Predict the Right Price Range of a Mobile Phone

Data Capstone: Preliminary Results

Zhen Zhang

Introduction

- **Data Overview**
- **Exploratory Data Analysis and Feature Engineering:**
  - Statistics Analysis
  - Data Engineer”
    - Check Missing Value
    - Delete Some Variable
    - Encoding
  - Univariate Variable and Pair Variable Analysis
- **Models:**
  - Data Pre-processing
  - Machine Learning Part:
    - Naïve Bayes
    - Support Vector Machine (SVM)
    - Decision
  - Neural Network: Multilayer Perceptron (MLP)
- **Models Comparison and Selected**
- **Conclusion**

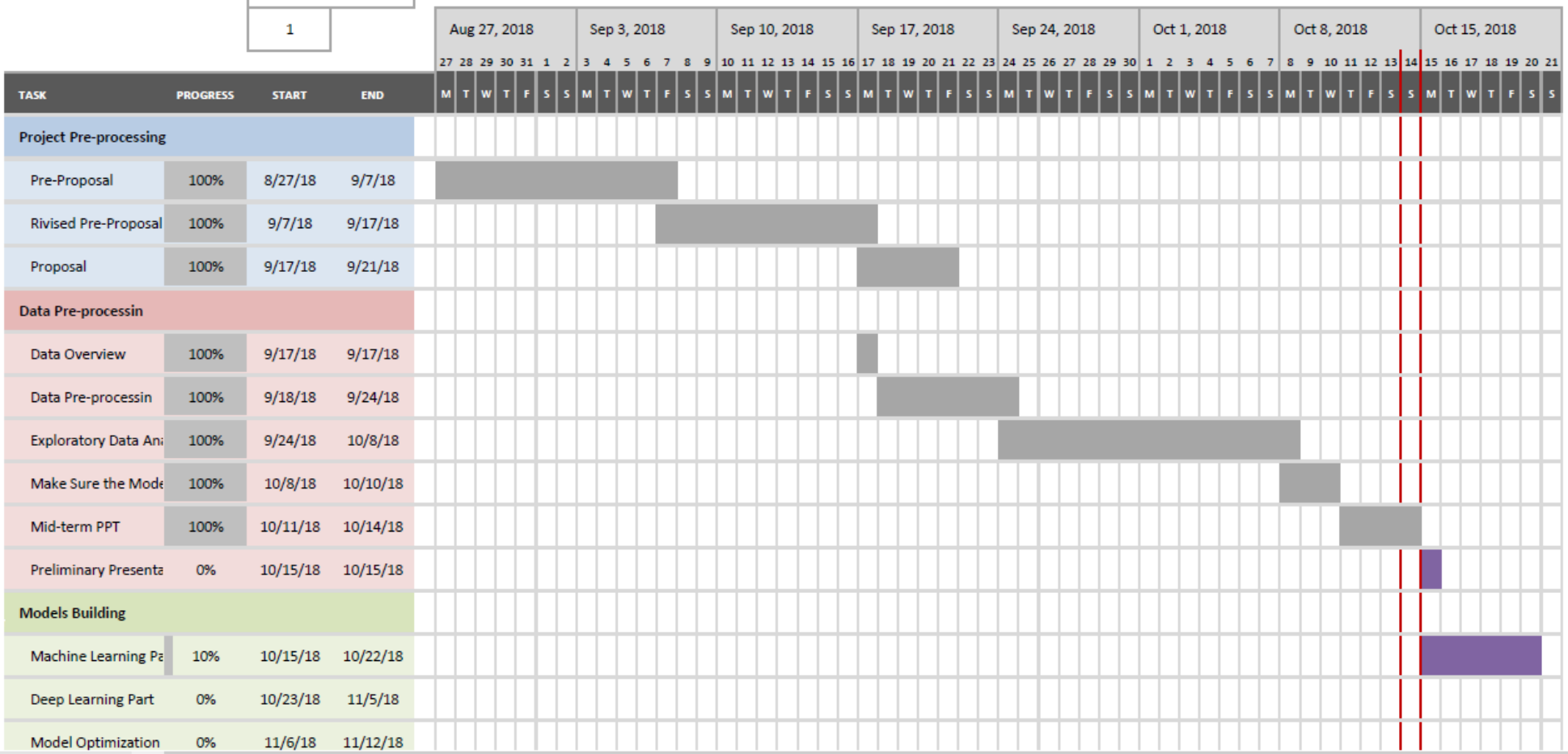
Gantt Chart

Data Science Capston Project

DATS 6501
Zhen Zhang

Fri, 9/21/2018
1

50%

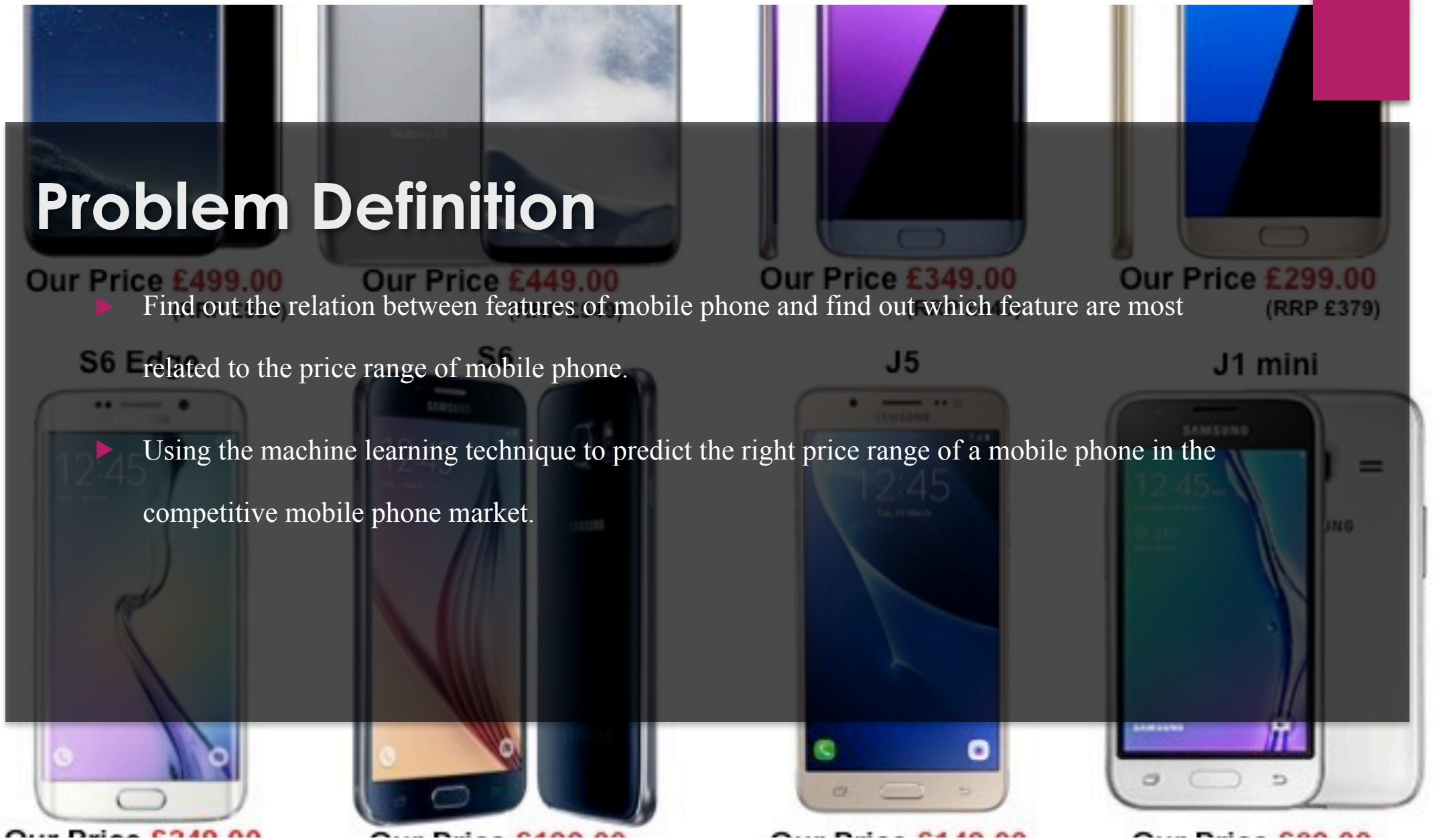


Dataset Introduction and Limitation

- The datasets were taken from Kaggle. There are two datasets, one is “train.csv” (2000 rows and 21 columns), another is “test.csv” (1000 rows and 21 columns).
- The initial database contains 21 features of mobile phone of varies companies.
- The limitation of this dataset is the dataset just provide us the price range of mobile phone instead of specific price of phone.

Problem Definition

- Find out the relation between features of mobile phone and find out which feature are most related to the price range of mobile phone.
- Using the machine learning technique to predict the right price range of a mobile phone in the competitive mobile phone market.



Data Overview

	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	n_cores	...	px_height	px_width	ram	sc_h	sc_w	talk_time
0	842	no	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19
1	1021	yes	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7
2	563	yes	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9
3	615	yes	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11
4	1821	yes	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15

(2000, 21)

	id	battery_power	blue	clock_speed	dual_sim	fc	four_g	int_memory	m_dep	mobile_wt	...	pc	px_height	px_width	ram	sc_h	sc_w	talk_time
0	1	1043	yes	1.8	1	14	0	5	0.1	193	...	16	226	1412	3476	12	7	2
1	2	841	yes	0.5	1	4	1	61	0.8	191	...	12	746	857	3895	6	0	7
2	3	1807	yes	2.8	0	1	0	27	0.9	186	...	4	1270	1366	2396	17	10	10
3	4	1546	no	0.5	1	18	1	25	0.5	96	...	20	295	1752	3893	10	0	7
4	5	1434	no	1.4	0	11	1	49	0.5	108	...	18	749	810	1773	15	8	7

(1000, 21)

