به نام خدا

دانشگاه تهران

دانشکده مهندسی برق و کامپیوتر

شبکه های عصبی و یادگیری عمیق

تمرین سری دوم

نام و نام خانوادگی	فاطمه سلیقه
شماره دانشجویی	۸۱۰۱۹۸۳۰۶
تاریخ ارسال گزارش	

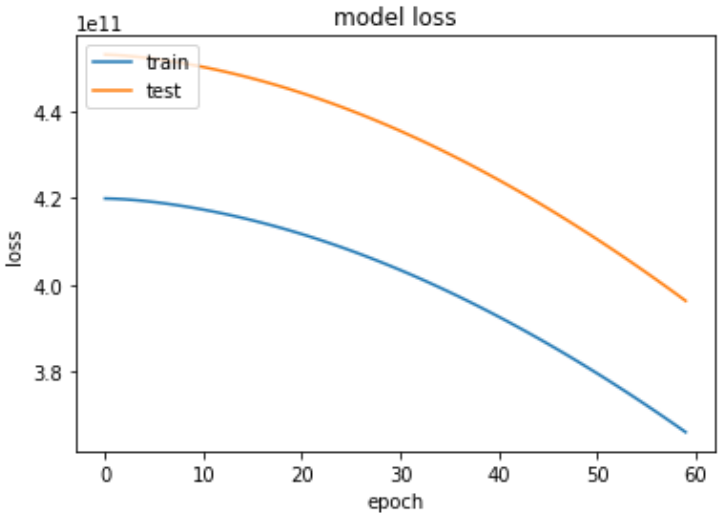
فهرست گزارش سوالات (لطفاً پس از تکمیل گزارش، این فهرست را به‌روز کنید.)

سوال ۲ – MLP

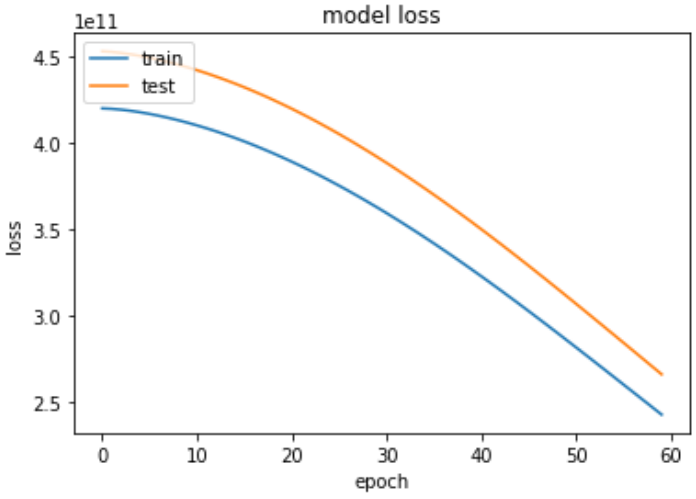
شبکه تک لایه ای :

تعداد نورون ها در سه مقدار ۲۰ و ۶۰و۱۲۸ تست شده اند و میزان خطا در ۱۲۸ کمتر است .

```
model = Sequential()
model.add(Dense(20, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(1))
```

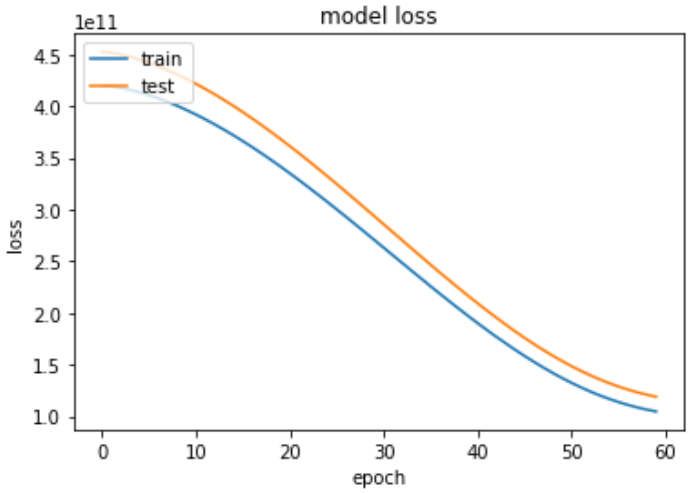



```
model = Sequential()
model.add(Dense(60, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(1))
```




```
model = Sequential()
model.add(Dense(128, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(1))
```

```
loss =  315837554525.0317 acc =  416556.2588191071
```



