0.0	1.0	1.0	1
2	1	3	2.0	0.0	26.0	0	0	7.9250	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0
3	1	1	2.0	0.0	35.0	1	0	53.1000	1.0	0.0	0.0	0.0	0.0	0.0	1.0	1.0	1
4	0	3	1.0	1.0	35.0	0	0	8.0500	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	3.0	1.0	27.0	0	0	13.0000	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0
887	1	1	2.0	0.0	19.0	0	0	30.0000	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0
888	0	3	2.0	0.0	23.0	1	2	23.4500	1.0	0.0	0.0	1.0	0.0	0.0	1.0	1.0	3
889	1	1	1.0	1.0	26.0	0	0	30.0000	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0
890	0	3	1.0	1.0	32.0	0	0	7.7500	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0

891 rows × 17 columns

در تمامی قسمت ها جز قسمت امتیازی،dataset را به نسبت 80 به 20 split کردیم.

Logistic Regression

ضرایب به دست آمده به صورت زیر است.



همان طور که می توان دید،افرادی که همراه با خانواده بوده و تنها نبودند،شانس بیشتری برای زنده ماندن داشتند.در مورد بعدی می توان kind 3 را دید که مربوط به زنان می باشد که شانس بیشتری برای زنده ماندن داشتند.این مورد با بررسی هایی که در تمرین قبل کردیم هم مطابقت دارد.از سویی دیگر،با مشاهده Pclass می توان دید افرادی که در کابین های سطح پایین تری بودند،شانس کمتری برای زنده ماندن داشتند.با دیدن ویژگی های sex و title،می توان دید که بیشتر مردان کشته شدند چون بیشتر افراد کشتی از مردان بودند.

Score و confusion matrix به صورت زیر است.

```
Train score is :0.824438202247191
Test score is :0.8659217877094972
The Confusion Matrix is :[[109  10]
 [ 14  46]]
```

KNN

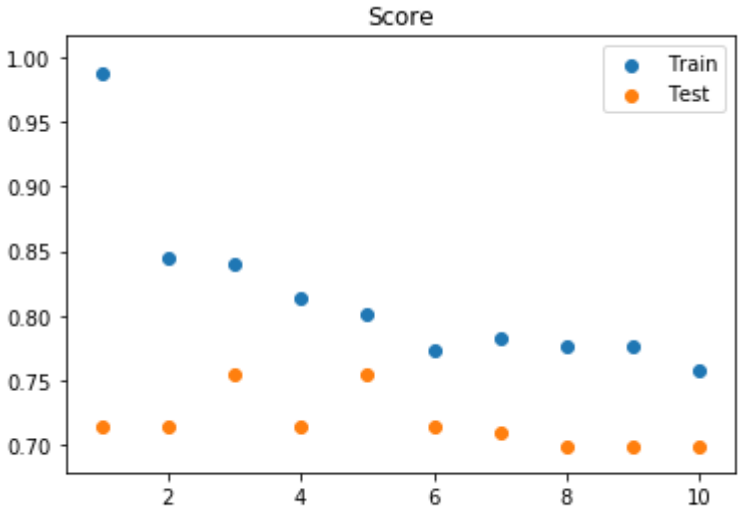
Score برای نرم های مختلف به صورت زیر است.

```
Test score for l1_norm is :0.770949720670391
Train score for l1_norm is :0.8651685393258427

Test score for l2_norm is :0.7262569832402235
Train score for l2_norm is :0.851123595505618

Test score for l3_norm is :0.7318435754189944
Train score for l3_norm is :0.8441011235955056
```

حال برای k های مختلف،score ها را به دست می آوریم.



مشاهده می شود که برای k=3،داده تست بیشترین score را دارد.

```
Max_score is for k = 3
```

ماتریس confusion به صورت زیر است.

```
The Confusion Matrix is :[[91 20]
 [24 44]]
```

SVM

در ابتدا مشاهده می شود که با ضریب regularization برابر با 1،کرنل های poly و rbf،خروجی مطلوبی ندارند.بنابراین ضریب c=10 را برای آنها در نظر می گیریم.

```
Test score for linear kernel is =0.8435754189944135
Train score for linear kernel is =0.8103932584269663
The F1-Score for linear kernel is =0.7971014492753623
```

```
Test score for poly kernel is =0.776536312849162
Train score for poly kernel is =0.9438202247191011
The F1-Score for poly kernel is =0.696969696969697
```

```
Test score for rbf kernel is =0.7262569832402235
Train score for rbf kernel is =0.9396067415730337
The F1-Score for rbf kernel is =0.6754966887417219
```

به طور کلی F1_score،پراکندگی الگوریتم اعمال شده بر روی داده ها را نشان می دهد.