Variable Name:	Variable Description:
battery_power	Total energy a battery can store in one time measured in mAh
blue	Has bluetooth or not
clock_speed	speed at which microprocessor executes instructions
dual_sim	Has dual sim support or not
fc	Front Camera mega pixels
four_g	Has 4G or not
int_memory	Internal Memory in Gigabytes
m_dep	Mobile Depth in cm
mobile_wt	Weight of mobile phone
n_cores	Number of cores of processor
pc	Primary Camera mega pixels
px_height	Pixel Resolution Height
px_width	Pixel Resolution Width
ram	Random Access Memory in Mega Bytes
sc_h	Screen Height of mobile in cm
sc_w	Screen Width of mobile in cm
talk_time	longest time that a single battery charge will last when you are
three_gHas	3G or not
touch_screen	Has touch screen or not
wifi	Has wifi or not
price_range	This is the target variable with value of 0(low cost), 1 (medium cost), 2(high cost) and 3(very high cost)

Data Dictionary

	count	mean	std	min	25%	50%	75%	max
battery_power	2000.0	1238.51850	439.418206	501.0	851.75	1226.0	1615.25	1998.0
clock_speed	2000.0	1.52225	0.816004	0.5	0.70	1.5	2.20	3.0
dual_sim	2000.0	0.50950	0.500035	0.0	0.00	1.0	1.00	1.0
fc	2000.0	4.30950	4.341444	0.0	1.00	3.0	7.00	19.0
four_g	2000.0	0.52150	0.499662	0.0	0.00	1.0	1.00	1.0
int_memory	2000.0	32.04650	18.145715	2.0	16.00	32.0	48.00	64.0
m_dep	2000.0	0.50175	0.288416	0.1	0.20	0.5	0.80	1.0
mobile_wt	2000.0	140.24900	35.399655	80.0	109.00	141.0	170.00	200.0
n_cores	2000.0	4.52050	2.287837	1.0	3.00	4.0	7.00	8.0
pc	2000.0	9.91650	6.064315	0.0	5.00	10.0	15.00	20.0
px_height	2000.0	645.10800	443.780811	0.0	282.75	564.0	947.25	1960.0
px_width	2000.0	1251.51550	432.199447	500.0	874.75	1247.0	1633.00	1998.0
ram	2000.0	2124.21300	1084.732044	256.0	1207.50	2146.5	3064.50	3998.0
sc_h	2000.0	12.30650	4.213245	5.0	9.00	12.0	16.00	19.0
sc_w	2000.0	5.76700	4.356398	0.0	2.00	5.0	9.00	18.0
talk_time	2000.0	11.01100	5.463955	2.0	6.00	11.0	16.00	20.0
three_g	2000.0	0.76150	0.426273	0.0	1.00	1.0	1.00	1.0
touch_screen	2000.0	0.50300	0.500116	0.0	0.00	1.0	1.00	1.0
price_range	2000.0	1.50000	1.118314	0.0	0.75	1.5	2.25	3.0

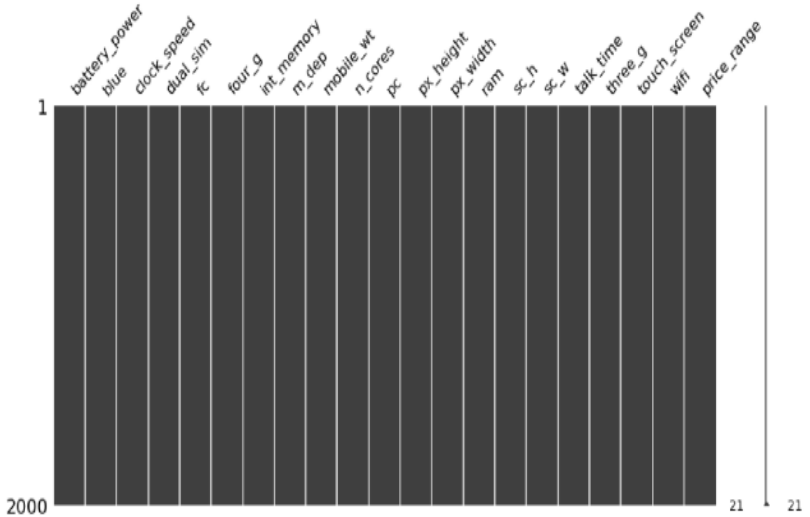
Initial Data Analysis

Exploratory Data Analysis and Feature Engineering
-Check Missing Value

isnull()

```
battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc               0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
price_range      0
dtype: int64
```

