گزارش فاز دوم پروژه ی یادگیری آماری

استاد درس : دکتر هدی محمد زاده

دانشجو:

سعید منصورلکورج – 99102304

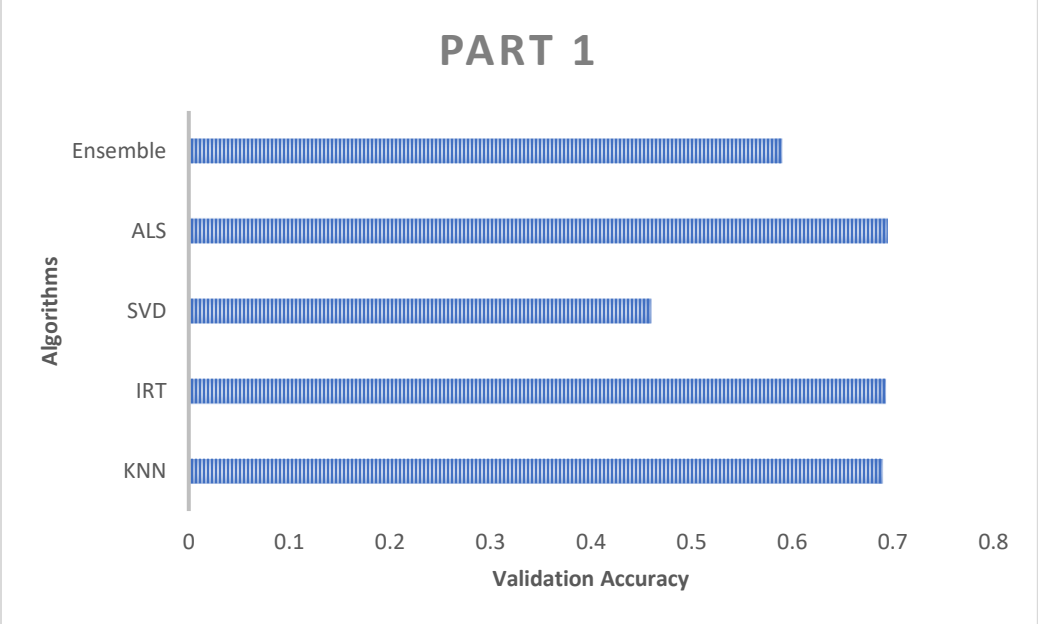
فاز دوم پروژه

مقدمه :

به صورت کلی اولین ایده ای که برای فاز دوم به ذهنم رسید افزایش ویژگی ها به داده ها بود که برای اینکار از فایل question_meta.csv استفاده کردم و با استفاده از تعریف توابع یک ستون به train و validation اضافه کردم که آرایه ای از subject ها در آنجا قرار دارد اما نکته ی مورد نظر این است که با وجود چنین آرایه ای امکان classification وجود ندارد پس در این بخش مانند تمرین 5 باید از encoder ها استفاده می کردیم که من از MultilabeBinarizer استفاده کردم که هر subject را به صورت 0 و یک در می آورد اما همین مورد نیز مشکلات مربوط به خودش را دارد یعنی مثلا اینقدر تعداد متغیر ها را زیاد می کند که احتمالا باعث overfitting زیادی می شود و این مورد مطلوب ما نمی باشد و در اینجا ایده ی کاهش متغیر ها به کمک PCA و دیگر روش ها متولد شد اما الگوریتم ما چه باشد؟

در ابتدا به دنبال بهبود روش Ensemble بودم که از base-learner های متفاوتی مانند knn و... استفاده کردم، مدل را از bagging به adaboost تبدیل کردم و... (بخشی از این کد ها در فایل های پایتون نیز موجود می باشد) اما در همه ی این موارد نتوانستم به دقتی بیشتر از 0.6 برسم در صورتی که در فاز اول حتی در بخش هایی مانند IRT و KNN حتی به دقت هایی حدود 0.7 هم رسیده بودم پس هنوز جایی ایراد داشتم.

