MTH 415
2/13/25
Sophia Hollins
Midterm Exam 1

1. Introduction

Mobile phones are being increasingly prevalent in modern day society. There were over 296 million smartphone users in the United States in 2023 and this number is projected to continue increasing over years ahead (Statistica). As companies compete to release devices with the latest features and technologies, consumers are faced with significant price differences across the market. This highlights the relevance of a classification model that can predict price of a mobile phone based on included features, particularly in the context of setting a price for a new phone in which misclassification can be detrimental.

Generally, the aim of this work is to build a model to make predictions in classification problems. In the context of this paper, the goal is to build a classification model that can successfully predict the price range for a mobile phone based on 20 features, some of which are binary. Features include physical factors such as phone height and weight, as well as internal factors such as, WIFI capability, camera features, and battery life. The dataset used in this project *Phone_csv.* It includes 20 predictor variables and 1 response variable. The response variable, *price_range*, is qualitative and has four categories presenting the different price ranges (0, 1, 2, and 3).

As mentioned, the model needs to predict price range, a qualitative variable with four classes. In instances where this is the case, there are multiple classification models that may be appropriate. A multinomial logistic regression model is one option. Other relevant classification models may include linear discriminant analysis, quadratic discriminant analysis, K nearest neighbor, and naïve Bayes. Ultimately, in this work I tested the models mentioned above to pick the one that performed the best on the dataset given. I found that multinomial logistic regression was able to provide the best outcome.

The paper is organized as follows: The *Methods* section presents the mathematical rationale for each technique used in this analysis, multinomial logistic regression. The *Results* section displays and summarizes results obtained from the model. The following section, Discussion, reveals the final classification model and presents practical implications. Finally, *References* and *Appendix*, contain works citied and R code respectively.

2. Methods

2.1 Multinomial Logistic Regression

A multinomial logistic regression model is often used in predictive modeling in classification problems when the response variable has more than two classes, as in the case of this problem where the response variable has four classes. In logistic regression, when making predictions, the

class of the response variable (*k)* is picked when the probability of that class *k* is higher than that of other class *k.*

A baseline class *K* is chosen initially, and *Equation 1* and *Equation 2* are used to make predict the probability that a given observation will fall in class *k* with respect to the baseline class.

$$Pr(Y = \mathrm{k}|X = \mathrm{x}) = \frac{e^{\beta_{k0}+\beta_{k1x1}+\ldots+\beta_{kpxp}}}{1 + \sum_{i=1}^{K-1} e^{\beta_{i0}+\beta_{i1x1}+\ldots+\beta_{ipxp}}} \qquad (Eq\ 1)$$

For *k*=1,…, K-1, and

$$Pr(Y = \mathrm{K}|X = \mathrm{x}) = \frac{1}{1 + \sum_{i=1}^{K-1} e^{\beta_{i0}+\beta_{i1x1}+\ldots+\beta_{ipxp}}} \qquad (Eq\ 2)$$

The log odds are obtained with *Equation 3*

$$ln\left(\frac{Pr(Y=k|X=x)}{Pr(Y=K|X=x)}\right) = \beta_{k0} + \beta_{k1x1} + \beta_{k2x2} + \ldots + \beta_{kpxp} \qquad \text{(Eq 3)}$$

Equation 4 below shows the softmax formulation of multinomial logistic regression. In this case, rather than assigning K to a baseline class, normalize to make the probabilities of each class *k* equal to one. Similarly to *Equation 2* and *Equation 3***,** *Equation* 4 is used to estimate probability that a given observation will be classified into class *k* (Gareth et al., 2021).

$$Pr(Y = \mathrm{k}|X = \mathrm{x}) = \frac{e^{\beta\prime_{k0}+\beta\prime_{k1x1}+\ldots+\beta\prime_{kpxp}}}{\sum_{i=1}^{K} e^{\beta\prime_{i0}+\beta\prime_{i1x1}+\ldots+\beta\prime_{ipxp}}} \qquad (Eq\ 4)$$

$$ln\left(\frac{Pr(Y=k|X=x)}{Pr(Y=k\prime|X=x)}\right) = \beta'_{k0} + \beta'_{k1x1} + \beta_{k2x2} + \ldots + \beta'_{kpxp} \qquad \text{(Eq 5)}$$

Where $\beta'_{kp} = (\beta_{kp} - \beta'_{kp})$ which is provided automatically in RStudio. Additionally, further information about the above process can be obtained in the *Introduction to Statistical Learning* from chapter 4 which is cited in the references.