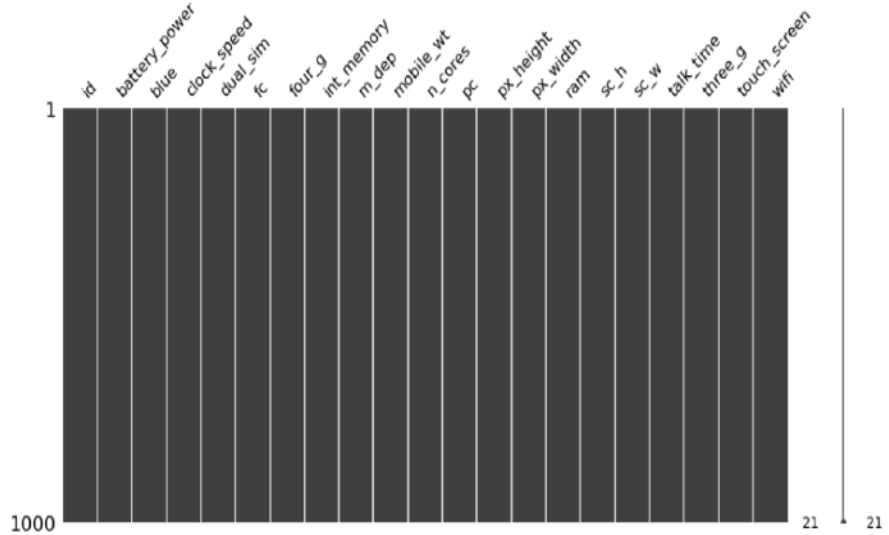
msno.matrix



isnull()

```
id               0
battery_power    0
blue             0
clock_speed      0
dual_sim         0
fc               0
four_g           0
int_memory       0
m_dep            0
mobile_wt        0
n_cores          0
pc               0
px_height        0
px_width         0
ram              0
sc_h             0
sc_w             0
talk_time        0
three_g          0
touch_screen     0
wifi             0
dtype: int64
```

msno.matrix



Handle with Categorial Variable

```
Data types and their frequency
int64      17
object      2
float64     2
dtype: int64
```

```
blue :
no     1010
yes     990
Name: blue, dtype: int64

wifi :
yes    1014
no      986
Name: wifi, dtype: int64
```

blue_no	blue_yes	wifi_no	wifi_yes
0	1	1	0
0	1	1	0
0	1	0	1
1	0	1	0
1	0	0	1

pandas.get_dummies( )

```
Data types and their frequency
int64      17
object      2
float64     2
dtype: int64
```

```
blue :
yes    516
no     484
Name: blue, dtype: int64

wifi :
yes    507
no     493
Name: wifi, dtype: int64
```

blue_no	blue_yes	wifi_no	wifi_yes
1	0	0	1
0	1	1	0
0	1	1	0
0	1	1	0
0	1	1	0

Narrow Down Columns-Delete Columns

1. Random Forest Classifier-Check Unimportant Variable

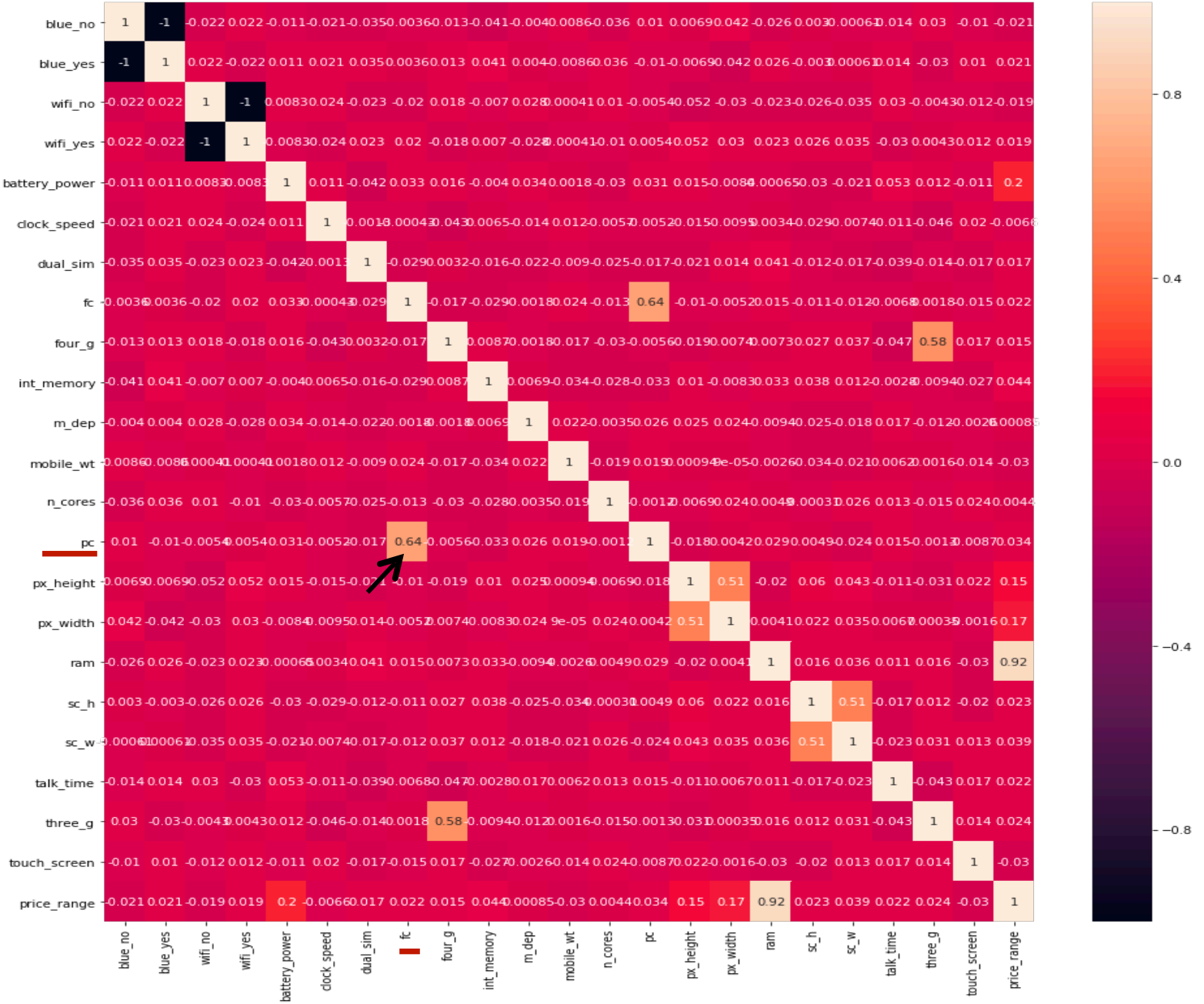
	ram	battery_power	px_width	px_height	mobile_wt	pc	int_memory	clock_speed	sc_w	sc_h
0	0.450633	0.073538	0.060289	0.052911	0.042498	0.037254	0.034922	0.03342	0.030152	0.029678

