بسم الله الرحمن الرحیم

گزارش کار پروژه­ی نظریه­ی یادگیری ماشین

گردآورندگان:

فاطمه اکبری، سنا آریا، محمد جواد عباسپور، سمیرا مرادزاده، احمد یزدانی

استاد راهنما:

آقای دکتر حسین حاجی ابوالحسن

نیم سال اول 1400-1401

1. **مجموعه داده­ی BBBP**

* **توصیف مجموعه داده**

این مجموعه دارای 2039 سطر از داده است. داده ها از جنس طبقه بندی بوده و دارای 200 ویژگی هستند.

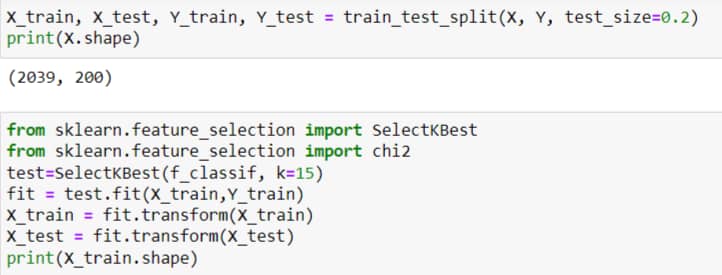
* **فراخوانی داده ها و خواندن مقادیر داده ها**

با دستور read\_csv فایل های حاوی داده ها را میخوانیم و با دستور

**next(header)،** سر تیتر ها را مشخص میکنیم.

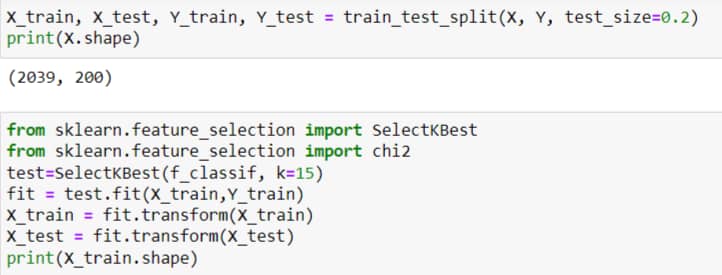
* **تقسیم بندی داده ها**

مجموعه ی داده هارا با استفاده از دستور train\_test\_split به دو قسمت آموزشی و آزمایشی با مقیاس 0.2 تقسیم میکنیم.



* **انتخاب بهترین ویژگی ها**

**با استفاده از دستورselectbest تعداد 15 ویژگی برتر را انتخاب میکند.**



* **بررسی الگوریتم های مختلف بر روی داده**
* **DecisionTree**

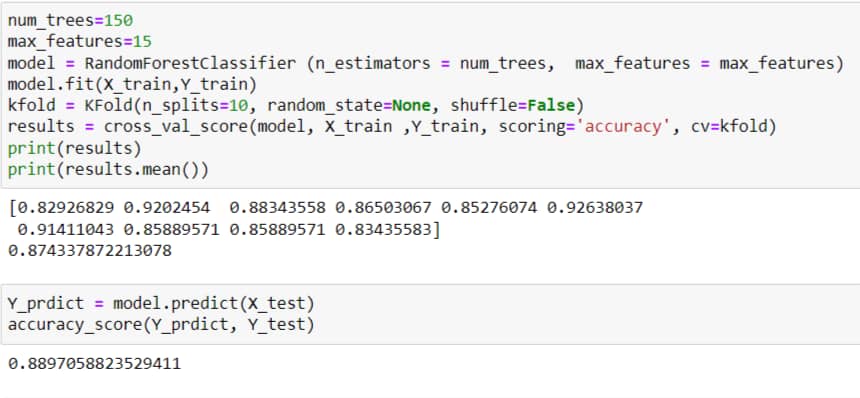
1. یک آبجکت DecisionTreeClassifier، با هایپر پارامتر 10، که نشان دهنده ی عمق درخت است تعریف میکنیم.
2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



دقت این مدل برابر با 0.8553 است.

* **RandomForest**

1. به آبجکت RandomForestClassifier، هایپرپارامتر های 150، که برای تعداد تخمین گر ها، و هایپرپارامتر 15، که بیشترین عمق درخت است را نسبت میدهیم.
2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



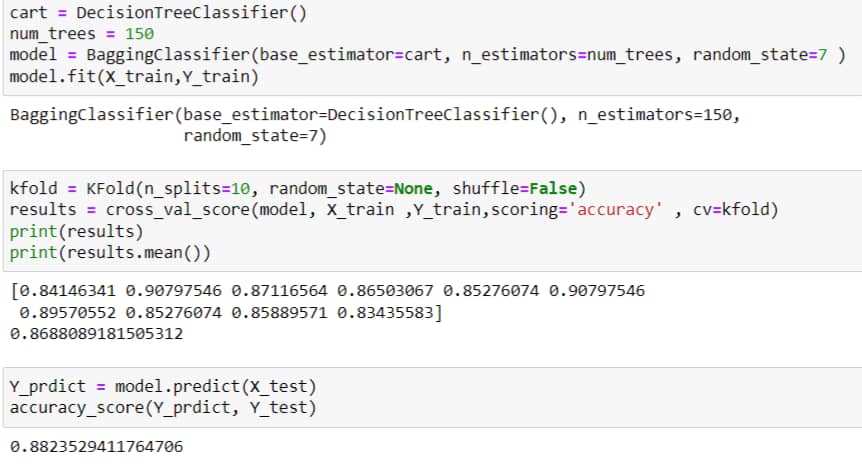
دقت این مدل برابر است با 0.8897 است.