در شکل زیر نمودار دقت ها را در فاز یک مشاهده می کنید:



ایده ای که به ذهنم رسید استفاده از data processing بود که به این نتیجه رسیدم که میتوانم دو متغیر خودم بر اساس داده ها به صورت مستقیم (یعنی به عنوان بایاس و نه به عنوان متغیر جدید در داده ها) اضافه کنم، که ایده ی این مورد را از بخش IRT در فاز یک گرفتم پس دو متغیر میزان هوش دانش آموزان را به داده ها اضافه کردم و نحوه ی اعمال هم به صورت ضریب در نتیجه ی نهایی بود اما مشکل بعدی این بود که در classification نتیجه ها پیش بینی شده به صورت 0 و 1 می باشد و اینطور این ضرایب هیچ تاثیری ندارند. برای رفع این مشکل باید به جای classification مستقیم از regression استفاده می کردم وسپس با اعمال threshold آنرا به classification تبدیل کرده؛ البته باید دقت کرد این

روش جدیدی نمی باشد زیرا در ANN نیز از این روش ها استفاده می کنیم که از نمونه های آن می توان به Perceptron اشاره کرد. من در اینجا از DecisionTreeRegressor استفاده کردم و سپس بعد از اعمال ضرایب به آن با threshold 0.5 آنرا classify کردم که دقت به 0.67 رسید، که با اینکه نسبت به بهترین algorithm های فاز یک دقت بهتری ندارد اما از 0.58 مربوط به Decision Tree فاز یک پیشرفت قابل توجهی کرده است.(در اینجا فقط هوش دانش آموزان را در نظر گرفتم)

در مرحله ی بعد همینکار را با question ها کردم که دقتی حدود 0.62 داد که نشان می دهد زیاد بایاس قابل اعتماد نمی باشد.

حالا همین کار را با KNieghborRegressor امتحان کردم که دقت از 0.55 به حدود 0.65 رسید (دقت شود روش استفاده شده از KNN در فاز یک کاملا متفاوت می بود اما در اینجا ما دقت اولیه ی حدود 0.55 داشتیم با استفاده از تابع KNieghborClassifire)

با تغییر threshold ها در سه مدل به نتیجه ی زیر رسیدم (تعیین مقدار هایپرپارامتر ها) :

Accuracy of tree 1 is: 0.6838837143663562

Accuracy of knn is : 0.6591871295512278

Accuracy of tree 2 is: 0.6773920406435224

حالا مانند روش Ensemble از نتایج این سه مورد رای گیری کرده:

Accuracy of tree 1 is: 0.6836014676827548

Accuracy of knn is : 0.659046006209427

Accuracy of tree 2 is: 0.6748518204911093

Accuracy of voted is : 0.6418289585097375

مشاهده می شود که دقت کمتر شد پس روش چندان خوبی نمی باشد، حالا threshold را پس از voting اعمال کرده شاید نتیجه ی بهتری بگیریم.

در این صورت هم دقت ما حدود 0.682 شد که باز هم کمی از tree1 کمتر می باشد اما میتوانیم آنرا بهترین نتیجه در نظر بگیریم.

مروری بر الگوریتم :

- در ابتدا subject ها هم به data اضافه کردم
- سپس PCA گرفته و تعداد ویژگی ها را به 100 کاهش دادم
- از سه مدل استفاده کردم (دو تا درخت و یکی KNN)
- متغیری بر اساس داده های train درست کردم که هوشمندی دانشجویان را نشان می داد
- با استفاده از مدل ها validation set را پیش بینی کرده و با توجه به هوشمندی دانشجویان آنرا تضعیف یا تقویت کرده
- عمل voting را انجام داده و سپس تابع threshold را روی آن صدا کرده
- حدود 10 درصد دقت ما نسبت به voting موجود در فاز یک افزایش پیدا می کند

از آنجایی که من روند کد را می خواستم نشان دهم کد ها کمی پیچیده و نامفهموم شدند برای همین من برای test کردن داده های جدید در آخر قطعه کدی را کامنت کردم که می توان از آن استفاده کرد:

```
test = ...
X1_test = np.array(test_data['question_id'])
X2_test = np.array(test_data['user_id'])
for i in range(len(X2_test)):
    x_test.append([X1_test[i], X2_test[i], lis_topic_test[i]])
x_test = np.array(x_test)
processed_test_data = preprocess_test_data(x_test, mlb)
x_test_pca = pca.transform(processed_test_data)
y_pred_knn = knn.predict(x_test_pca)
y_pred_tree2 = tree2.predict(x_test_pca)
y_pred_tree1 = tree1.predict(x_test_pca)
voted = (y_pred_tree2+y_pred_knn+y_pred_tree1)/3
acc_voted = accuracy_score(y_test, multiply_and_threshold(voted, test_student_intel))
print("Accuracy of voted test is is :", acc_voted)
```

در اینجا تنها در ابتدا باید آدرس را به test داده. (فقط من فرض کردم فرم test مانند validation می باشد)