2.2 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is another model considered for this report. Generally, LDA is used over linear regression when there is significant separation between classes, sample size is small, and it may be preferred as it can extend naturally to response variables with $k > 2$.

LDA with multiple predictor variables assumes that predictor variables (X) share a common covariance matrix and that X=(X1, X2,…,Xp) come from normal multivariate distribution; $X \sim N(\mu, \Sigma)$ (Gareth et al.).

$$\delta_k(x) = x^T\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k + log(\pi k) \qquad (Eq\ 6)$$

According to the *Introduction to Statistical Learning* (*equation 4.24, page 146),* Equation 6 above refers to the Bayes Classifier assigning observation X=x to class where $\delta_k(x)$ is largest. LDA assigns new observation to X=x by classifying to class where $\hat{\delta}k(x)$ is largest. Generally, LDA has more bias and less variance than similar quadratic discriminant analysis (Gareth et al.).

More detailed analysis on this topic can be found in Chapter 4 of *The Elements of Statistical Learning* which is cited in the references.

2.3 Quadratic Discriminant Analysis (QDA)

QDA is similar to LDA, but unlike LDA, QDA assumes that each class has its own covariance matrix. It has less bias but more variance compared to LDA.

$$\delta_k(x) = -\frac{1}{2}\, x^T\Sigma_k^{-1}x + x^T\Sigma_k^{-1}\mu k - \frac{1}{2} - \frac{1}{2}\mu_k^T\Sigma^{-1}\mu_k - \frac{1}{2}log|\Sigma_k| + log\ \pi_k \quad (Eq\ 7)$$

The above equation is used to assign an observation X=x to the class for which it is the largest. More information about this process is available in the textbook, *Introduction to Statistical Learning.*

2.4 Naïve Bayes

Compared to LDA and QDA, assumptions are more relaxed. Within each class k, it is assumed that the predictors are independent (Eq 8).

For k=1,…,K

$$f_k(x) = f_{k1}(x_1)\ \times\ f_{k2}(x_2) \times\ldots\times f_{kp}(x_p) \qquad (Eq\ 8)$$

Probability is obtained using the below equation.

$$Pr(Y = k|X = x) = \frac{\pi_k \times f_{k1}(x_1)\ \times\ f_{k2}(x_2) \times\ldots\times f_{kp}(x_p)}{\sum_{i=1}^{K}\pi i\ \times f_{i1}(x_1)\ \times\ f_{i2}(x_2) \times\ldots\times f_{ip}(x_p)} \quad (Eq\ 9)$$

For k=1,…,K

The textbook mentions that the naïve Bayes could be a good model to use as it has the ability to balance the bias – variance trade off relatively well (Gareth et al.). Similar, the textbook *Introduction to Statistical Learning* cited in the references contains further information about this process.

2.5 K-Nearest Neighbor

K-Nearest Neighbor is another method and it does not use the Bayes Classifier instead using equation 10 to estimate conditional probability.

$$P(Y = j|X = x_0) = \frac{1}{K}\sum_{i\in \mathrm{N0}} I(yi = j) \qquad (Eq\ 10)$$

Note: Further information on all of the above models may be obtained from both of the sources cited in the references.

2.6 Assessing Model Performance

To test model performance, I considered the overall model accuracy, precision by class, and recall by class. Below are the equations to calculate key model performance metrics.

