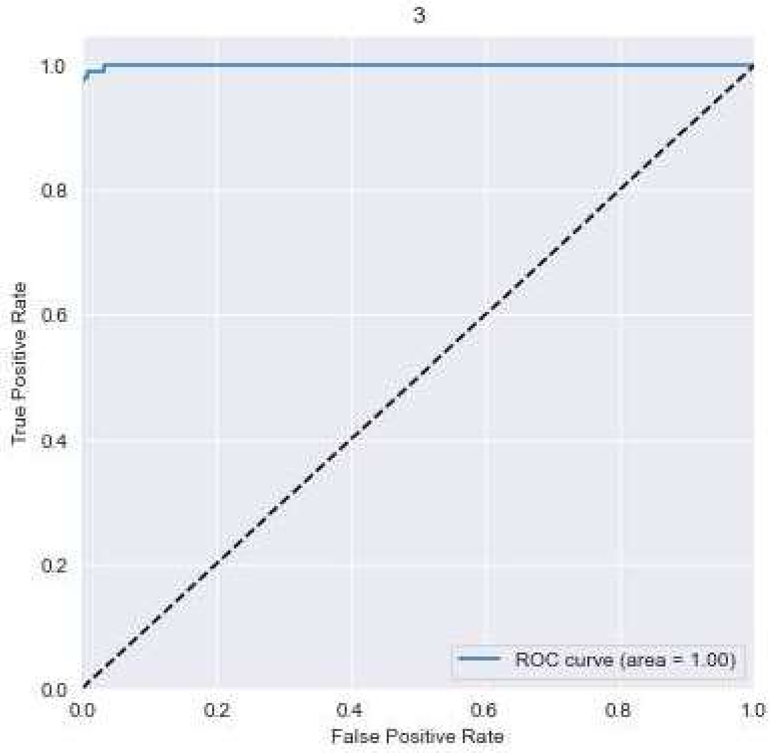


Test.csv

In [41]:

```
dft= pd.read_csv('test.csv')
dft.head()
```

Out[41]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt
0	1	1043	1	1.8	1	14	0	5	0.1	193
1	2	841	1	0.5	1	4	1	61	0.8	191
2	3	1807	1	2.8	0	1	0	27	0.9	186
3	4	1546	0	0.5	1	18	1	25	0.5	96
4	5	1434	0	1.4	0	11	1	49	0.5	108

5 rows × 21 columns

In [42]:

```
dft.sample(5)
```

Out[42]:

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile
414	415	658	1	2.4	0	0	1	38	1.0	
973	974	1796	1	0.5	0	0	0	44	0.2	
860	861	1951	0	2.1	0	5	0	22	0.5	
855	856	1270	0	0.5	0	1	1	62	0.5	
938	939	940	1	2.0	1	7	1	24	0.6	

5 rows × 21 columns

In [43]:

```
print(df.shape, '\n', dft.shape)
```

```
(2000, 21)
 (1000, 21)
```

focusing on train data

In [44]:

```
colt=dft.columns

colt
```

Out[44]:

```
Index(['id', 'battery_power', 'blue', 'clock_speed', 'dual_sim', 'fc',
       'four_g', 'int_memory', 'm_dep', 'mobile_wt', 'n_cores', 'pc',
       'px_height', 'px_width', 'ram', 'sc_h', 'sc_w', 'talk_time', 'three_
g',
       'touch_screen', 'wifi'],
      dtype='object')
```

data preparation

In [45]:

```
X=df.drop('price_range' , axis=1)
y=df.price_range

X_train , X_test , y_train ,y_test=train_test_split(X,y , test_size=0.2 , random_state=
```

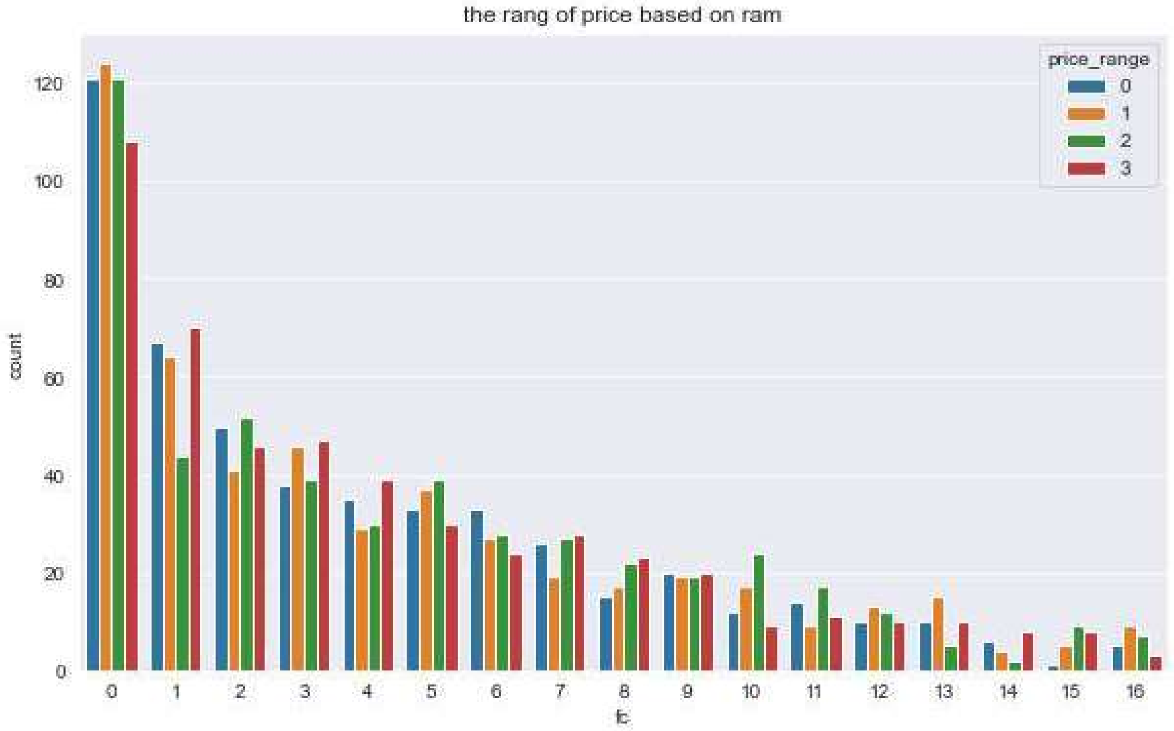
In [46]:

```
# #   remove the outlier


df=df[(df['fc']<17)]
```

In [47]:

```
plt.figure(figsize=(10,6))
sns.countplot(x='fc', data=df, hue= 'price_range').set_title('the rang of price based c
```



Baseline Model

In [48]:

```
def baseline_model(n_preds, pred):
    return pd.Series([pred for n in range(n_preds)])

# make baseline preds
baseline_preds = baseline_model(len(y_test), np.mean(y_train))
```

In [49]:

```
mse_bl=mean_squared_error(y_true=y_test,
                   y_pred=baseline_preds,
                   squared=False)
mse_bl
```

Out[49]:

```
1.1228495071134867
```

Linear Regression model