

## 1. Forward Selection

ابتدا طبق خواسته ی سوال برای راحتی رنج قیمت را از 4 به 2 کلاس تبدیل کردیم (به این صورت که رنج 0 و 1 تبدیل به کلاس 0، 2 و 3 تبدیل به کلاس 1 شدند. همچنین آنها را اسکیل کردیم و رنج قیمت را از بقیه داده ها جدا کردیم.

در نهایت با توجه به معیار AUC با استفاده از متود فروارد سلکشن، فیچر هارا انتخاب می کنیم.

```
[10] from sklearn.metrics import roc_auc_score
def forward_selection(x, y):
    features = []
    scores = []
    left_features = x.columns
    for i in range(len(x.columns)):
        max_auc_score = 0
        best_feature = ""
        for feature in left_features:
            new_x = x[features + [feature]]
            x_train, x_test, y_train, y_test = train_test_split(new_x, y, test_size=0.2, random_state=21)
            logisticReg = LogisticRegression()
            logisticReg.fit(x_train, y_train)
            y_pred = logisticReg.predict(x_test)
            cur_auc_score = roc_auc_score(y_test, y_pred)

            if cur_auc_score > max_auc_score :
                max_auc_score = cur_auc_score
                best_feature = feature
            left_features = left_features.drop(best_feature)
            features.append(best_feature)
            scores.append(max_auc_score)

        best_index = scores.index(max(scores))

    result = {}
    result['scores'] = scores
    result['features'] = features
    result['best_features'] = features[:best_index+1]
    return result
```

0s completed at 17:16

5 تا از بهترین فیچرها در طی این پروسه انتخاب می شوند.

['ram', 'battery\_power', 'px\_height', 'px\_width', 'mobile\_wt']

## 2. Tain Model

با استفاده از رگرسیون و فیچر های انتخاب شده مدل را آموزش می دهیم.

```
4] print(classification_report(y_test, y_pred))
```

	precision	recall	f1-score	support
0	0.98	1.00	0.99	309
1	1.00	0.98	0.99	291
accuracy			0.99	600
macro avg	0.99	0.99	0.99	600
weighted avg	0.99	0.99	0.99	600

### 3. PCA

به دلیل اینکه در مرحله قبل ۵ تا فیچر انتخاب شد در این مرحله نیز PCA را با ۵ فیچر انجام می دهیم.

### 4. Train Model

با فیچرهای سلکت شده مدل رگرسیون را آموزش می دهیم.

	precision	recall	f1-score	support
0	0.98	0.99	0.99	309
1	0.99	0.98	0.98	291
accuracy			0.98	600
macro avg	0.99	0.98	0.98	600
weighted avg	0.99	0.98	0.98	600

### 5. —

### 6. Feature Engineering

#### a. binning

سه مدل مختلف دیتای قدرت باتری را دسته بندی می کنیم و انرا تبدیل به دیتای کتگوریکال می کنیم. آخرین مدل را برای مرحله بعدی انتخاب می کنیم.

#### b. one hot encoding

دیتای به دست آمده از مرحله قبل تنها دیتای کتگوریکال می باشد که آنرا انکود می کنیم.

#### c. transformation

برای ترنسفورم کردن دیتا با لگاریتم باید دیتا **right-skewed** باشد. نمودار

های تمام فیچر هارا (به جز باینری ها) رسم می کنیم. فقط ۳ فیچر هستند که میتوانیم آنها را ترنسفورم کنیم.

['fc', 'sc\_w', 'px\_height']

#### d. new feature

فیچر جدید حجم را با استفاده از فیچرهای قبلی درست می کنیم.

### 7. SVM

با استفاده از دیتای کتگوریکال ایجاد شده در بخش اول سوال قبل که در بخش دوم نومریکال شد مدل اول را آموزش می دهیم.

مدل دوم با استفاده از دیتای ترنسفورم شده ی ۳ فیچر آموزش داده می شود.

a) SVM with binned and encoded battery power

```

[58] from sklearn.svm import SVC

svc = SVC()
x_train, x_test, y_train, y_test = train_test_split(encoded_x, y, test_size=0.3, random_state=43)
svc.fit(x_train, y_train)
score = svc.score(x_test, y_test)
score

0.7883333333333333

```

b) SVM with new feature

```

[59] x_train, x_test, y_train, y_test = train_test_split(new_x, y, test_size=0.3, random_state=43)
svc.fit(x_train, y_train)
score = svc.score(x_test, y_test)
score

0.9416666666666667

```

0s completed at 17:16

مدل سوم را با استفاده از فیچرهای قبلی به علاوه فیچر جدید حجم آموزش دادیم  
مدل چهارم به صورت ترکیبی از فیچرهای تغییر یافته در مراحل قبلی آموزش داده می شود.

c) SVM with transformed data

```

[60] x_train, x_test, y_train, y_test = train_test_split(transformed_x, y, test_size=0.3, random_state=43)
svc.fit(x_train, y_train)
score = svc.score(x_test, y_test)
score

0.9016666666666666

```

d) SVM with binned and encoded battery power and new feature

```

[61] joined_x = new_x.drop(['battery_power'], axis=1)
joined_x['bat_po_low'] = encoded_x['bat_po_low']
joined_x['bat_po_medium'] = encoded_x['bat_po_medium']
joined_x['bat_po_high'] = encoded_x['bat_po_high']
joined_x['sc_w'] = transformed_x['sc_w']
joined_x['fc'] = transformed_x['fc']
joined_x['px_height'] = transformed_x['px_height']

[62] svc = SVC()
x_train, x_test, y_train, y_test = train_test_split(joined_x, y, test_size=0.3, random_state=43)
svc.fit(x_train, y_train)
score = svc.score(x_test, y_test)
score

0.77

```

در مقایسه این چهار مدل، مدل دوم که با فیچر جدید ساخته شد از دیگر مدل ها امتیاز بالاتری داشت و همچنین مدلی که با ترکیب فیچرهای تغییر یافته آموزش داده شد از دیگر مدل ها ضعیف تر عمل کرد.

- دیگر بخش ها در نوت بوک پاسخ داده اند.