* Bagging

1- به آبجکت BaggingClassifier، هایپر پارامتر های DecisionTreeClassifier() که نشان دهنده تخمین گر پایه و عدد 150 که نشان دهنده ی تعداد تخمین گر ها است را نسبت میدهیم.

2- داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم

3- دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



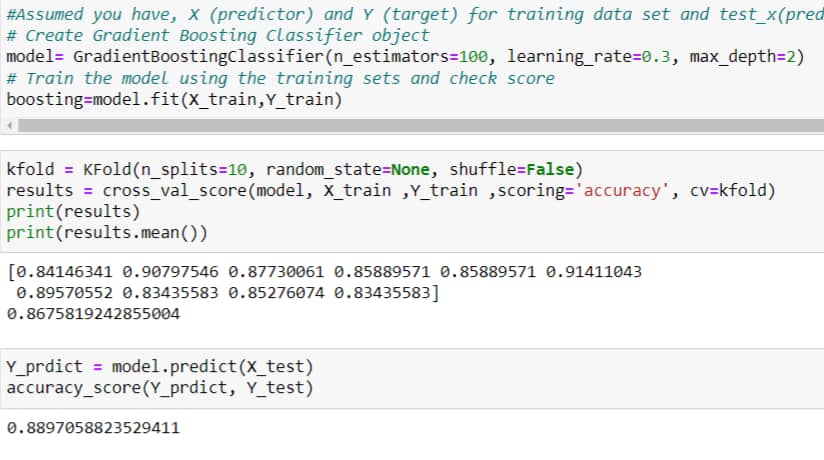
دقت این مدل برابر است با 0.8823 است.

* Boosting

1- به آبجکت GradientBoostingClassifier، هایپر پارامتر های 100 که نشان دهنده تعداد تخمین گر ها ، عدد 0.3 که نشان دهنده ی وزن هر تخمین گر و عدد 2 که نشان دهنده ی بیشینه عمق درخت است را نسبت میدهیم.

2- داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم

3- دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



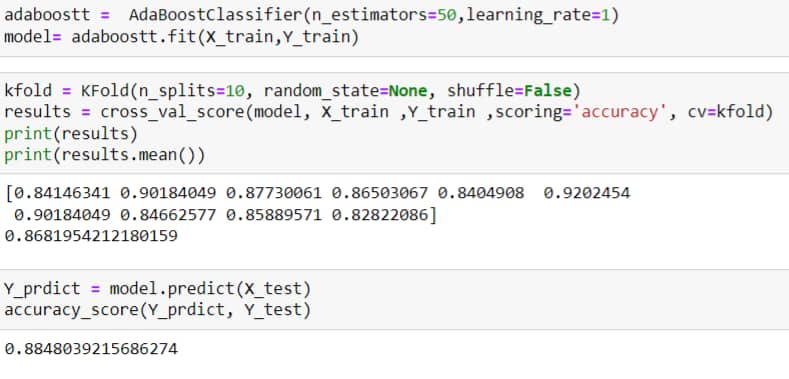
دقت این مدل برابر است با 0.8897 است.

* AdaBoost

1-به آبجکت AdaBoostClassifier، هایپر پارامتر های 50 که نشان دهنده تعداد تخمین گر ها و عدد 1 که نشان دهنده ی وزن هر تخمین گر است را نسبت میدهیم.

2- داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم

3- دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



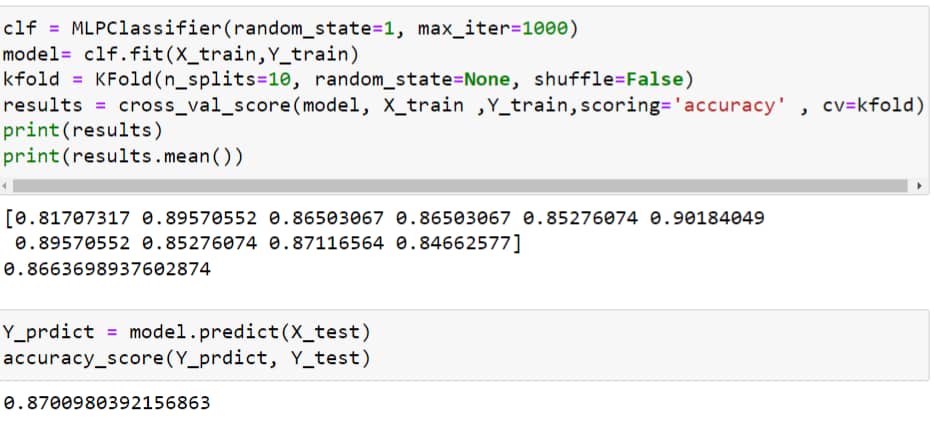
این مدل برابر است با 0.8848 است.

* MLP

1-به آبجکت MLPClassifier، هایپر پارامتر 1000 که نشان دهنده بیشینه تکرار مدل است را نسبت میدهیم.

2- داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم

3- دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



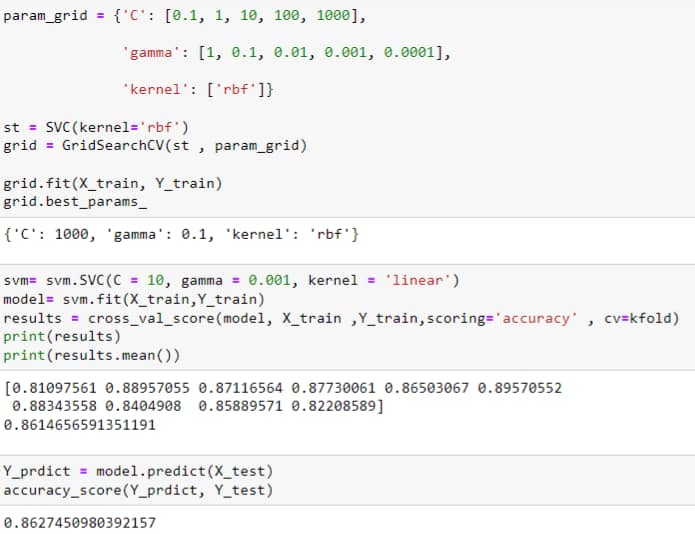
این مدل برابر است با 0.8700 است.

* SVM

1- با دستور gridsearchCV بهترین هایپرپارامتر ها را متناسب با مجموعه داده انتخاب کرده و سپس در مدل جایگذاری میکنیم.

2- داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.

3- دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



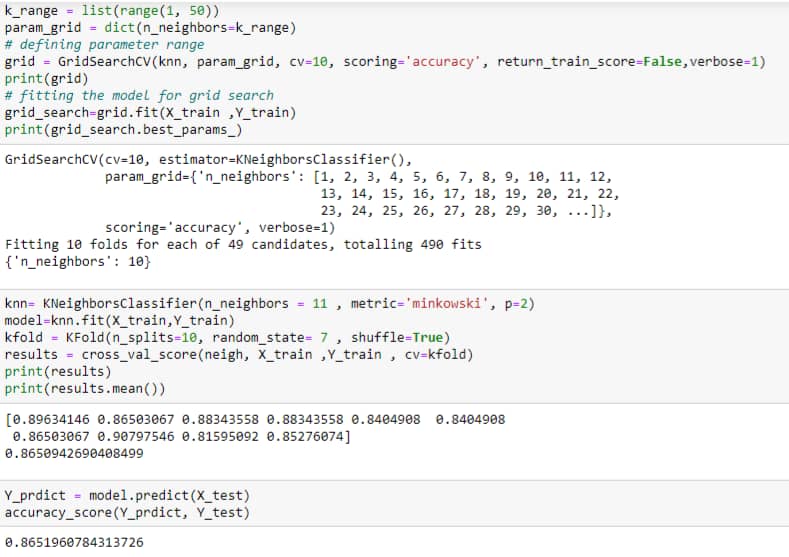
دقت این مدل برابر با 0.8627 است.

* KNN

1- آبجکت KNeighborsClassifier را تعریف کرده و با استفاده از دستور gridsearchCV بهترین k که برابر10 شد را انتخاب میکنیم.

2- داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.

3- دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



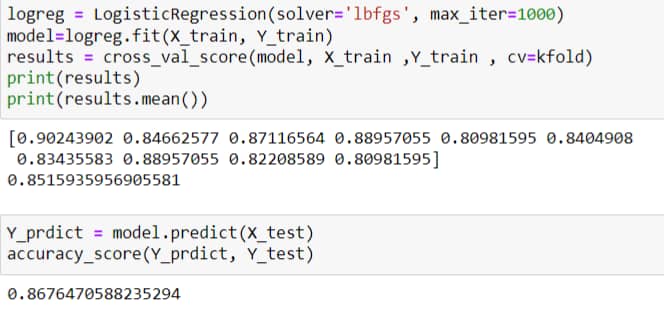
دقت این مدل 0.8651 است.

* LogisticRegression

1- آبجکت LogisticRegression را با هایپر پارامتر 1000 که نشان دهنده بیشینه تکرار مدل است تعریف میکنیم.

2- داده را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.

3- دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.



دقت این مدل برابر با 0.8676 است.

بهترین مدل bagging با 0.89 دقت است.

1. **مجموعه داده­ی QM7**

* **توصیف داده**

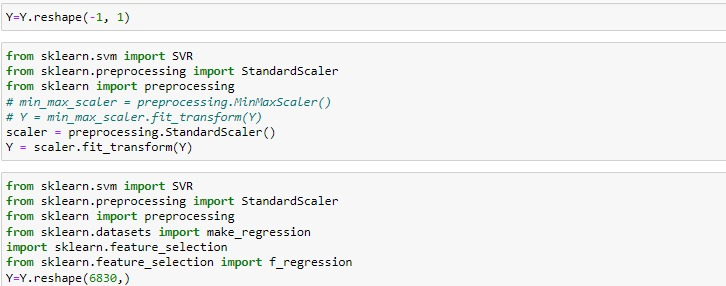
این مجموعه دارای 6830 سطر از داده است. داده ها از جنس رگرسیون بوده و دارای 200 ویژگی هستند.

* **فراخوانی داده ها و خواندن مقادیر داده ها**

ابتدا فایل داده را با دستور read\_csv میخوانیم و مقادیر آن را به متغییر X و Y نسبت می دهیم.

* **مقیاس بندی داده ها**

حال Y هارا با مقیاس استاندار، مقیاس بندی میکنیم.



* **تقسیم بندی داده ها**

مجموعه ی داده هارا با استفاده از دستور train\_test\_split به دو قسمت آموزشی و آزمایشی با مقیاس 0.2 تقسیم میکنیم.