فرض کنیم که 100 تراکنش داریم که 97 آن ها درست بوده و 3 تراکنش جعلی می باشند.می خواهیم مدل ما تراکنش های جعلی را پیدا کند.ماتریس confusion به صورت زیر است.

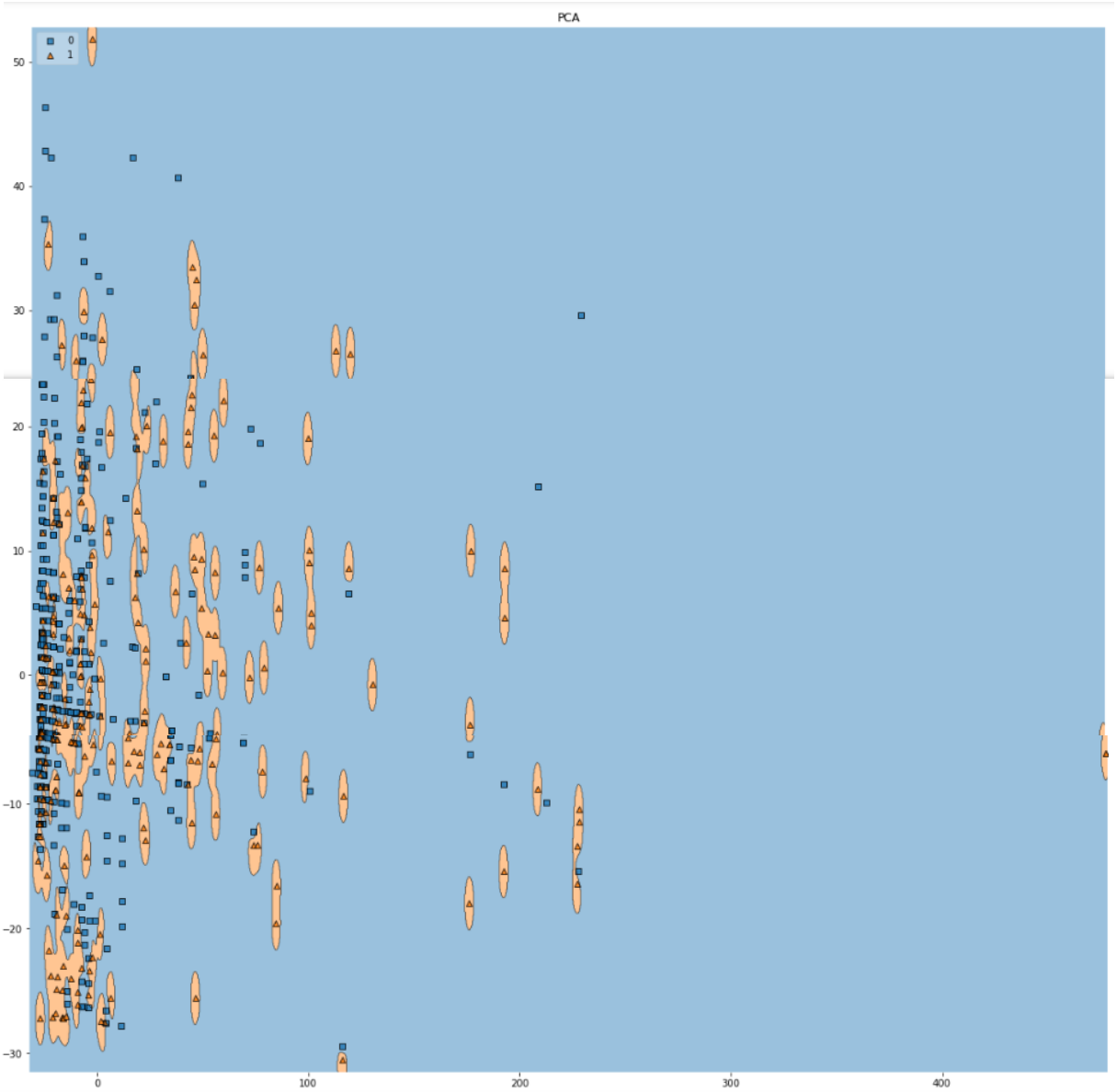
$$\begin{pmatrix} 3 & 97 \\ 0 & 0 \end{pmatrix}$$

پس در نتیجه ما به میانگین 51 درصد می رسیم که مطلوب نیست.بنابراین از میانگین هارمونیک($\frac{2xy}{x+y}$) استفاده می کنیم که به مقدار کوچکتر نزدیک تر است.

