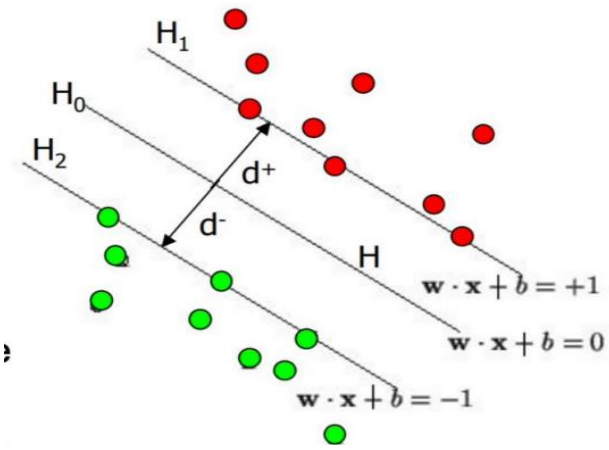
PREDICTING THE PRICE OF MOBILE USING SUPPORT VECTOR MACHINE

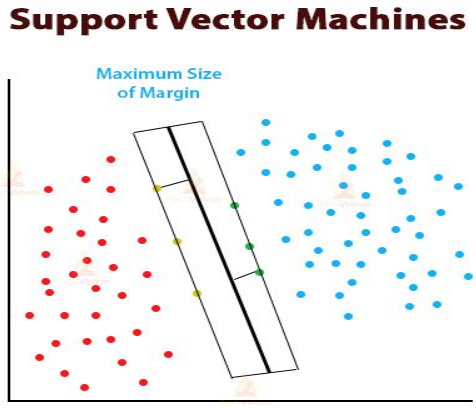
Briyana Rana
School of Engineering and Applied Sciences
University at Buffalo
Buffalo, NY, USA
brana@buffalo.edu

I. SUPPORT VECTOR MACHINES

A supervised machine learning approach called Support Vector Machine (SVM) is utilized for regression and/or classification. Classification is where it is most frequently employed, even though it may occasionally be quite beneficial for regression. Essentially, SVM finds a hyper-plane that makes a separation between the different kinds of data. In two-dimensional space, this hyper-plane is nothing more than a line. The total number of features and attributes in the dataset, N, is used to plot each dataset item using SVM in an N-dimensional space. Then, the data should be divided using the ideal hyperplane. By now, you must be aware that SVM can only do binary classification naturally (i.e., choose between two classes).



Here, the plane H0 has equation $\boldsymbol{w^T x} + b = 0$ and the points on H1 and H2 are the support vectors tips. Support vectors determines the weight. If we shift the decision boundary (hyperplane equation) for example H0 support vectors also moves. The goal is to find w and b such that it gives zero training error.



Special data points in the dataset are support vectors. They are the spots that are closest to the hyperplane and are in charge of building it. The hyperplane's location would change if these points were eliminated. Decision boundaries surround the hyperplane.
The support vectors help in decreasing and increasing the size of the boundaries. They are the main components in making an SVM.

A. Mathematical understanding:

$$\mathbf{a_0 + a_1X_1 + a_2X_2 + ... + a_nX_n}$$

Here a0 is the intercept of the hyperplane. Also, a1 and a2 define the first and second axes respectively. X1 and X2 are for two dimensions.
The equation below shows how the margins of support vector machines are related:

$$\phi(w) = \frac{1}{2} w^T w \; - \; minimize$$

$$Subject\ to\ d_i(w^T x + b) \geq 1 \; \forall \; i$$

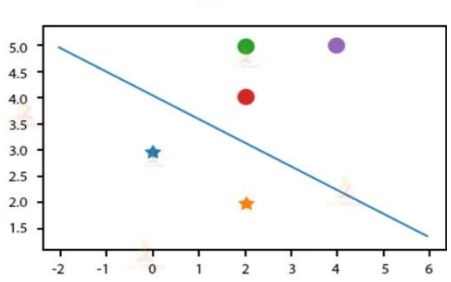
The cost is 0 if the predicted value and the actual value are of the same sign. If they are not, we then calculate the loss value. We also add a regularization parameter the cost function. The objective of the regularization parameter is to balance the

margin maximization and loss. After adding the regularization parameter, the cost functions look as below.

$$\text{minimize}_{w,b}\ \frac{1}{2}||w||^2$$
$$\text{subject to}\ \ y_i(w^T x_i + b) \geq 1 \quad i = 1, \ldots, N$$

The gradient margin:

$$d_H(\phi(x_0)) = \frac{|w^T(\phi(x_0)) + b|}{||w||_2}$$



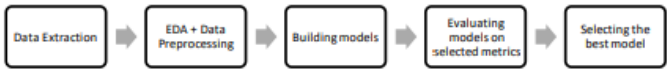
In the above diagram, the hyperplane is represented by the middle line. Because the dimension is 2-D, the hyperplane in this instance is a line. The hyperplane would have been a 2-D plane itself if we had a 3-D plane. The study of the hyperplane involves a great deal of mathematics.