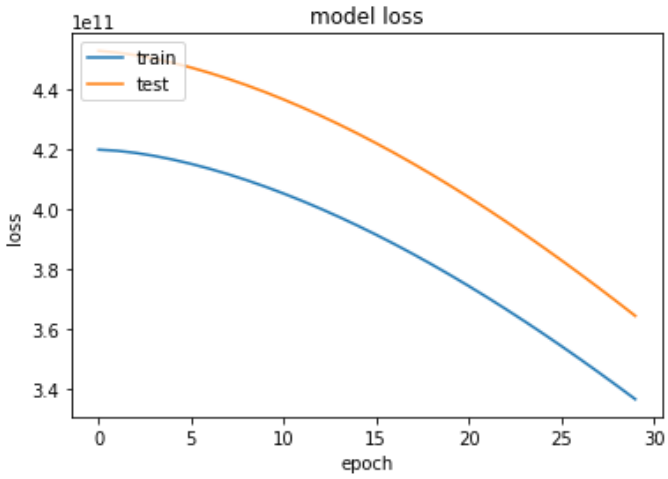
در حالت دو لایه ای :

در دو حالت (64,64) و (64,128) تست شده است

در مدل اول دقت پایینتر است اما خطا کمتر است.

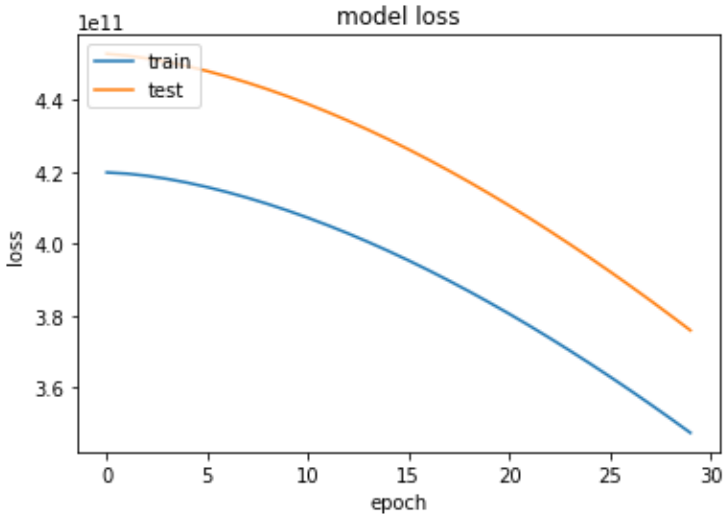
```
model2 = Sequential()
model2.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(64, activation='relu'))
model2.add(Dense(1))
```

```
loss =  364246222575.59656 acc =  467548.1944034814
```



```
model2 = Sequential()
model2.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(128, activation='relu'))
model2.add(Dense(1))
```

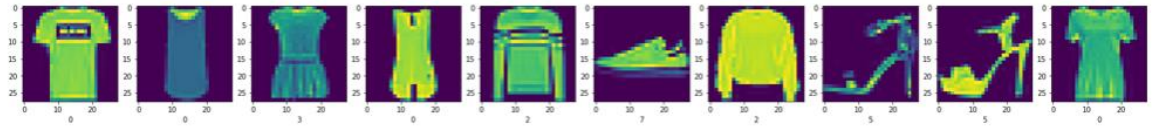
```
loss =  375833832216.5755 acc =  478993.26164555864
```



همان طور که در بالا مشاهده می شود هر دو مدل تک لایه و دو لایه مانند هم عمل می کنند ولی تک لایه خطای کمتری دارد.