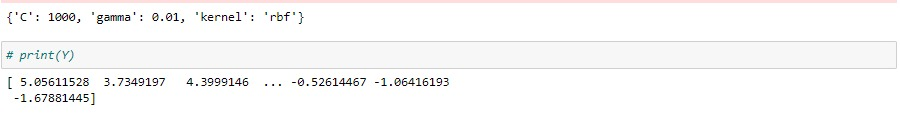
* **کاهش بعد**

به دلیل اینکه ویژگی ها به یکدیگر وابستگی زیادی دارند، انتخاب کمتر از 100 ویژگی باعث کم شدن دقت مدل میشود. بنابراین با دستور select\_best ، تعداد ویژگی ها را از 200 به 100 کاهش دادیم.

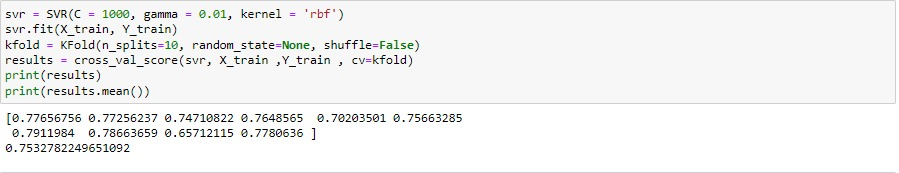
* **بررسی الگوریتم های مختلف بر روی داده**
* **SVM**

1. با دستور gridsearchCV بهترین هایپرپارامتر ها را متناسب با مجموعه داده انتخاب کرده و سپس در مدل جایگذاری میکنیم.





2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.



3. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

دقت این مدل برابر با 0.703351 است.

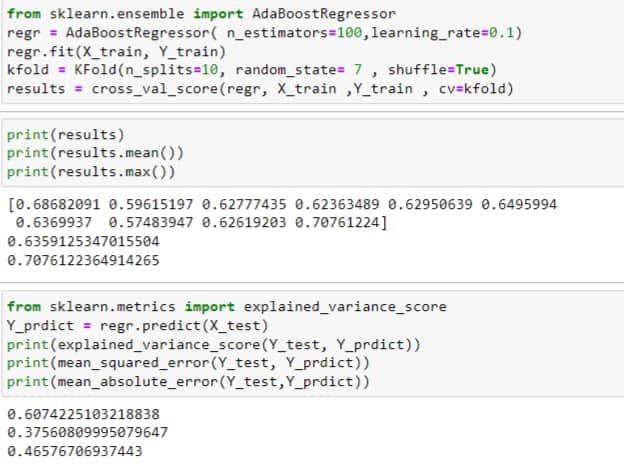


* **AdaBoost**

1. به آبجکت AdaBoostRegressor، هایپرپارامتر های 100، که تعداد تخمین گر، و 0.1، که وزن هرکدام از تخمین گر هاست را نسبت میدهیم (مقدار وزن کمتر باشد دقت ما بیشتر میشود).
2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.

3. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

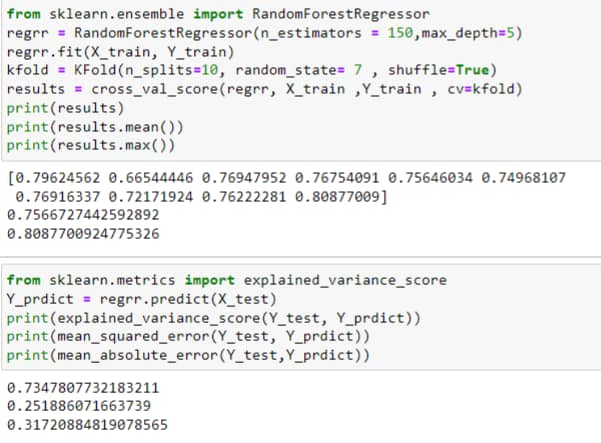
دقت این مدل برابر با 0.6074 است.



* **RandomForest**

1. به آبجکت RandomForestRegressor، هایپر پارامترهای 150، که تعداد تخمین گر، و 5، که بیشترین عمق درخت را نشان میدهد، نسبت می دهیم.( که هرچه تعداد درخت ها و عمقشان بیشتر باشد دقت ما بیشتر است)
2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

دقت این مدل برابر با 0.7347 است.



* **Bagging**

1. به آبجکت BaggingRegressor ، هایپر پارامتر 10، که تعداد تخمین گر است را نسبت می دهیم.
2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

دقت این مدل برابر با 0.7744 است.



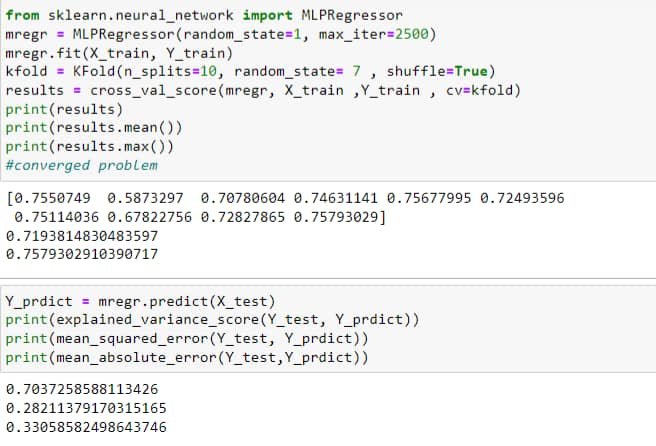
* **MLP**

1 .به آبجکت MLPRegressor، هایپرپارامتر 2500، که بیشترین دفعات تکرار برای همگرایی مدل شبکه های عصبی است را نسبت می دهیم.

2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.