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \qquad Precision = \frac{True\ Positives}{True\ Positives\ +\ False\ Positives}$$

$$Accuracy = \frac{True\ Positives + True\ Negatives}{True\ Positives + False\ Positives + False\ Nagatives + True\ Negatives}$$

3. Results

3.1 Exploratory Data Analysis

The *Phone.csv* dataset contains 20 explanatory variables and a categorial response variable with four classes (*0,1,2,3).* Each class contains 500 observations. Out of the 20 explanatory variables, six are binary (*blue, dual_sim, four_g, three_g, touch_screen, and wifi*) and the remaining 14 are quantitative. The full dataset contains 2000 observations, and there are no missing values. Table 1 displays the five number summaries of the quantitative variables.

Table 1: *Five Number Summary of Quantitative Variables*

	Min	**Q1**	**Median**	**Mean**	**Q3**	**Max**
battery_power(mAh)	501.0	851.8	1226.0	1238.5	1615.2	1998.0
clock_speed	0.500	0.700	1.500	1.522	2.200	3.000
fc(MP)	0	1.00	3.00	4.31	7.00	19.00
int_memory (GB)	2.00	16.00	32.00	32.05	48.00	64.00
mobile_wt (g)	80.0	109.0	141.0	140.2	170.0	200.0
m_dep (cm)	0.1000	0.2000	0.5000	0.5018	0.8000	1.0000
n_cores (count)	1.00	3.00	4.00	4.52	7.00	8.00
pc (MP)	0	5.000	10.000	9.916	15.000	20.000
sc_h (cm)	5.00	9.00	12.00	12.31	16.00	19.00
sc_w (cm)	0.000	2.000	5.000	5.767	9.00	18.000
talk_time (time)	2.00	6.00	11.00	11.01	16.00	20.00
px_width	500.0	874.8	1247.0	1251.5	1633.0	1998..0
px_height	0.0	282.8	564.0	645.1	947.2	1960.0
ram	256	1209	2146	2124	3064	3998

Figure 1 displays the distribution of selected quantitative variables that appeared more relevant. The boxplot on the left of the figure, *Battey Power by Price,* shows that there is a difference in battery power based on the price range. As you may notice, the higher price range tends to have

devices with more battery power. The same is true for the boxplot in the far left of the figure, *RAM by Price,* in which the RAM varies significantly based on the price range, with those in the higher price range having a much higher RAM configuration. Both cases indicate that the variables will be influential in the final model. Finally, the boxplot in the middle, *Weight by Price,* indicates that as price range increases, device weight decreases. This difference is also important to note as device weight may be an important variable in the final classification model.

Figure 1: *Boxplots of Selected Quantitative Variables*

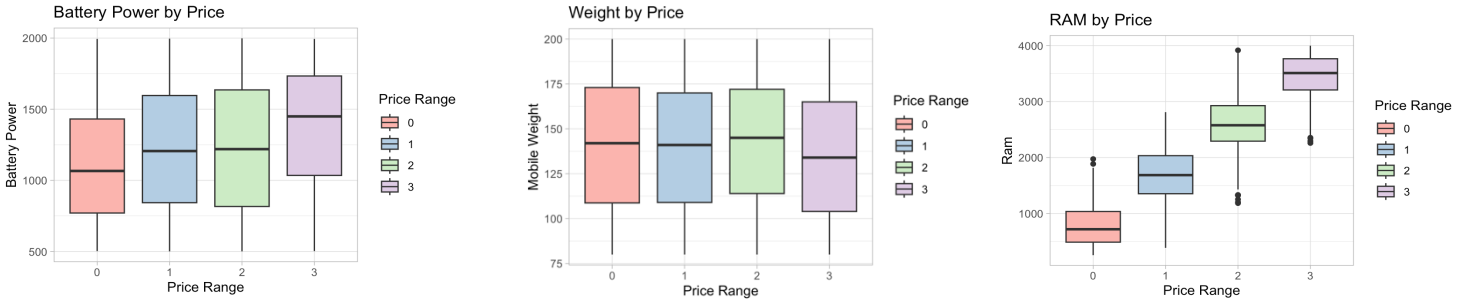
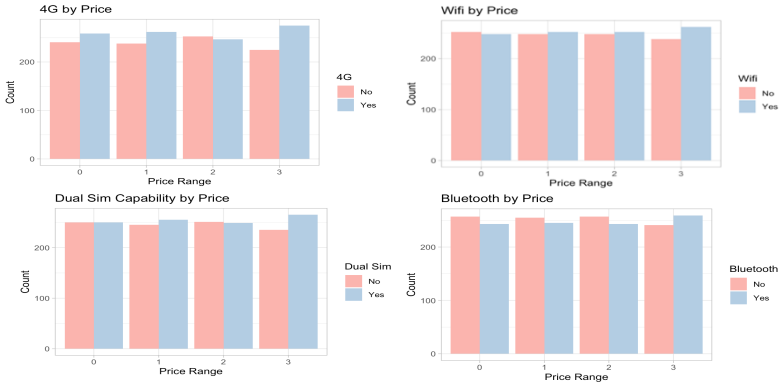