سوال ۳ – MLP

در ابتدا ۱۰ ردیف اول را با برچسبشان نشان می دهیم :



شبکه به صورت زیر تعریف شده است :

```
def create_model(neurones1 = 16,neurones2 = 32):
    my_input=layers.Input(shape=(28,28,1))
    conv1=layers.Conv2D (neurones1,3,activation='relu', padding='same',strides=1)(my_input)
    pool1=layers.MaxPool2D(pool_size=2)(conv1)
    conv2=layers.Conv2D (neurones2,3,activation='relu', padding='same',strides=1)(pool1)
    pool2=layers.MaxPool2D(pool_size=2)(conv2)
    flat=layers.Flatten()(pool2)
    out=layers.Dense(10,activation = 'softmax')(flat)

    myModel=Model(my_input,out)

    print(myModel.summary())
    myModel.compile(optimizer='Adam', loss='categorical_crossentropy', metrics=['accuracy'])
    return myModel
```

الف)

در هنگام خواندن داده ها به دو دسته test ,train تقسیم می شوند :

```
(X_train, y_train), (X_test, y_test) = fashion_mnist.load_data()
```

سپس در هنگام آموزش شبکه می توان داده validation را جدا کرد :

```
trained_model=myModel.fit(X_train,y_train , batch_size=32 , epochs=10 , validation_split=0.2)
```

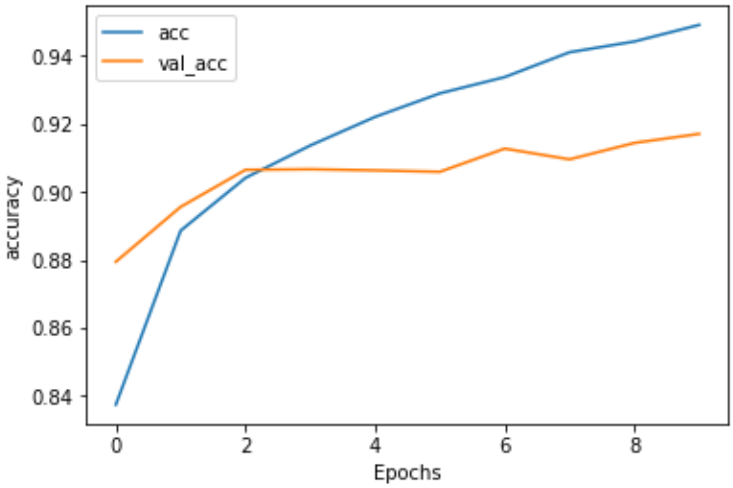
ب)

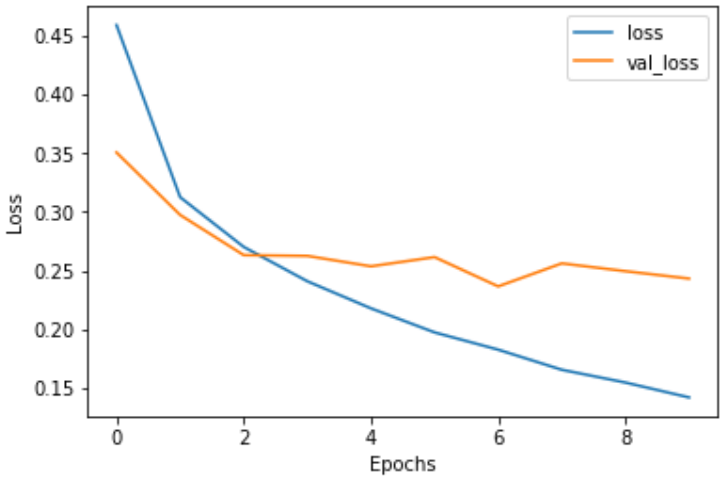
برای اینکه بفهیم کدام مقادیر برای پارامتر ها مناسبند از gridsearch استفاده شده است . که بهترین نتیجه در قسمت د نشان داده شده است .