3.دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

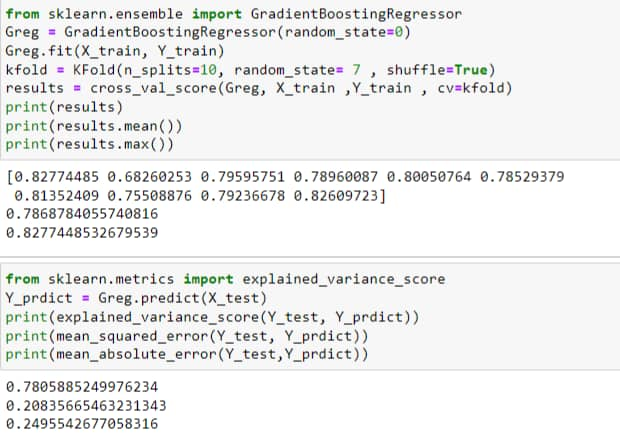
دقت این مدل برابر 0.7037 است.



* **Boosting**

1. آبجکت GradientBoostingRegressor را تعریف میکنیم که پیش فرض آن 100 تخمین گر است.
2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

دقت این مدل برابر با 0.7805 است.

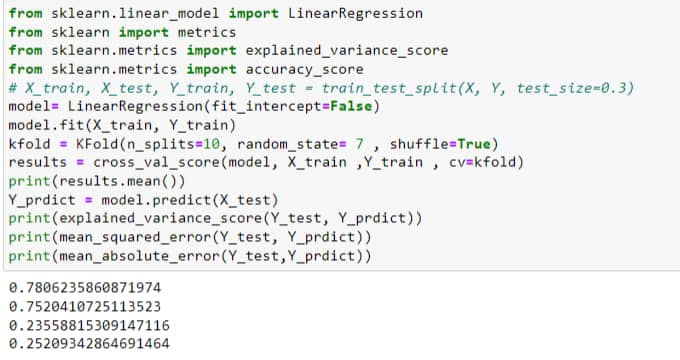


* **LinearRegression**

1.آبجکت LinearRegression را تعریف میکنیم و داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.

2. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

دقت این مدل برابر با 0.7520 است.



* **KNeighbors**

1. آبجکت KNeighborsRegressor را تعریف کرده و با استفاده از دستور gridsearchCV بهترین k را انتخاب میکنیم.
2. داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

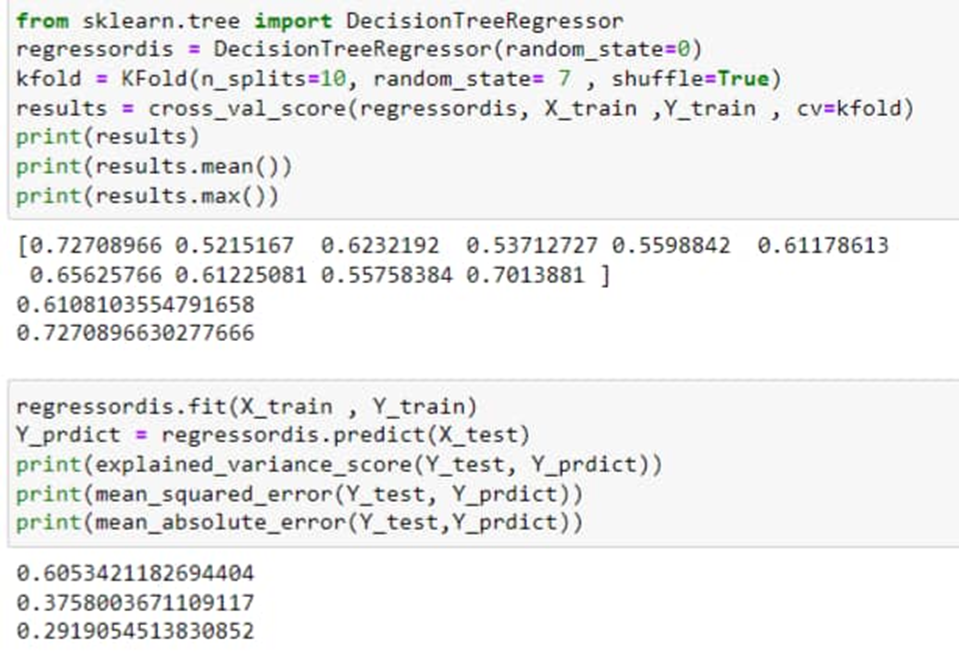
دقت این مدل 0.4358 است.



* **DecisionTree**

1. آبجکت DecisionTreeRegressor را تعریف کرده و داده ها را با دستور kfold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
2. دقت مدل را بر روی داده های آزمایشی می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

دقت این مدل برابر با 0.6053 است.



* **انتخاب بهترین مدل**

با توجه به بررسی های انجام شده با مدل های بالا به این نتیجه رسیدیم که بهترین دقت را مدل های boosting و bagging با دقت های 0.78 دارند.

1. **مجموعه داده­ی Tox21**

* **توصیف مجموعه داده**

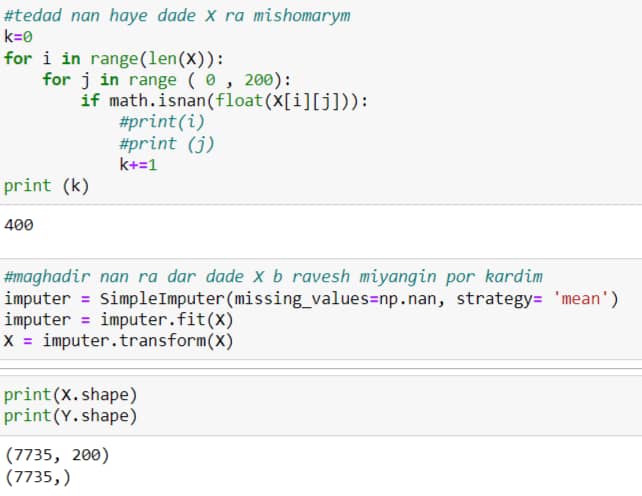
این مجموعه دارای 7832 سطر از داده است. داده ها از جنس طبقه بندی بوده و دارای 200 ویژگی هستند.