II. EXPERIMENT MOTIVATION

Mobile phones are undoubtedly the most exciting invention of our generation. Connecting millions of people, no matter where they live. But this invention can a bit expensive. The price range of mobiles greatly depend on their feature set offered by the manufacturer. In this experiment we are focused on how these prices are related to the feature set offered.

III. METHODOLOGY

This experiment focuses on classification whether a data point of a certain attribute will have a higher price or lower price or somewhere in between of a mobile phone of a certain brand. The experiment primarily entails data extraction, which results in EDA and data preprocessing, developing our necessary data model in this case, support vector machines, evaluating the model's particular evaluation criteria, and finally choosing the best model.



A. *Data description:*

The dataset I have used id from Kaggle which contained around 21 columns and around 2000 rows.

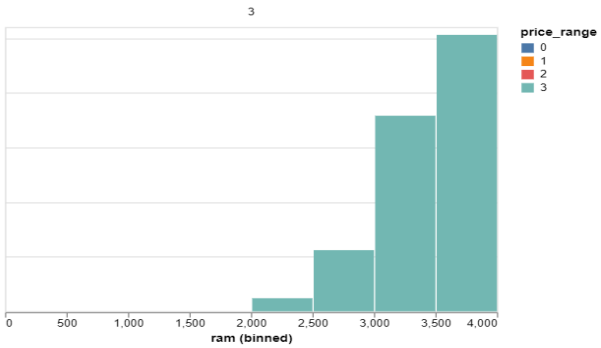
i. battery_power: Total energy a battery can store in one time measured in mAh
ii. blue: Has bluetooth or not
iii. clock_speed: speed at which microprocessor executes instructions
iv. dual_sim: Has dual sim support or not
v. fc: Front Camera mega pixels
vi. four_g: Has 4G or not
vii. int_memory: Internal Memory in Gigabytes
viii. m_dep: Mobile Depth in cm
ix. mobile_wt: Weight of mobile phone
x. n_cores: Number of cores of processor
xi. pc: Primary Camera mega pixels
xii. px_height: Pixel Resolution Height
xiii. px_width: Pixel Resolution Width
xiv. ram: Random Access Memory in Mega Bytes
xv. sc_h: Screen Height of mobile in cm
xvi. sc_w: Screen Width of mobile in cm
xvii. talk_time: longest time that a single battery charge will last when you are
xviii. three_g: Has 3G or not
xix. touch_screen: Has touch screen or not
xx. wifi: Has wifi or not
xxi. price_range: This is the target variable with value of 0(low cost), 1(medium cost), 2(high cost) and 3(very high cost).

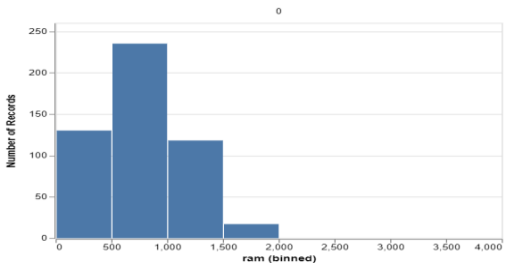
B. *Exploratory data analysis and data preprocessing*

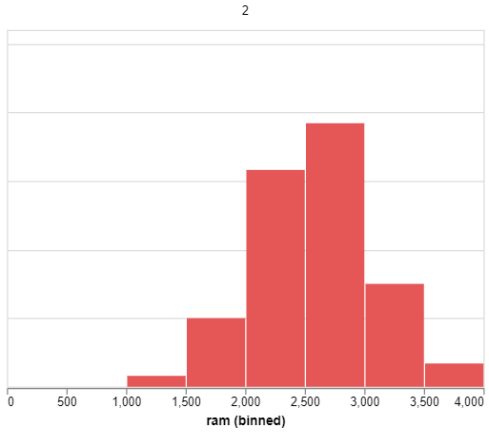
I have attached the following screenshot of the data to better look at the values:

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	th
842	0	2.2	0	1	0	7	0.6	188	2	...	20	756	2549	9	7	19	
1021	1	0.5	1	0	1	53	0.7	136	3	...	905	1988	2631	17	3	7	
563	1	0.5	1	2	1	41	0.9	145	5	...	1263	1716	2603	11	2	9	
615	1	2.5	0	0	0	10	0.8	131	6	...	1216	1786	2769	16	8	11	
1821	1	1.2	0	13	1	44	0.6	141	2	...	1208	1212	1411	8	2	15	

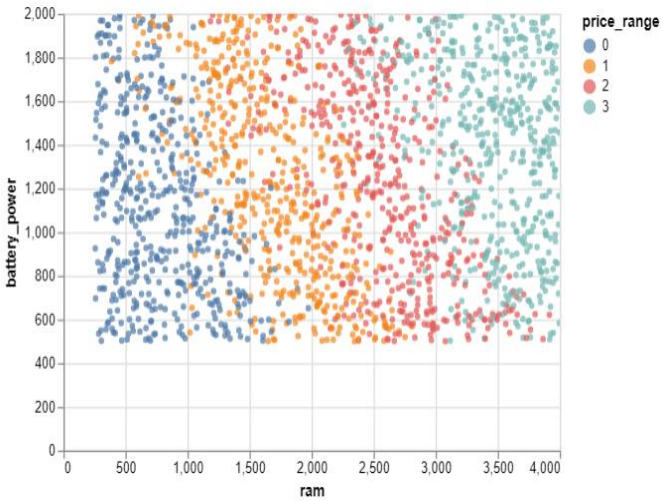
I took out the record of number of same size ram mobiles within the classified price ranges in the form of count plots as below:

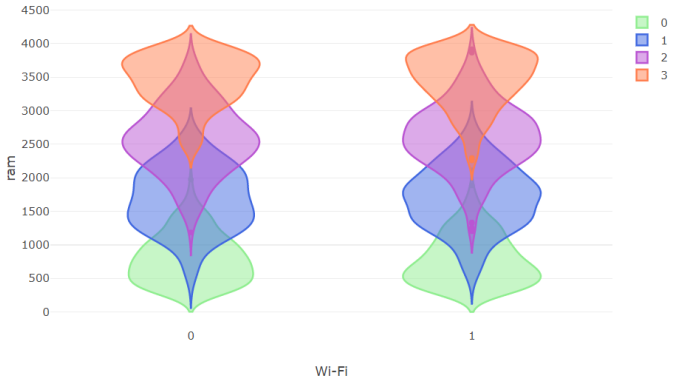




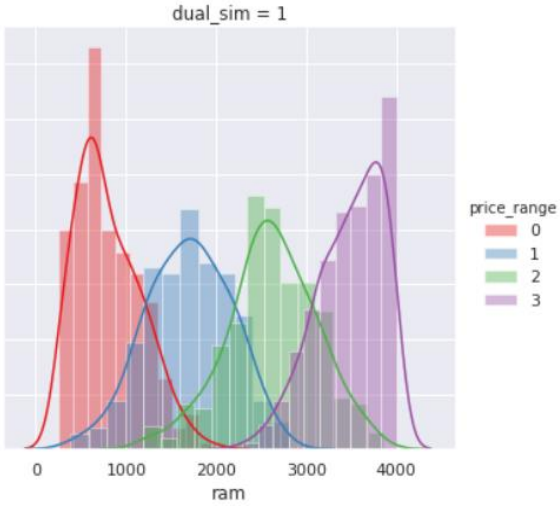


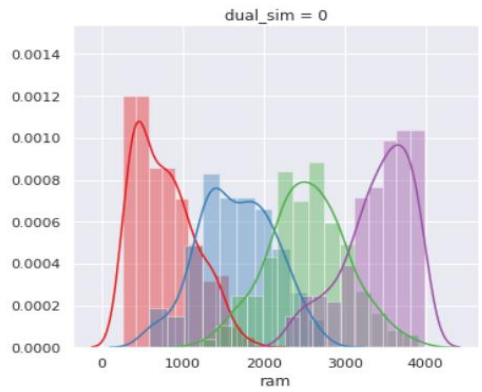
This is the scatter plot displaying the relation of battery and the ram with the price range. As you can see that battery has no relation with the price range but thus, we can say higher ram tends to higher price:





This graph also supports the fact that higher ram gives higher price, having wifi or no wifi has no difference in the price range.