یکی دیگر از حالتها :

تعداد نورون های لایه اول ۳۲ و لایه دوم ۶۴

```
loss =  0.2565839589655399 acc =  0.9112
```

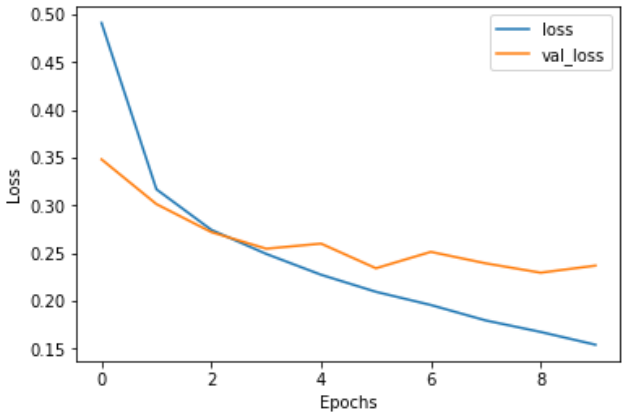


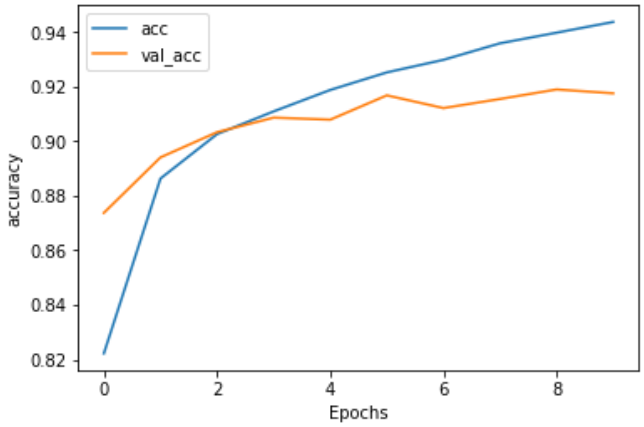


ج)

مدل با اندازه batch-size = 64

```
loss =  0.2569613894701004 acc =  0.9129
```





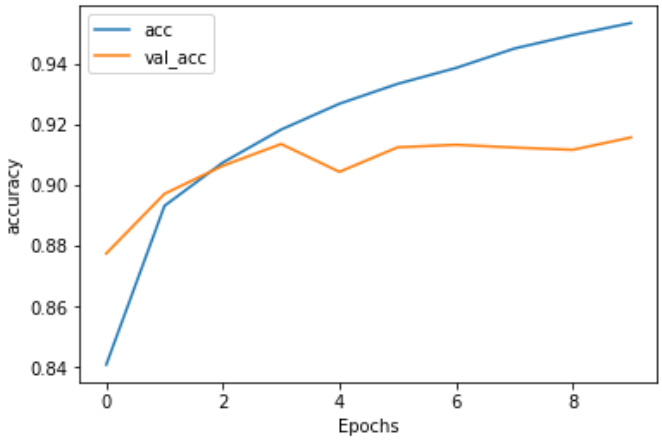
د)