m_dep	n_cores	blue_no	touch_screen	wifi_no	dual_sim	four_g	three_g	blue_yes	wifi_yes
0.023871	0.022788	0.008494	0.007676	0.00669	0.006415	0.006391	0.005967	0.005806	0.005415

1. There are no useless columns

2. "ram" and "battery_power" are most important to the price range

Correlation Coefficient-Check Correlated Dependent Features



Drop “fc”

three_g
0
1
0
1
1

four_g
0
1
0
1
1

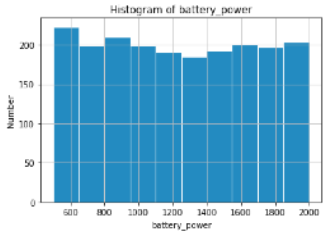
	battery_power	network	blue	dual_sim	fc	int_memory
0	842	0	0	0	1	7
1	1021	4	1	1	0	53
2	563	4	1	1	2	41
3	615	3	1	0	0	10
4	1821	4	1	0	13	44

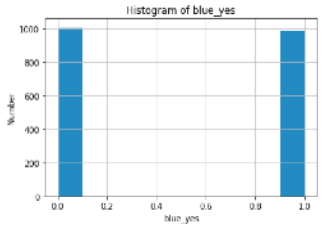
4G: 4
3G: 3
Neither 4G Nor 3G: 0

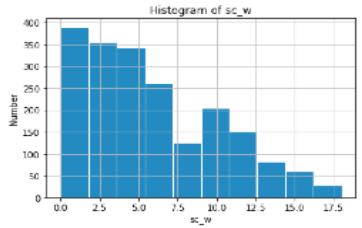
```
(2000, 21)

(1000, 21)
```

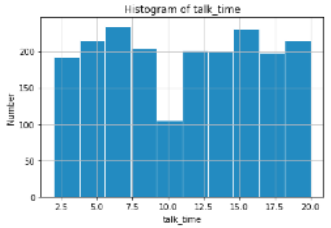
Data Engineering- Encoding

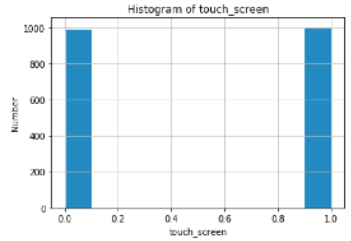
Visualization of Remaining Variable Univariate Analysis