Figure 2 below displays bar charts representing the distribution of selected categorical explanatory variables from the dataset (*four_g, wifi, dual_sim, and blue)*. There do not appear to be significant differences in the characteristics across the four price ranges for any of the selected qualitative variables. Similarity, the graphs for the two qualitative variables not included below (*three_g and touch_screen)* did not appear to show significant differences in quantity across price ranges. The binary variables were mostly balanced as well except for *three_g* (0=477, 1=1523).

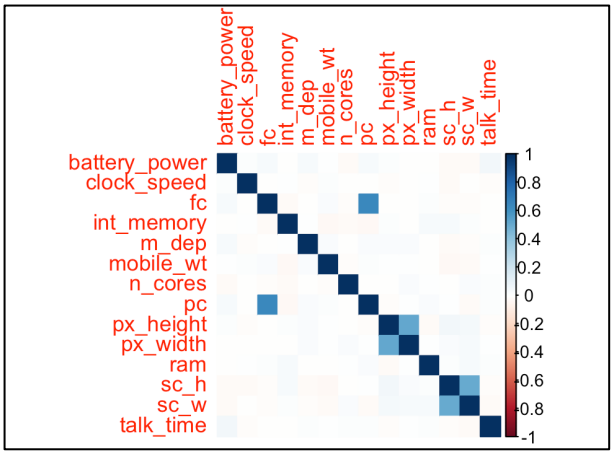
Figure 2: *Distribution of Selected Binary Explanatory Variables*



In addition, I checked for correlations between quantitative explanatory variables. Figure 3 below is the correlation plot. The largest correlation that exists is between the variables

px_height and *px_width* (0.51). There do not appear to be any significant correlations between variables, so this was not a significant concern when formulating the final model.

Figure 3: *Correlation Plot of Quantitative Variables by Color*



Data was also split into train (80%) and test (20%) before modeling.

3.2 Final Model (Multinomial Logistic Regression)

The baseline K assigned to response variable class 0. The model below includes all of the explanatory variables. Figures 4- 6 below display the full model as it is based on baseline. The AIC obtained is 181.5579.

$$ln\left(\frac{P(Y=1|X=x)}{Pr(Y=0|X=x)}\right) = -553.2385 + 0.1416(\text{battery\_power}) - 0.4718(\text{blue}) - 3.2928(\text{clock\_speed}) - 3.3597(\text{dual\_sim}) - 0.3030(\text{fc}) - 0.2389(\text{four\_g}) + 0.1075(\text{int\_memory}) + 1.6638(\text{m\_dep}) - 0.2754(\text{mobile\_wt}) + 1.3812(\text{n\_cores}) + 0.0843(\text{pc}) + 0.0828(\text{px\_height}) + 0.0818(\text{px\_width}) + 0.2302(\text{ram}) - 0.2020(\text{sc\_h}) - 0.4599(\text{sc\_w}) - 0.3545(\text{talk\_time}) - 1.1979(\text{three\_g}) - 0.6346(\text{touch\_screen}) - 4.3355(\text{wifi})$$ (Figure 4- *Class 1*)