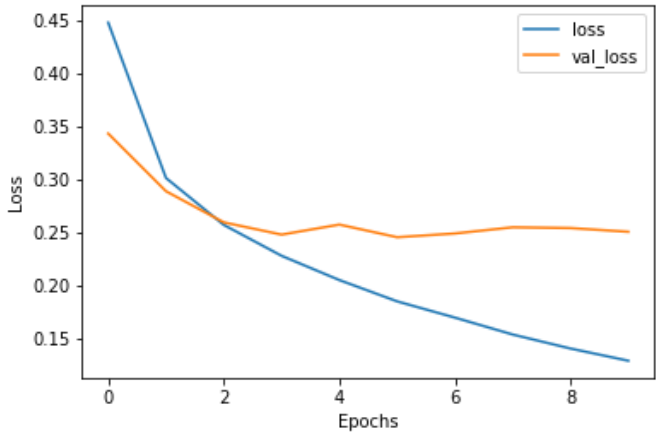
حال برای تست مقدار مناسب برای تعداد نورون ها و تعداد batch ها از gridSearch استفاده نموده (البته چون زمان اجرا طولانی بود فقط در یک epoch محاسبه شده است .) :

```
Best: -0.350424 using {'batch_size': 32, 'neurones1': 64, 'neurones2': 64}
-0.378016 (0.008031) with: {'batch_size': 32, 'neurones1': 32, 'neurones2': 32}
-0.365393 (0.002485) with: {'batch_size': 32, 'neurones1': 32, 'neurones2': 64}
-0.371473 (0.019191) with: {'batch_size': 32, 'neurones1': 64, 'neurones2': 32}
-0.350424 (0.015651) with: {'batch_size': 32, 'neurones1': 64, 'neurones2': 64}
-0.410020 (0.013901) with: {'batch_size': 64, 'neurones1': 32, 'neurones2': 32}
-0.393715 (0.012351) with: {'batch_size': 64, 'neurones1': 32, 'neurones2': 64}
-0.388357 (0.006958) with: {'batch_size': 64, 'neurones1': 64, 'neurones2': 32}
-0.365543 (0.009468) with: {'batch_size': 64, 'neurones1': 64, 'neurones2': 64}
-0.497706 (0.009804) with: {'batch_size': 256, 'neurones1': 32, 'neurones2': 32}
-0.467032 (0.009267) with: {'batch_size': 256, 'neurones1': 32, 'neurones2': 64}
-0.480698 (0.009724) with: {'batch_size': 256, 'neurones1': 64, 'neurones2': 32}
-0.450288 (0.008519) with: {'batch_size': 256, 'neurones1': 64, 'neurones2': 64}
```

حال با استفاده از بهترین پارامترهای معرفی شده و در epochs = 10 شبکه را اموزش داده و تست کرده و نتیجه به صورت زیر است :

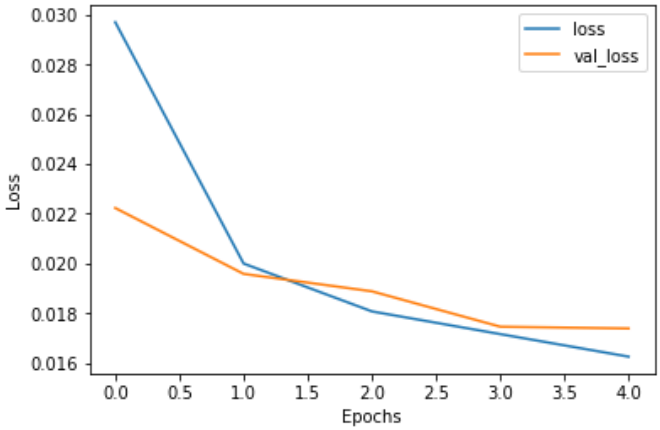
```
loss =  0.27310247611403465 acc =  0.9137
```