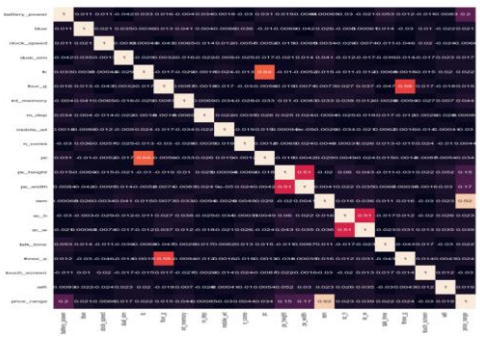




The ram and price are related to dual sim in a unique way. Having a dual sim has a higher price compared to having a single sim.

Data preprocessing steps:

The data has all non-null values and the attributes having data type as object were converted into labels to make an analysis for the relation with the target attribute. Also, dataset chosen has 2000 rows and 21 columns.



As observed in the correlation matrix, sc_h is correlated with sc_w hence we dropped sc_h, also the clock_speed is highly uncorrelated with our target value, thus it is also dropped. Similarly features other than ['ram','px_width', 'px_height','battery_power','sc_w'] were dropped. In 0 class(low cost) Ram values are changing between 0- 2000 megabytes.In 1 class(medium cost) Ram values are changing between 0-3000 megabytes.In 2 class(high cost) Ram values are changing between 1000-4000 mb.In 3 class(very high cost) Ram values are changing between 2000 and 4000 mb(mostly 3500-4000 mb). Now, the dataset is set to be trained on model to make predictions on test data. The data set was having few attributes with non-standard values thus using ceiling function from math library the column was converted into standard form. Later the dataset was split into Target variable Y and other features were kept together. Now SVM kernel will be run on train dataset to predict the output.

	ram	px_width	px_height	battery_power	sc_w	price_range
count	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000	2000.000000
mean	2124.213000	1251.515500	645.108000	1238.518500	5.767000	1.500000
std	1084.732044	432.199447	443.780811	439.418206	4.356398	1.118314
min	256.000000	500.000000	0.000000	501.000000	0.000000	0.000000
25%	1207.500000	874.750000	282.750000	851.750000	2.000000	0.750000
50%	2146.500000	1247.000000	564.000000	1226.000000	5.000000	1.500000
75%	3064.500000	1633.000000	947.250000	1615.250000	9.000000	2.250000
max	3998.000000	1998.000000	1960.000000	1998.000000	18.000000	3.000000

The statistical model of the extracted features with the targetfeature is displayed which has distinctness within the values.

C. SVM MODEL BUILDING

Looking at the data distribution, we can see that the data is a bit skewed and not normally distributed. I divided the train and test data into 75-25% and then around 80-20% to see if the value of the evaluation metrics is changing or not.

Looking at the values of the SVM score we can see that the values are on the higher side, stating that the model correctly predicted most of the values.

IV. RESULTS AND EVALUATION METRIC

```
from sklearn.svm import SVC
svm = SVC(random_state=1)
svm.fit(X_train,y_train)
print("train accuracy:",svm.score(X_train,y_train))
print("test accuracy:",svm.score(X_test,y_test))
```

```
train accuracy: 0.952
test accuracy: 0.968
```

A. Evaluation Metrics

The model's performance is evaluated based on precision, recall, F1 score and sensitivity, which are calculated as follows.

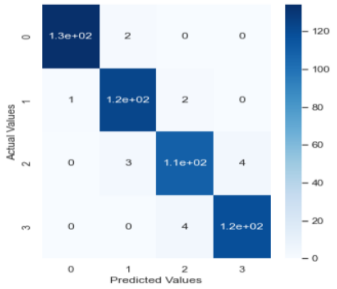
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall(or)Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

B. Confusion metrics

The confusion matrix below shows how correctly my algorithm identified the classes. With around 99% accuracy in all the class as shown in the confusion matrix below:



C. Precision , recall , F1 score and precision-recall plot and accuracy

The results for the evaluation metrics are shown below:

Recall	0.76
Precision score	0.88
Accuracy	0.96

V. CONCLUSION

To conclude, the model performed perfectly with around 94% accuracy and less error rate. Thus, the model will be able to predict the mobile price correctly.

VI. SUMMARY AND FUTURE WORK

The accuracy of the model is almost flawless, as can be seen from the assessment metrics above, and the confusion metrics support this. In addition, we require more data to increase accuracy and prevent overfitting/underfitting of the data.

VII. REFERENCES

i. Changyou Chen, CSE 574 Lecture Slides(2022).

ii. https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification

iii. https://en.wikipedia.org/wiki/Support_vector_machine