می توان پس از تلاش های مداوم دید که کرنل rbf،خروجی مطلوبی را در زمان کمتری می دهد.به کمک PCA، بعد dataset را به 2 کاهش می دهیم.

```
Test score with PCA and rbf kernel is =0.553072625698324
Train score with PCA and rbf kernel is =0.9367977528089888
The F1-Score with PCA and rbf kernel is =0.3548387096774193
```

برای داده های تمرین ،مرزهای تصمیم گیری به صورت زیر است.



همان طور که مشاهده می شود،شاهد overfitting هستیم.

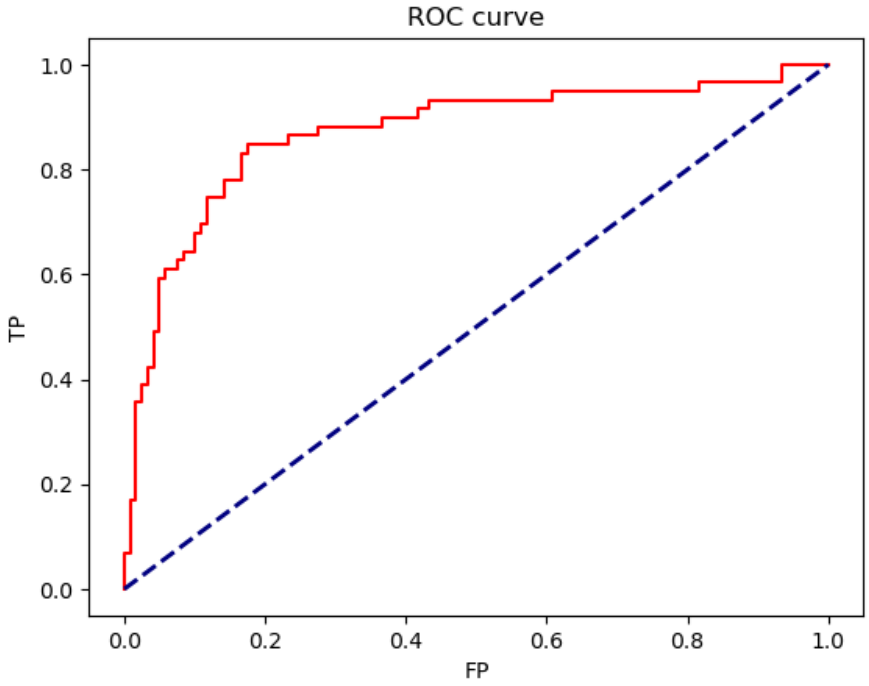
Naïve Bayes

پس از طراحی،خروجی های خواسته شده به صورت زیر هستند.

```
Test score is =0.8268156424581006
Train score is =0.7935393258426966
The AUC is =0.805166147455867
```

به طورکلی ضریب smoothing،مقداری از بزرگترین واریانس تمام dataset ما می باشد که توازن بیشتر،به واریانس feature ها اضافه می شود.

AUC،سطح زیر منحنی ROC بوده که به صورت زیر می باشد.



ROC،تعداد داده های مثبتی که درست پیش بینی شده اند را به ازای تعداد داده های مثبتی که غلط پیش بینی شده اند،برای treshhold های متفاوت رسم می کند.

در accuracy،تعداد داده هایی که درست پیش بینی شده اند به کل داده ها حساب می شود.می توان حدس زد زمانی که تعداد داده های نادرست زیادی داشته باشیم،این مور معیار خوبی برای ما نبوده و می توان ار AUC استفاده کرد.