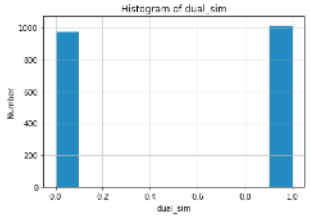


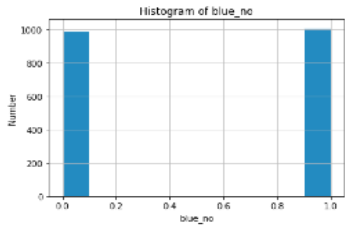


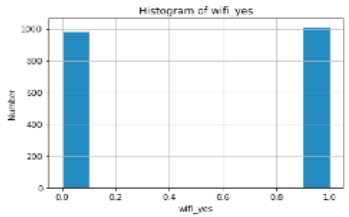


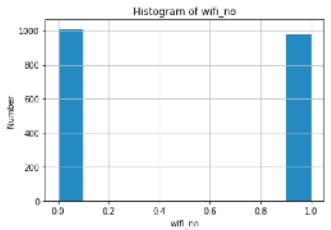


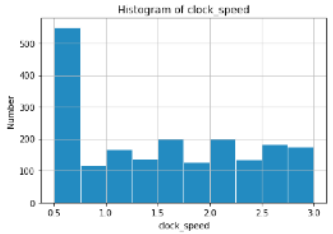


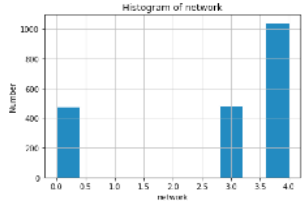


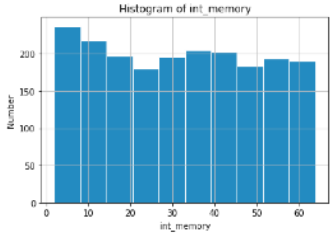


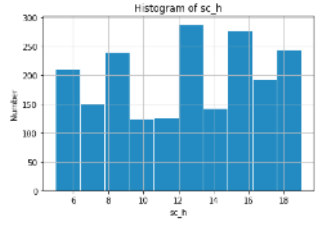


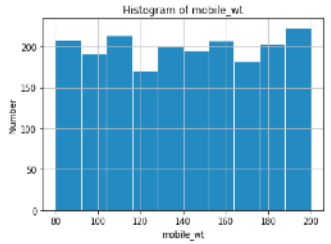


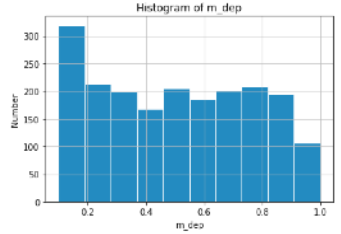


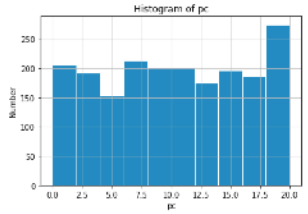


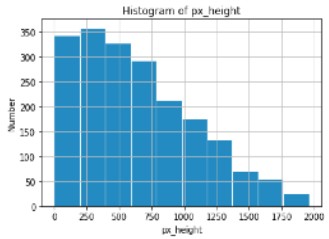


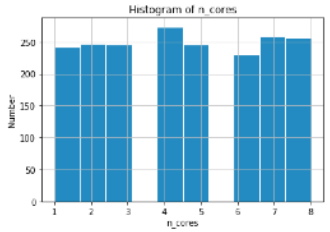


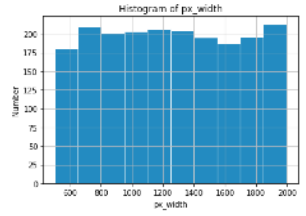


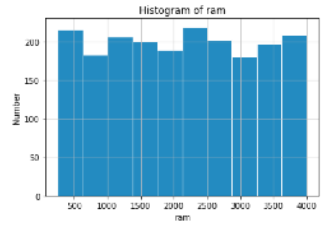






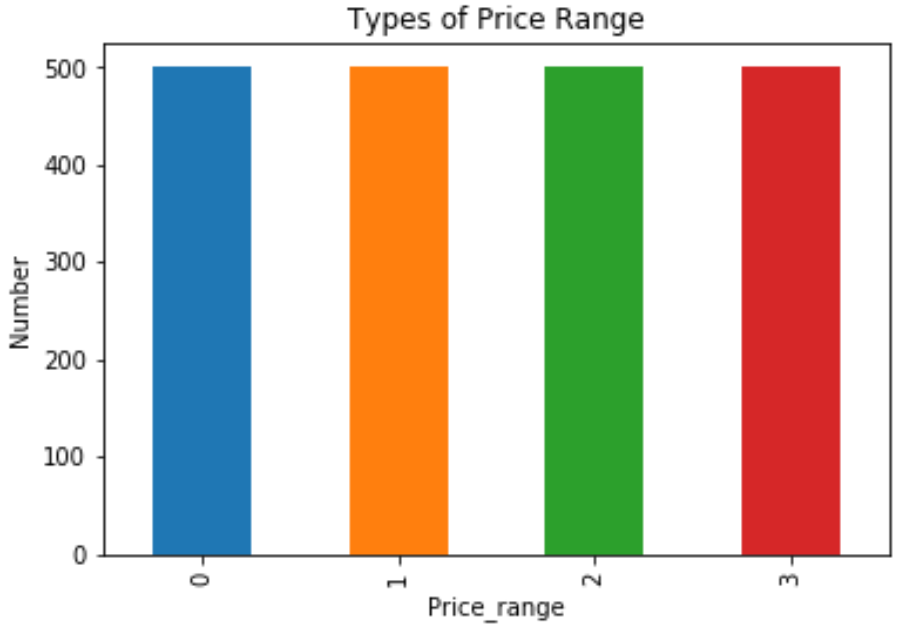


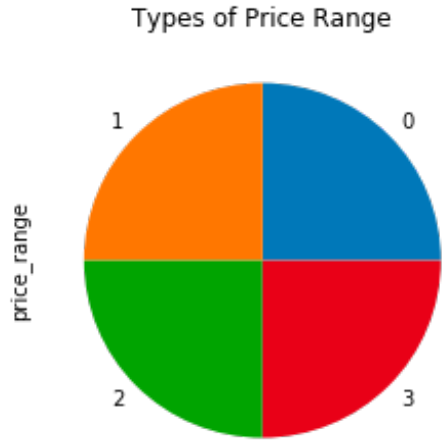




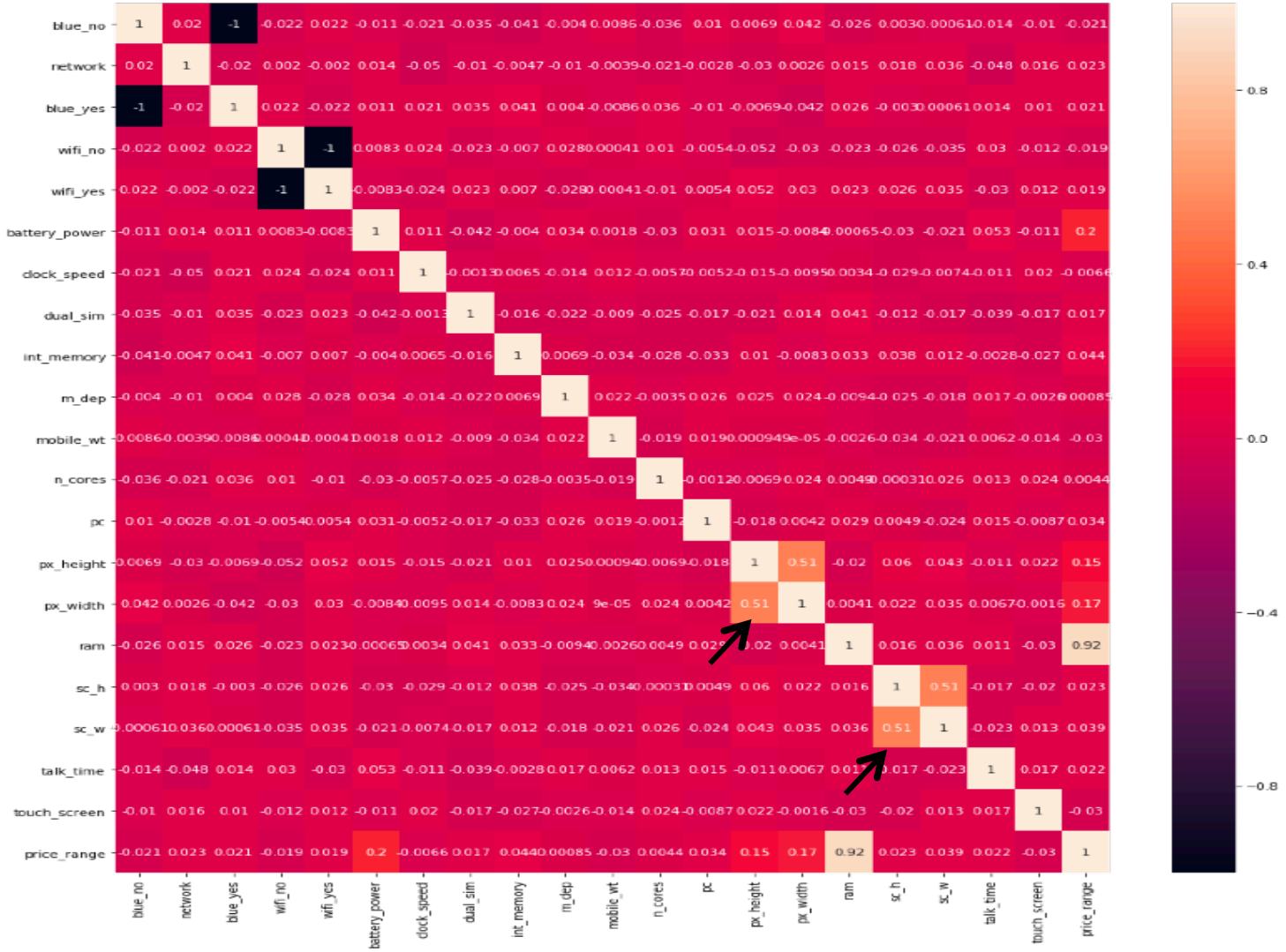
Univariate Variable Analysis- Target Variable

pandas.value_counts()