* **فراخوانی داده ها و خواندن مقادیر داده ها**

با دستور read\_csv فایل های حاوی داده ها را میخوانیم و با دستور

**next(header)،** سر تیتر ها را مشخص میکنیم.

* **حذف داده های گمشده ی لیبل ها و X های متناظر با آن**
* **جانهی مقادیر گمشده**



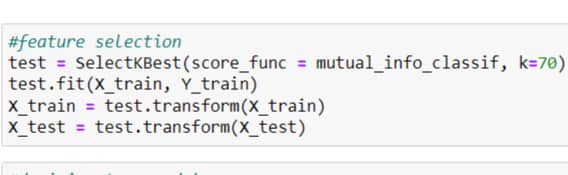
* **تقسیم بندی داده ها**

مجموعه ی داده هارا با استفاده از دستور train\_test\_split به دو قسمت آموزشی و آزمایشی با مقیاس 0.2 تقسیم میکنیم.



* **انتخاب بهترین ویژگی ها**

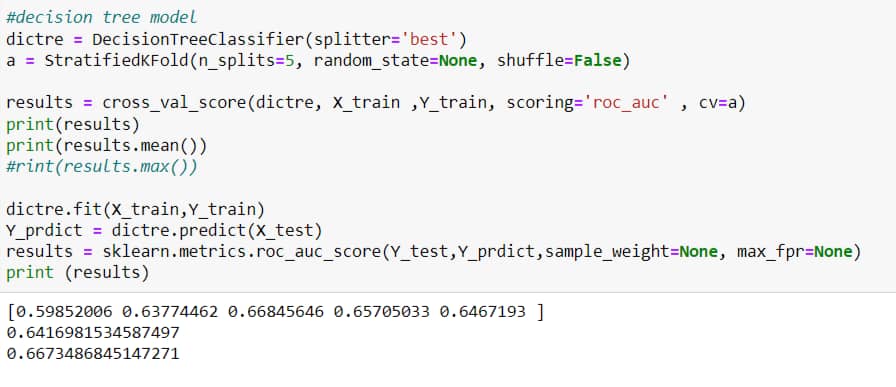
با استفاده از دستورselectbest تعداد 70 ویژگی برتر را انتخاب میکند.



* **بررسی الگوریتم های مختلف بر روی داده**
* **DecisionTree**

1. **ابتدا آبجکت DecisionTreeClassifier را فرا خوانی میکنیم.**
2. داده ها را با دستور StratifiedKFold به 5 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

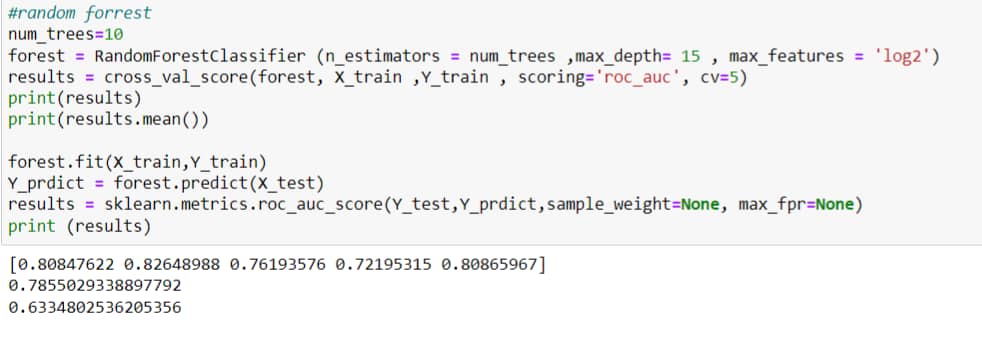
دقت برابر با 0.6673 است.

****

* **RandomForest**

1. ابتدا با دستور GridSearchCV بهترین هایپرپارامتر ها را انتخاب میکنیم.
2. آبجکت RandomForestClassifier را با هایپر پارامتر های پیشنهادی فرا خوانی میکنیم.
3. داده ها را با دستور cv به 5 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
4. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

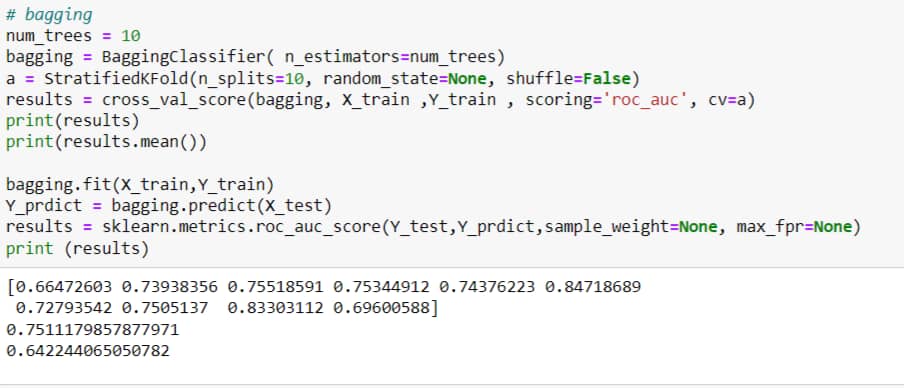
دقت برابر با 0.6334 است که با خطر بیش برازش رو به رو هستیم.



