

1- در ابتدا داده ها را اسکیل کردیم و x , y را جدا کردیم و طبق مطلب گفته شده کلاس ها را از چهار گروه به دو گروه گران و ارزان تبدیل کردیم (0, 1). سپس forward feature selection را پیاده سازی کردیم.

```
def forward_selection(x, y):
    features = []
    final_features = {'features': [], 'scores': []}
    rem_features = x.columns
    for i in range(len(x.columns)):
        max_score = 0
        best_feature = ""
        best_score = 0
        for feature in rem_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)
            logisticregression = LogisticRegression()
            logisticregression.fit(x_train, y_train)
            y_pred = logisticregression.predict(x_test)
            score = metrics.roc_auc_score(y_test, y_pred)

            if score > max_score :
                max_score = score
                best_feature = feature
                best_score = score
            rem_features = rem_features.drop(best_feature)
            features.append(best_feature)

        final_features['features'].append(best_feature)
        final_features['scores'].append(best_score)

    final_features['features_rank'] = range(len(x.columns))

    best_index = 0
    mx_feature = 0
    for i in range(len(final_features['scores'])):
        if final_features['scores'][i] > mx_feature:
            mx_feature = final_features['scores'][i]
            best_index = i

    final_features['best_features'] = final_features['features'][:best_index]
    return final_features
```

با استفاده از آن و مقایسه امتیاز ها 4 تا از بهترین فیچر ها انتخاب شدند.

```
'best_features': ['ram', 'px_height', 'battery_power', 'px_width']}
```

2- حال نوبت به آموزش می‌رسد. 20 درصد داده‌ها را برای تست جدا می‌کنیم. برای آموزش از رگرسیون

لجستیک استفاده می‌کنیم. و بعد از آن پیش بینی مقادیر تست را انجام می‌دهیم. سپس

classification report را چاپ می‌کنیم که در آن مقادیر precision, recall, f1-score,

accuracy آمده.

	precision	recall	f1-score	support
0	1.00	0.98	0.99	215
1	0.97	0.99	0.98	185
accuracy			0.98	400
macro avg	0.98	0.99	0.98	400
weighted avg	0.99	0.98	0.99	400

با توجه به شکل بالا متوجه می‌شویم که مدل به خوبی آموزش دیده و تقریباً درست پیش‌بینی کرده.

3- از PCA استفاده کرده و آن را روی داده ها فیت می‌کنیم با تعداد 4 component. دیتاست را تغییر

داده.

```
pca = PCA(n_components=4)
pca.fit_transform(x)
pca.n_components_
✓ 0.6s
```

4- از همان رگرسیون لجستیک برای آموزش داده ها استفاده می‌کنیم تا نتایج را بسنجیم و پس از آموزش

از classification report استفاده کرده تا گزارش کاملی بدهد و کار مقایسه راحت تر باشد.

	precision	recall	f1-score	support
0	0.99	0.99	0.99	215
1	0.98	0.99	0.99	185
accuracy			0.99	400
macro avg	0.99	0.99	0.99	400
weighted avg	0.99	0.99	0.99	400

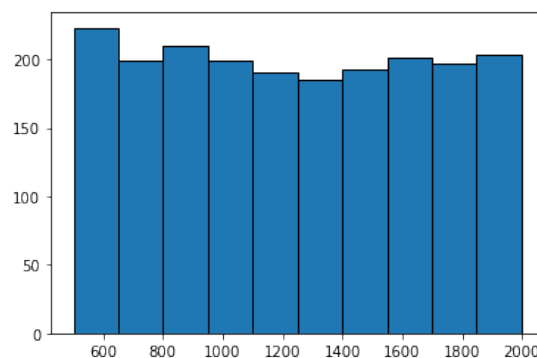
همانطور که مشاهده می‌شود تقریباً هردو به خوبی عمل کرده اند و نتایجی نزدیک به صد درصد داده اند.

5- ندارد.

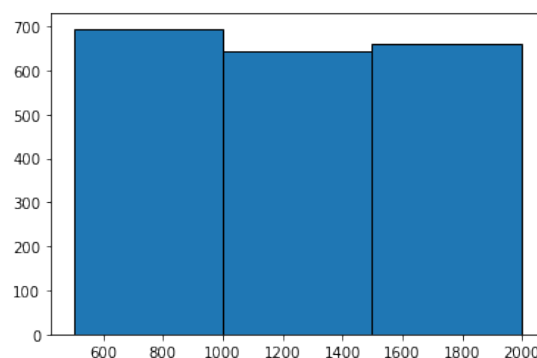
6-

(الف)

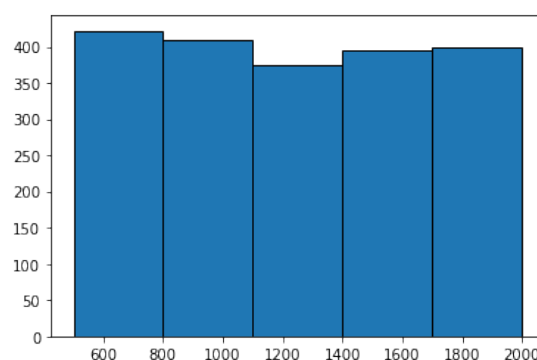
10 بین که اندازه هرکدام 149.7 است.