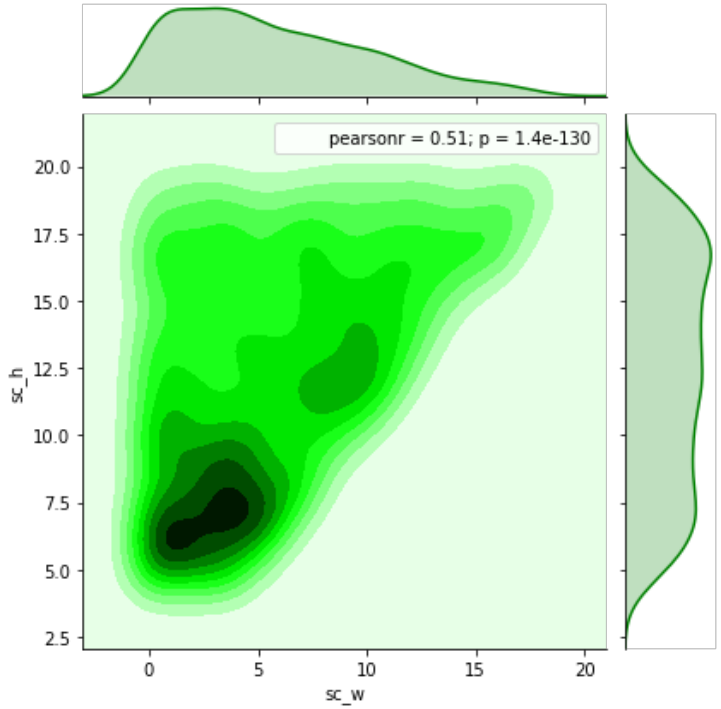


Pair-Wise Relationship

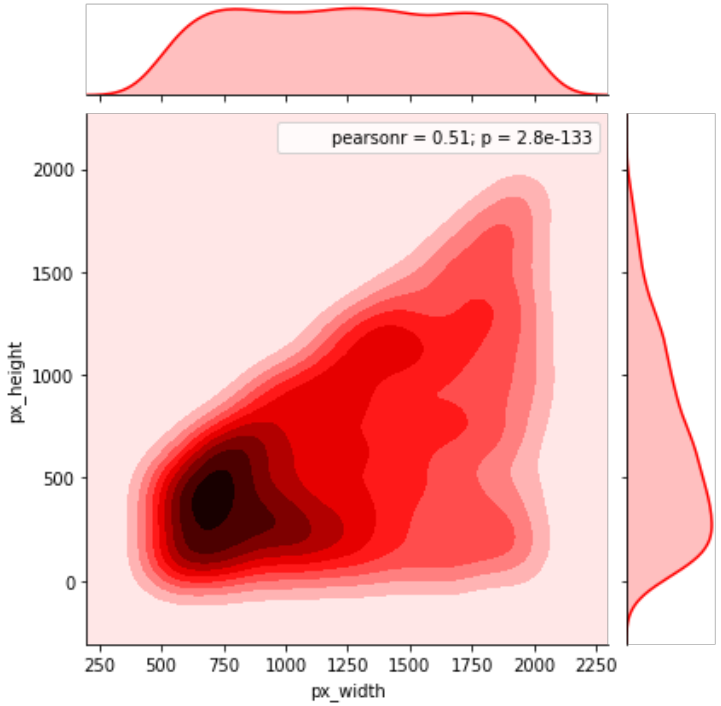


We see there are moderate correlation between "px_width" and "px_height", "sc_w" and "sc_h".

2) The distribution between 'px_width' and 'px_height'



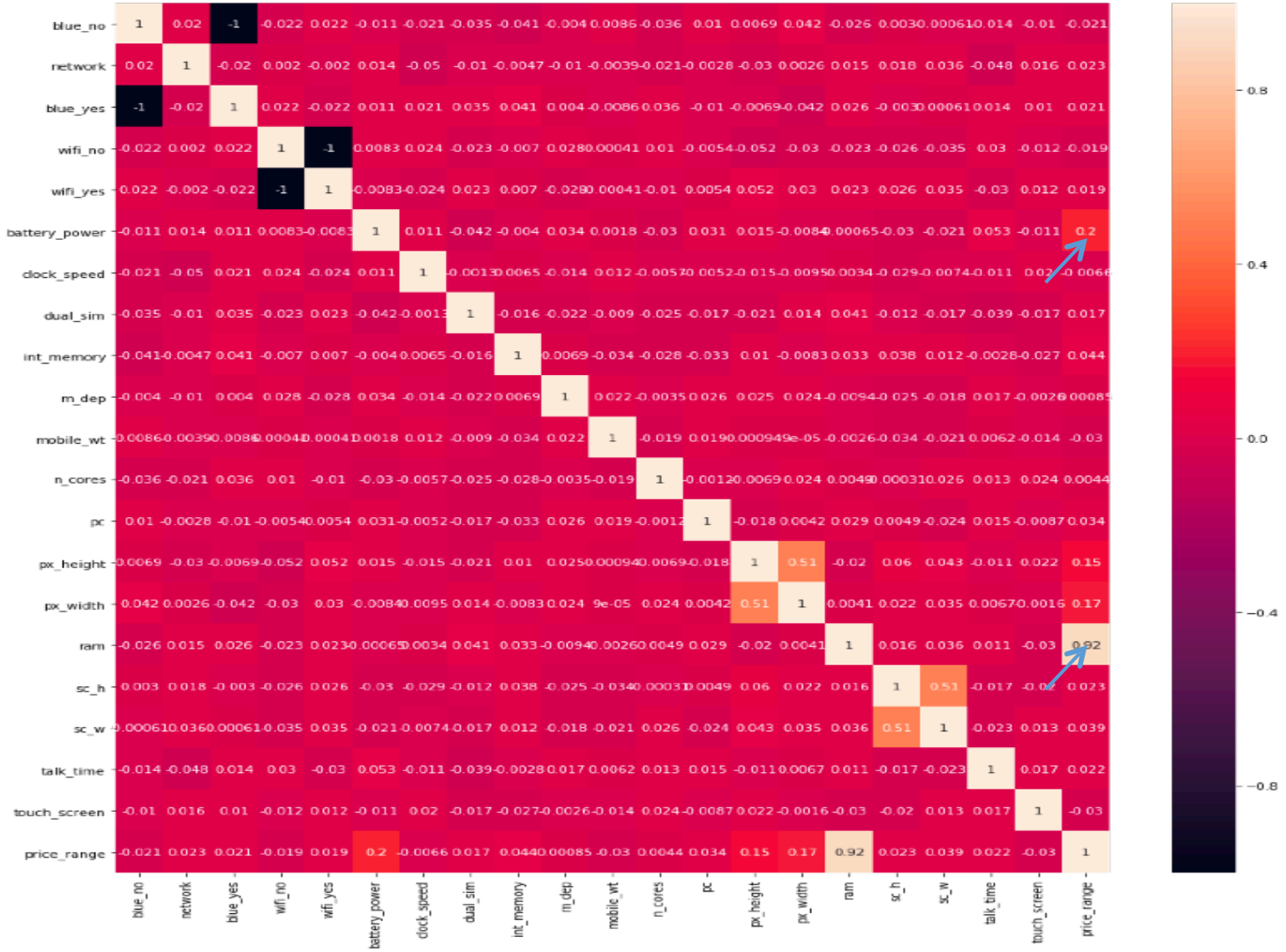
3)The distribution between "sc_w" and "sc_h"