* **Bagging**

1. آبجکت BaggingClassifier را فراخوانی میکنیم.
2. داده ها را با دستور StratifiedKFold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

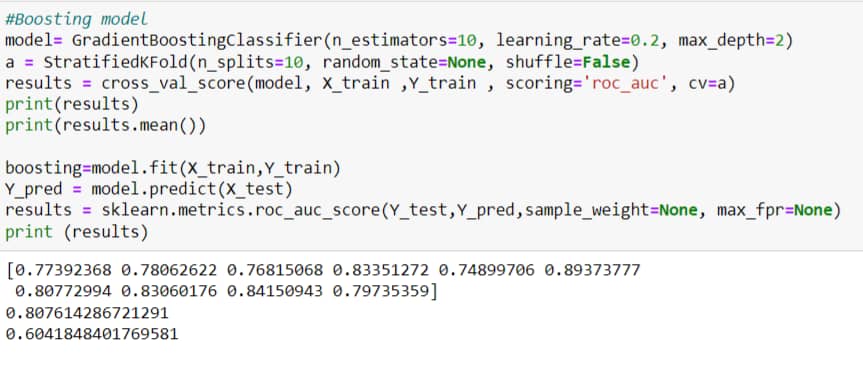
دقت این مدل برابر است با 0.64 که با خطر بیش برازش رو به رو هستیم.

****

* **Boosting**

1. آبجکت GradientBoostingClassifier را فراخوانی میکنیم.
2. داده ها را با دستور StratifiedKFold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

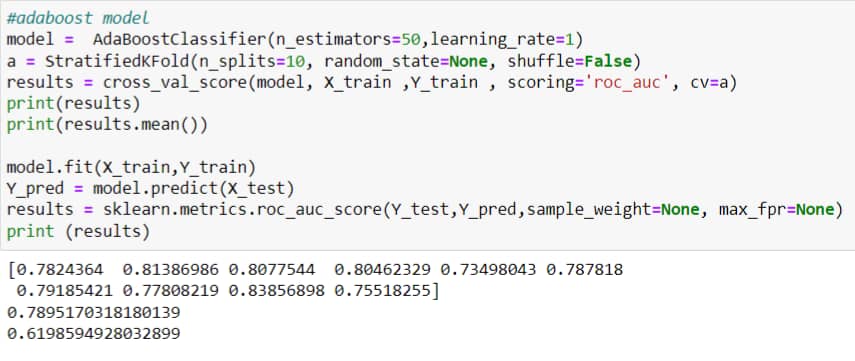
دقت برابر با 0.6041 است که با خطر بیش برازش رو به رو هستیم.

****

* **AdaBoost**

1. آبجکت AdaBoostClassifier را فراخوانی میکنیم.
2. داده ها را با دستور StratifiedKFold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

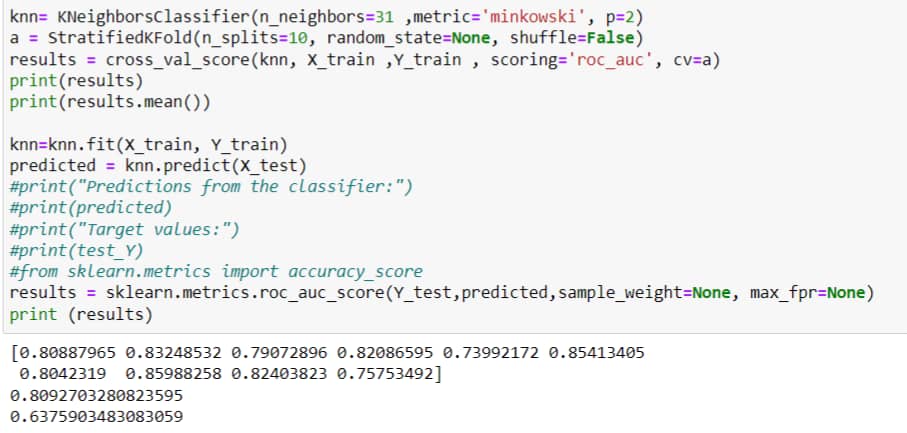
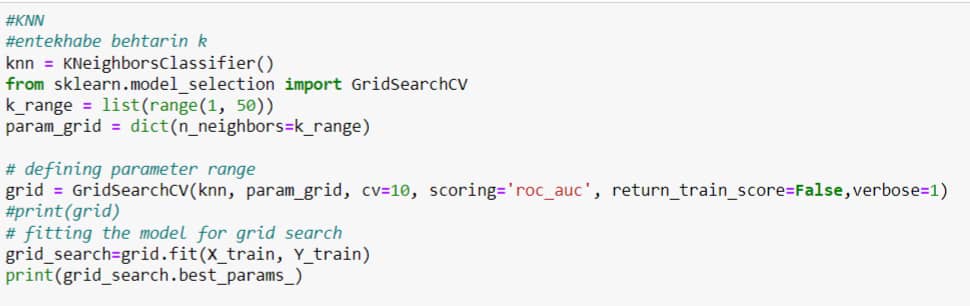
دقت این مدل برابر 0.61 است که با خطر بیش برازش رو به رو هستیم.