Random Forest

با استفاده از 5 fold،پارامتر های عمق درخت و تعداد درخت های جنگل را بهینه می کنیم.می توان دید که در میان تمام الگوریتم ها،این الگوریتم خروجی های مطلوب تری را به ما می دهد.بنابراین در قسمت امتیازی از همین الگوریتم استفاده می کنیم.

```
Optimal number of trees is = 31
Optimal number for depth is = 10
Test score is =0.8603351955307262
Train score is =0.9367977528089888
The F1-Score is =0.8031496062992127
```

Neural Net

با استفاده از 3 fold مقادیر بهینه را برای پارامتر های alpha و learning_rate در دو حالت به دست می آوریم. در حالتی دارای 20 و 10 نورون در لایه ها هستیم،خروجی ها به صورت زیر هستند.

```
Optimal number for learning_rate is = 0.01
Optimal number for alpha is = 0.1
Test score is =0.7653631284916201
Train score is =0.8061797752808989
The Confusion Matrix is :[[82 23]
 [19 55]]
```

برای حالتی که دارای 50 و 100 نورون هستیم،خروجی به صورت زیر است.

```
Optimal number for learning_rate is = 0.001
Optimal number for alpha is = 1.0
Test score is =0.7932960893854749
Train score is =0.8286516853932584
The Confusion Matrix is :[[98  7]
 [30 44]]
```

مشاهده می شود که در حالت دوم،خروجی ها مطلوب تر هستند.به طور مثال TP در حالت دوم بیشتر و FP کمتر است که این ها مطلوب ما هستند.score برای تست نیز در حالت دوم بیشتر است.

Bonus

در این قسمت بعضی از feature های ناکارآمد را حذف کرده و از dataset زیر استفاده می کنیم.

	Survived	Pclass	Title	Sex	Age	Kind1	Kind2	Kind3	Lonliness	Rel
0	0	3	1.0	1.0	22.0	0.0	1.0	0.0	1.0	1
1	1	1	2.0	0.0	38.0	0.0	0.0	1.0	1.0	1
2	1	3	2.0	0.0	26.0	0.0	0.0	1.0	0.0	0
3	1	1	2.0	0.0	35.0	0.0	0.0	1.0	1.0	1
4	0	3	1.0	1.0	35.0	0.0	1.0	0.0	0.0	0
...	...	...	...	...	...	...	...	...	...	...
886	0	2	3.0	1.0	27.0	0.0	1.0	0.0	0.0	0
887	1	1	2.0	0.0	19.0	0.0	0.0	1.0	0.0	0
888	0	3	2.0	0.0	23.0	0.0	0.0	1.0	1.0	3
889	1	1	1.0	1.0	26.0	0.0	1.0	0.0	0.0	0
890	0	3	1.0	1.0	32.0	0.0	1.0	0.0	0.0	0

891 rows × 10 columns

با استفاده از 5 fold،پارامتر های عمق درخت و تعداد درخت ها را بهینه می کنیم.در حالت اول،داده ها را به نسبت 80 به 20 split می کنیم.خروجی ها به صورت زیر هستند.

```
Test score is =0.9106145251396648
Train score is =0.797752808988764
The F1-Score is =0.8688524590163934
The Confusion Matrix is :[[110   9]
 [  7  53]]
The Accuracy is =0.9106145251396648
The Log Loss is =3.0873051323767755
The MAE is =0.0893854748603352
```

مشاهده می شود که F1_score نسبت به حالتهای قبل بیشتر شده که نشان می دهد که مقادیر به هم نزدیک تر هستند و الگوریتم بهتر عملکرده است.با بررسی ماتریس confusion،می بینیم که TP ما افزایش چشمگیری داشته است که نشان از عملکرد بهتر الگوریتم ما دارد.

حال به همان روش قبلی،پارامترهای بهینه را پیدا کرده ولی این با داده هارا با نسبت 90 به 10 split می کنیم.
خروجی ها به صورت زیر هستند.

```
Test score is =0.9333333333333333
Train score is =0.8114856429463171
The F1-Score is =0.9032258064516129
The Confusion Matrix is :[[56  2]
 [ 4 28]]
The Accuracy is =0.9333333333333333
The Log Loss is =2.3026028618258283
The MAE is =0.06666666666666667
```

در این حالت شاهد دقت 93.3 بر روی داده های تست هستیم.در این حالت F1_score به عدد 90 رسیده که بسیار مطلوب است.