Pair-Wise Relation

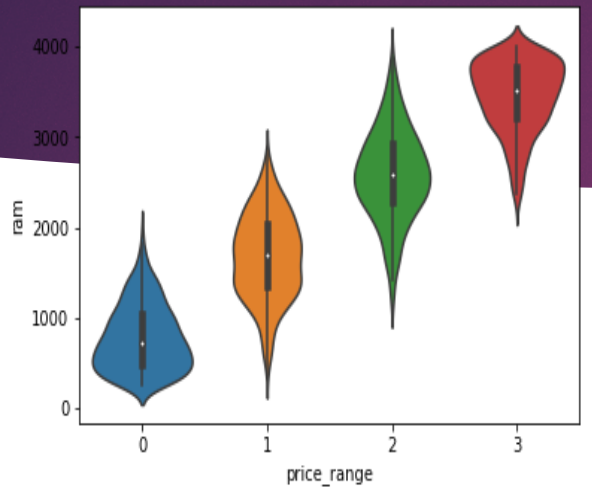
Jointplot

Pair-Wise Relationship

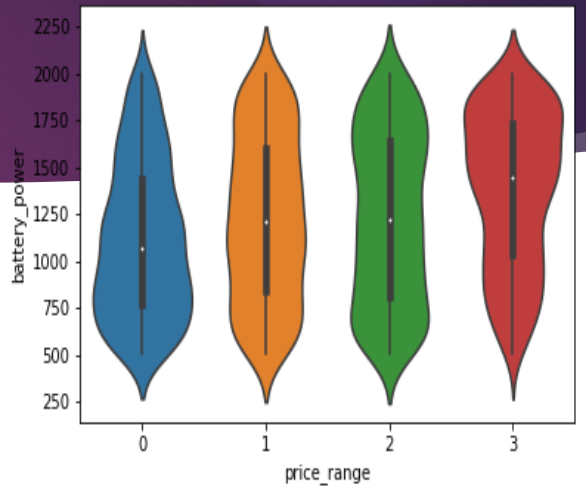


We can also clearly find the "ram", "battery_power" are most correlated to our target variable- "price_range".

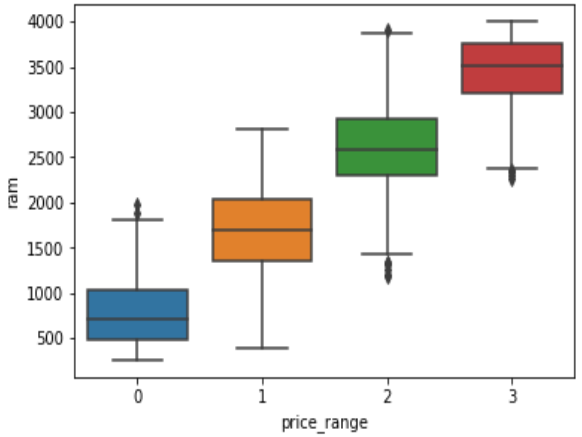
Pair-Wise Relation-Violin Plot and Boxplot

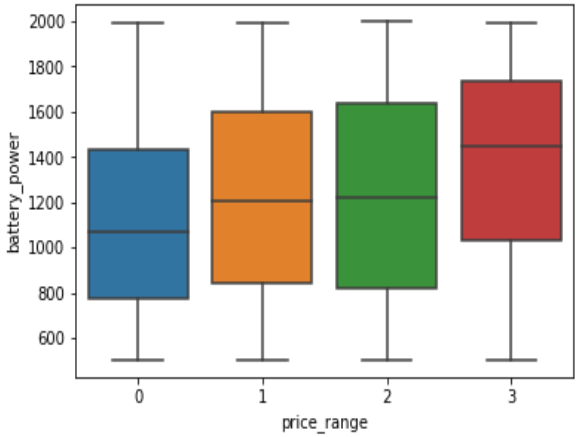


How does price affected by the ram?

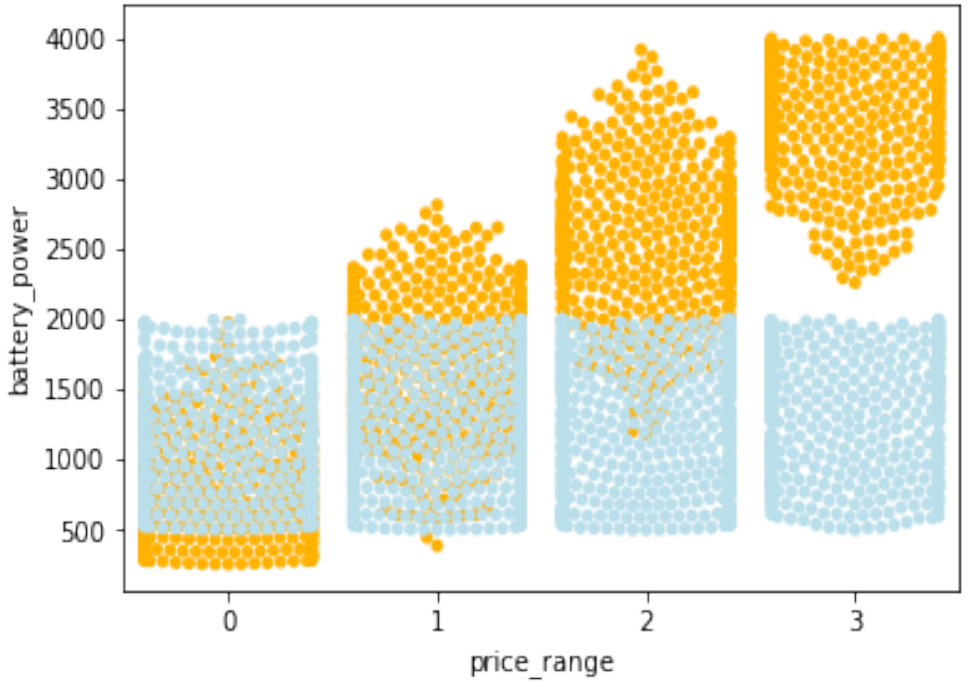


How does price affected by the battery power?