****

* **KNeighbors**

1. ابتدا با دستور GridSearchCV بهترین هایپر پارامتر ها را برای مدل KNeighborsClassifier انتخاب میکنیم.
2. داده ها را با دستور StratifiedKFold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

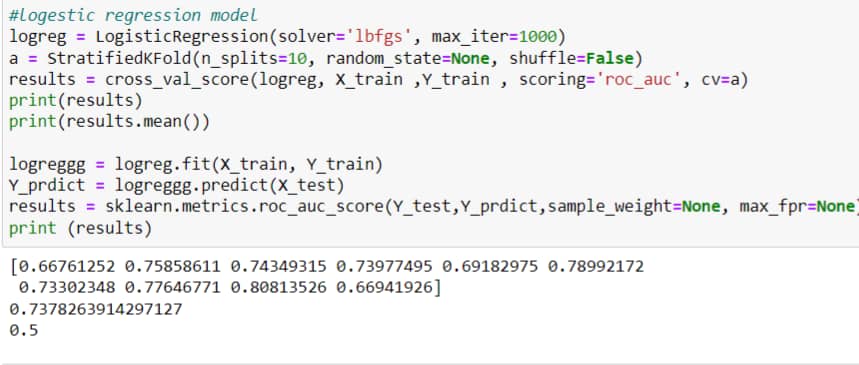
دقت این مدل برابر با 0.63 است که با خطر بیش برازش رو به رو هستیم.

****

* **LogisticRegression**

1. آبجکت LogisticRegression را فراخوانی میکنیم.
2. داده ها را با دستور StratifiedKFold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

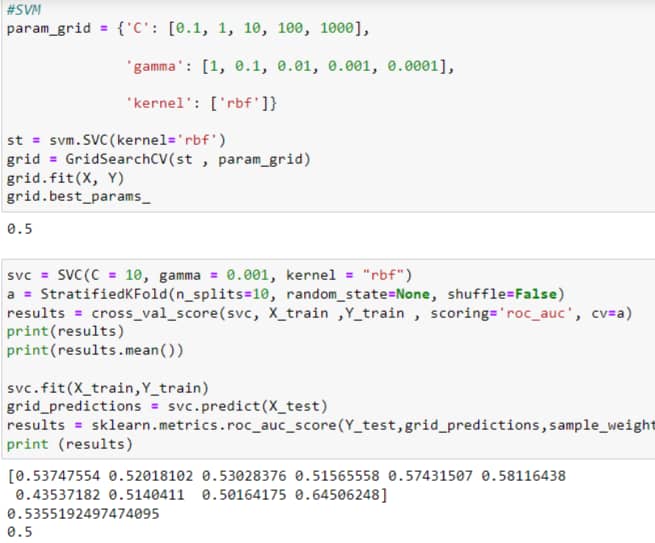
دقت این مدل برابر با 0.5 است که با خطر بیش برازش رو به رو هستیم.

****

* **SVM**

1. ابتدا با دستور GridSearchCV بهترین هایپر پارامتر ها را برای مدل SVC انتخاب میکنیم.
2. داده ها را با دستور StratifiedKFold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

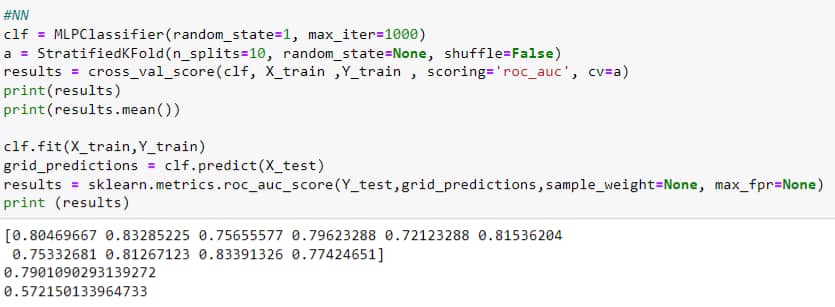
دقت این مدل برابر با 0.5 است.

****

* **MLP**

1. آبجکت MLPClassifier را فراخوانی میکنیم.
2. داده ها را با دستور StratifiedKFold به 10 قسمت تقسیم کرده و با استفاده از cross\_val\_score، مدل ها را ارزش گذاری میکنیم.
3. دقت مدل را بر روی داده های آزمایشی با متریک roc\_auc می­سنجیم و اگر دقت بدست آمده در بازه ی دقت مدل های ارزش گذاری شده باشد، از خطر بیش برازش در امانیم.

دقت این مدل برابر با 0.5721 است و با خطای بیش برازش رو به رو هستیم.



در دو مدل svm و decisiontree بیش برازش نداشتیم که دقت آنها برابر با 0.5 و 0.667 است.

Random\_state

* برای ورودی spilit از این پارامتر استفاده مینماییم. اگر ورودی این پارامتر None باشد آنگاه با هر بار اجرا جواب های مختلفی میگریم ولی اگر ورودی آن عددی حسابی باشد با اجرا های مختلف یک خروجی ثابت میگیریم. اعداد رایج مورد استفاده درآن 0 تا 42 است.
* برای ورودی model از این پارامتر استفاده مینماییم. اگر مقدار آن را برای مدلی مانند DecisionTree ، None در نظر بگیریم با هر بار اجرا درختی متفاوت به مجموعه ی اموزشی ما نسبت میدهد، ولی اگر مقدار آن را عددی حسابی در نظر بگیریم همواره یک نوع درخت ثابت با توجه به عدد ورودی را به داده های آموزشی نسبت میدهد.