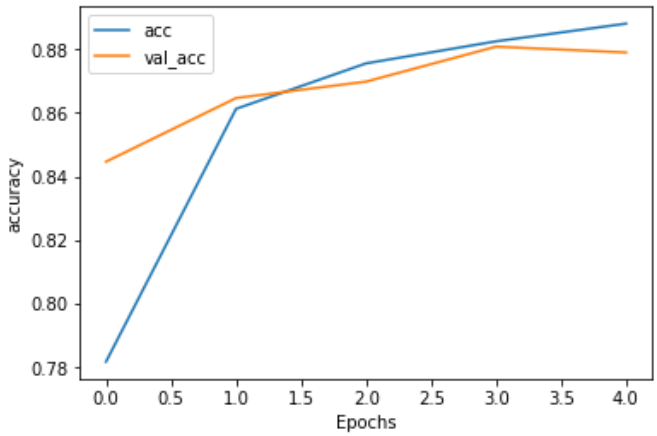

acc
val_acc
accuracy
0.94
0.92
0.90
0.88
0.86
0.84
0
2
4
6
8
Epochs


loss
val_loss
Loss
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0
2
4
6
8
Epochs

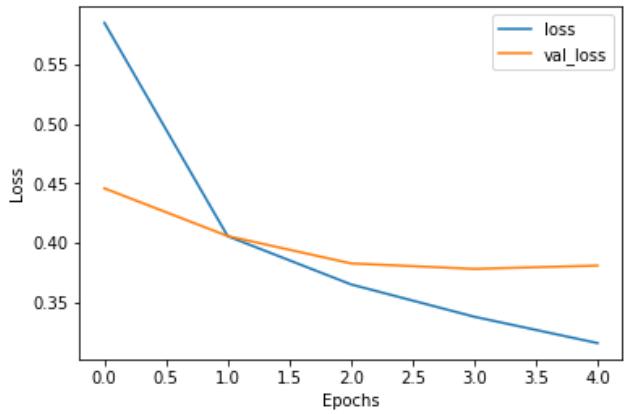
سوال ۴ – Dimensionality Reduction

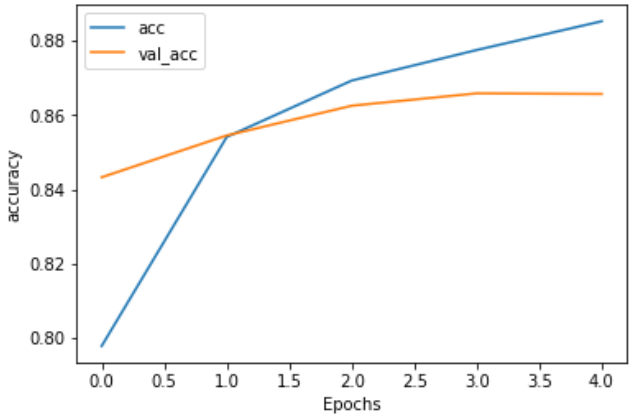
الف)