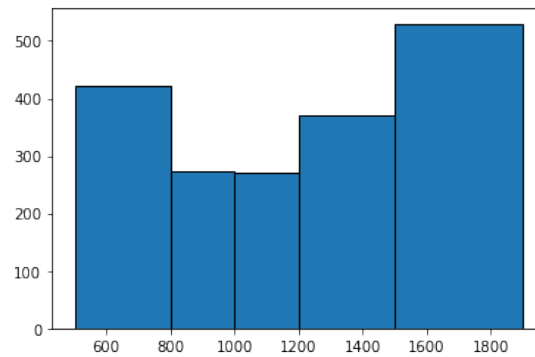
3 بین که اندازه هرکدام 499 است.



5 بین هم اندازه.



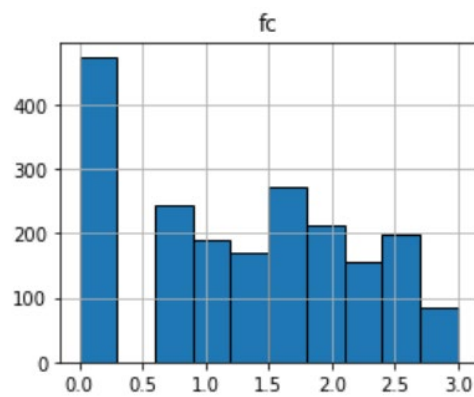
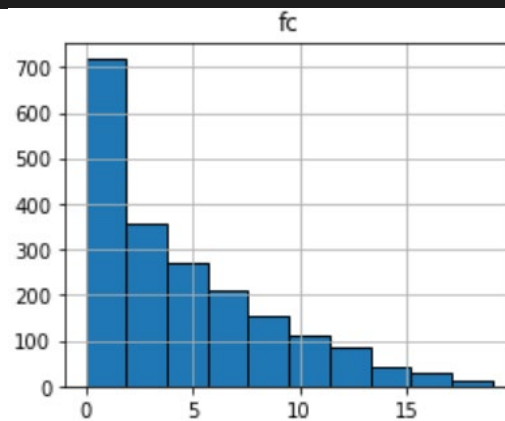
5 بین غیر هم اندازه.



ب) در فایل انجام شده.

ج) برای این فیچر ها انجام شده.

```
x['fc'] = np.log(x['fc']+1)
x['px_height'] = np.log(x['px_height']+1)
x['sc_w'] = np.log(x['sc_w']+1)
```



یک نمونه از قبل و بعد در اینجا مشاهده می شود.

د) فیچر حجم را ساختیم.

```
df['mobile_volume'] = df['sc_w'] * df['sc_h'] * df['m_dep']
```

```
df[['sc_h', 'sc_h', 'm_dep', 'mobile_volume']]
```

✓ 0.6s

	sc_h	sc_h	m_dep	mobile_volume
0	9	9	0.6	37.8
1	17	17	0.7	35.7
2	11	11	0.9	19.8
3	16	16	0.8	102.4
4	8	8	0.6	9.6
...
1995	13	13	0.8	41.6
1996	11	11	0.2	22.0
1997	9	9	0.7	6.3
1998	18	18	0.1	18.0
1999	19	19	0.9	68.4

2000 rows × 4 columns

همانطور که مشاهده میشود به خوبی حجم گوشی ها حساب شده و ستون آن به ستون های اصلی دیتافریم ما اضافه شده.

7- امتیازات مدل های ساخته شده توسط svm به صورت زیر است.

	precision	recall	f1-score	support
0	0.95	0.87	0.90	120
1	0.72	0.79	0.75	95
2	0.72	0.70	0.71	90
3	0.86	0.88	0.87	95
accuracy			0.81	400
macro avg	0.81	0.81	0.81	400
weighted avg	0.82	0.81	0.82	400

	precision	recall	f1-score	support
0	0.97	0.93	0.94	120
1	0.86	0.89	0.88	95
2	0.88	0.88	0.88	90
3	0.94	0.95	0.94	95
accuracy			0.91	400
macro avg	0.91	0.91	0.91	400
weighted avg	0.91	0.91	0.91	400

	precision	recall	f1-score	support
0	1.00	0.97	0.98	120
1	0.93	0.98	0.95	95
2	0.94	0.91	0.93	90
3	0.95	0.97	0.96	95
accuracy			0.96	400
macro avg	0.96	0.96	0.96	400
weighted avg	0.96	0.96	0.96	400

8, 9, 10- در فایل اصلی به طور کامل نوشته شده.