Indictors VS Target Value- Swarmplot



Orange: ram
Blue: battery_power

Models
Data Pre-processing

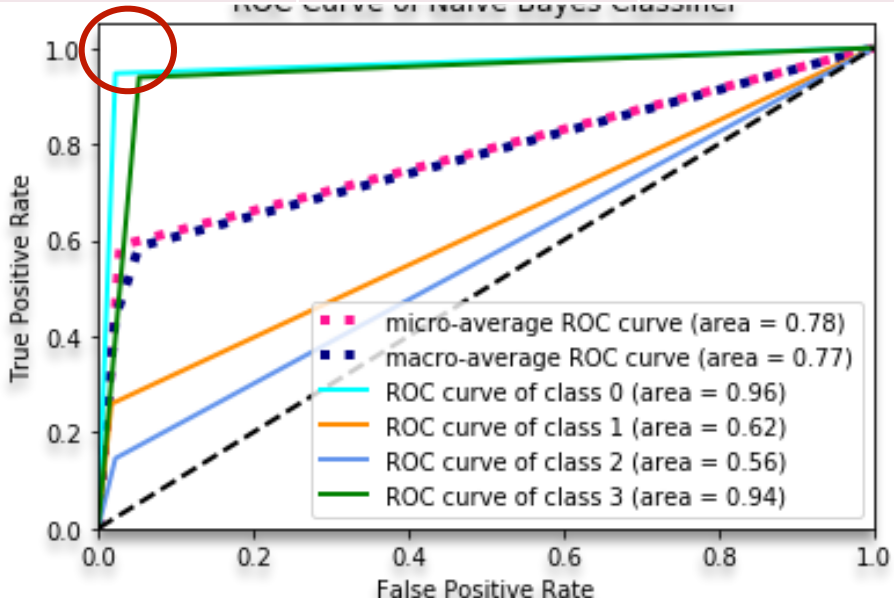
- **Set X and Y**
- **Get the training /testing dataset: 70% / 30%**
- **Convert the data into the same scale: use ‘StandardScaler()’**

Models-Machine Learning Part
Naive Bayes (Gaussian Naive Bayes algorithm)

- GaussianNB module
- ‘fit()’ function
- ‘predict()’
- Accuracy, precision, recall, f1 scores

Naive Bayes (Gaussian Naive Bayes algorithm)-Results

	Class 0	Class 1	Class 2	Class 3
Accuracy	93.38	71.85	71.52	92.02
Precision Score	93.38	74.05	72.48	88.76
Recall Score	93.38	71.85	71.52	92.02
F1-Score	93.38	72.93	72.00	90.36



Note: Binarize the output: set label_binarize()

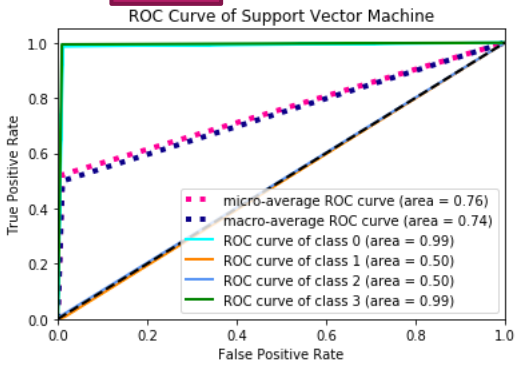
Other Models Building (Support Vector Machine (SVM), Decision Tree

- LinearSVC module, DecisionTreeClassifier
- ‘fit()’ function
- ‘predict()’
- Accuracy, precision, recall, f1 scores

Results for Other Models (SVM, Decision Tree)

SVM

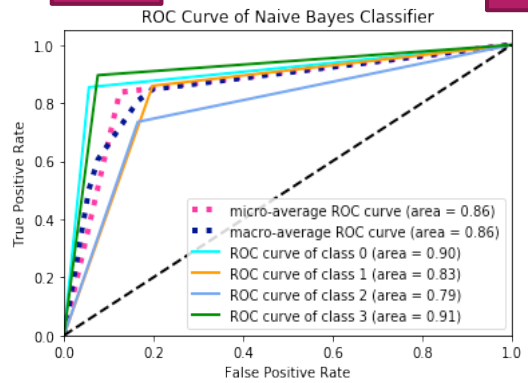
	Class 0	Class 1	Class 2	Class 3
Accuracy	100	72.59	72.85	99.39
Precision Score	94.97	73.13	78.57	97.01
Recall Score	100	72.59	72.85	99.39
F1-Score	97.42	72.86	75.76	98.18



SVM classifier has the better performance on price range 0 and price range 3.

Decision Tree

	Class 0	Class 1	Class 2	Class 3
Accuracy	82.12	71.85	60.26	79.14
Precision Score	87.32	61.39	62.76	83.23
Recall Score	82.12	71.85	60.26	79.14
F1-Score	84.64	66.21	61.49	81.13



Decision Tree classifier has the better performance on price range 0 and price range 3 compare to other price range。

Neural Network Part—
Multilayer Perceptron

Parameters:

- BATCH_SIZE = 50
- NUM_EPOCHS = 2000
- LEARNING_RATE = 0.005
- HIDDEN_NODE = 100
- SEED_START = 10
- LOSS_TARGET = 0.01
- SEED_STEP = 5

Net Architecture:

```
MLP(
  (inputLayer): Linear(in_features=20, out_features=100, bias=True)
  (hiddenLayer1): Linear(in_features=100, out_features=100, bias=True)
  (hiddenLayer2): Linear(in_features=100, out_features=100, bias=True)
  (hiddenLayer3): Linear(in_features=100, out_features=100, bias=True)
  (hiddenLayer4): Linear(in_features=100, out_features=100, bias=True)
  (hiddenLayer5): Linear(in_features=100, out_features=100, bias=True)
  (outputLayer): Linear(in_features=100, out_features=4, bias=True)
)
```

Performance of Multilayer Perceptron

	Class 0	Class 1	Class 2	Class 3
Accuracy	95	88	88	94
Precision Score	94.12	88.15	89.26	94.48
Recall Score	95.36	88.15	88.08	94.48
F1-Score	94.74	88.15	88.67	94.48

All index bigger than 80% which is good

Models Comparison

	Accuracy	Precision	Recall Score	F1 Score
Decision Tree	73.3425	73.6750	73.3425	73.3675
MLP	91.5175	91.5025	91.5175	91.5100
Naive Bayes	82.1925	82.1675	82.1925	82.1675
SVM	86.2075	85.9200	86.2075	86.0150

The performance of MLP is better than other models, and every indictors for MLP is bigger than 90%

Conclusion

1) We see there are strong correlation between "fc" and "pc" and there are moderate correlation between "px_width" and "px_height", "sc_w" and "sc_h".

2) We can also clearly find the "ram", "battery_power" are most related to our target variable- "price_range".

3) MLP is the best model to predict price range of mobile phone.

End….
Questions?

Thank You

Reference

- https://www.financialexpress.com/industry/gst-impact-on-mobile-phone-after-apple-iphone-asus-zenfone-3-zenfone-3-max-prices-cut-by-as-much-as-rs-3000-check-new-rates/745444/
- https://www.xcite.com/phones/mobile-phones.html
- https://www.mprc.co/mobile-phone-price-list/
- https://www.kaggle.com/iabhishekofficial/mobile-price-classification