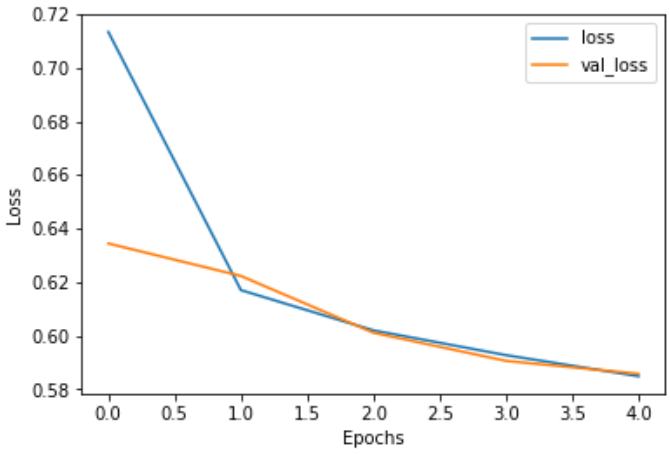


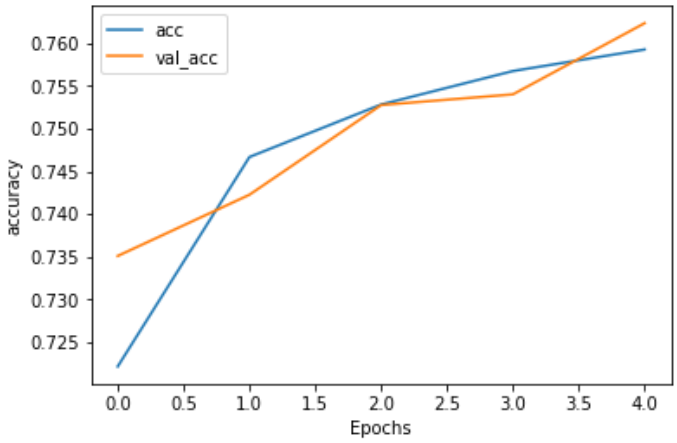
ب)


loss
val_loss
Loss
Epochs


acc
val_acc
accuracy
Epochs

ج)


loss
val_loss
Loss
Epochs



	دقت	خطا
سوال ۳	0.9137	0.2731
autoencoder	0.8772	0.0178
pca	0.856	0.4048
Cascaded RBM	0.7606	0.6026

همان طور که از نتایج دیده می شود در حالتی که کاهش بعد نداریم دقت بالاتر است .

سوال ۵ – مفاهیم

الف)زمانی که داده ها imbalanced باشند باعث می شود به صورت نادرستی accuracy افزایش پیدا کند . به این صورت که مثلا اگر فرض کنیم که دو کلاس داریم که اولی ۹۰ درصد داده ها را در بردارد و دارای میانگین ۰ و انحراف معیار ۲ است و کلاس دوم ۱۰ درصد داده ها را در بردارد و دارای میانگین ۱ و انحراف معیار ۱ است . بنابراین همیشه احتمال کلاس اول از دومی بالاتر است و در اکثر مواقع کلاس اول انتخاب می شود . بنابراین چون بیشتر داده ها هم مربوط به دسته اول هستند پس ما همیشه درست جواب می دهیم . این در حالی است که در دنیای واقعی که از هر دو کلاس به صورت متعادل داده وجود دارد باعث ایجاد خطای زیادی می شود .

بنابراین باید این مشکل را حل کنیم . یکی از راه های حل این مشکل undersampling و oversampling است . در undersampling از داده ای که بیشتر است نمونه گرفته و انها را حذف می کنیم . در oversampling از کلاس اول داده های بیشتری تولید می کنیم تا متعادل شوند .

ب) خیر زیرا دقت نشان می دهد که ما چه میزان درست گفته ایم . یعنی در چند درصد از داده ها انهایی را که عضو کلاس بوده اند و ما در ان کلاس قرار داده ایم و انهایی که در ان کلاس نبوده اند و ما درست تشخیص دادیم را درنظر می گیرد . اما در برخی موارد برای ما مهم است که ببینیم چند درصد از داده هایی که عضو یک کلاس نبوده اند را عضو آن کلاس قرار داده ایم یعنی FP . بنابراین دقت به تنهایی معیار مناسبی نیست .

ج) می توان correlation میان خروجی و ستون ها را پیدا کرد و انهایی که correlationکمتری دارند را حذف نمود . از راه های دیگر می توان اینگونه عمل کرد که هر بار یک ستون را حذف می کنیم و می بینیم که در کدام حالت بهتر عمل می کنیم و به همین ترتیب پیش می رویم . تا تعدادی از ستون ها را حذف کنیم .

د) می توان از روی ماتریس آشفتگی دقت ، precision و recall را بدست اورد

$$recall = \frac{tp}{tp + fn}$$

$$precision = \frac{tp}{tp + fp}$$

$$accuracy = \frac{tp + tn}{n}$$

ه) در نرمال کردن داده ها را به بازه [0,1] می بریم . اما در استاندارد سازی داده ها را به بازه با میانگین 0 و انحراف معیار ۱ می بریم .