$$ln\left(\frac{P(Y=2|X=x)}{Pr(Y=0|X=x)}\right) = -1275.7030 + 0.2714(\text{battery\_power}) + 1.9028(\text{blue}) - 4.5258(\text{clock\_speed}) - 5.1133(\text{dual\_sim}) - 0.3511(\text{fc}) - 2.6913(\text{four\_g}) + 0.3236(\text{int\_memory}) - 6.3224(\text{m\_dep}) - 0.4841(\text{mobile\_wt}) + 2.0261(\text{n\_cores}) + 0.5792(\text{pc}) + 0.1538(\text{px\_height}) + 0.1612(\text{px\_width}) + 0.4355(\text{ram}) - 0.2382(\text{sc\_h}) - 0.1562(\text{sc\_w}) - 0.0545(\text{talk\_time}) - 1.0417(\text{three\_g}) - 0.4989(\text{touch\_screen}) - 9.2900(\text{wifi})$$ (Figure 5-*Class 2)*

$$ln\left(\frac{P(Y=3|X=x)}{Pr(Y=0|X=x)}\right) = 2289.2596 + 0.4098(\text{battery\_power}) - 1.6622(\text{blue}) - 3.6000(\text{clock\_speed}) - 6.5914(\text{dual\_sim}) - 0.5926(\text{fc}) - 5.5873(\text{four\_g}) + 0.5586(\text{int\_memory}) + 0.8319(\text{m\_dep}) - 0.8546(\text{mobile\_wt}) + 2.2206(\text{n\_cores}) + 1.4442(\text{pc}) + 0.2411(\text{px\_height}) + 0.2439(\text{px\_width}) + 0.6694(\text{ram}) + 0.1229(\text{sc\_h}) + 0.4703(\text{sc\_w}) - 0.0412(\text{talk\_time}) - 3.3024(\text{three\_g}) - 2.2868(\text{touch\_screen}) - 12.4343(\text{wifi})$$ (Figure 6-Class 3)

3.2.1 Model Performance

A confusion matrix was created to access model performance (Table 2). Relevant statistics were obtained from the confusion matrix and are displayed in Table 3 below.

Table 2: *Confusion Matrix of Multinomial Logistic Regression Model*

		True Values			
		0	**1**	**2**	**3**
Predicted Values	**0**	101	0	0	0
	1	0	105	2	0
	2	0	0	107	5
	3	0	0	2	78

Table 3: *Recall, Precision, and Overall Accuracy for MLR Model*

	Class 0	*Class 1*	*Class 2*	*Class 3*
Recall	1.000000	1.000000	0.963964	0.939759
Precision	1.0000000	0.9813084	0.9553571	0.9750000
Overall Accuracy = 0.9775				

The overall accuracy of 0.9775 is quite high, this suggests that the model correctly predicts the class 97.75% of the time. Recall and precision were calculated by class. Class 0 and Class 1 have a recall of 1 which means that that the model identified true positives 100% of the time. Recall for Class 2 is slightly lower (0.964) and recall for Class 3 is the lowest (0.940). This means that the model is not as good at identifying true positives in those classes compared to the other classes, but the model still does fairly well. Precision in Class 0 is 1 which indicates that out of the positives, 100% are true positives. Precision in the other classes (1-3) is still relatively high with the lowest being Class 2 with a precision of 0.9554. Overall, these metrics suggest that the model performs well.

3.3 Other Models

While the multinomial logistic regression model was chosen for the final model, I did test different classification models discussed throughout the course and the results from those investigations are below. There is a wide range in variability.

3.3.1 Linear Discriminant Analysis

Table 4 and Table 5 below display the performance measures for LDA. The model obtained included all 20 explanatory variables. The overall accuracy 0.9625 is lower than that of multinomial logistic regression and the recall and precision across the classes are lower as well. This suggests that MLR performs better in this case.

Table 4: *Confusion Matrix LDA*

		True Values			
		0	**1**	**2**	**3**
Predicted Values	**0**	99	1	0	0
	1	2	103	6	0
	2	0	1	104	4
	3	0	0	1	79

Table 5: *Recall, Precision, Accuracy LDA*

	Class 0	*Class 1*	*Class 2*	*Class 3*
Recall	0.9801980	0.9809524	0.9541284	0.9875000
Precision	0.9900000	0.9279279	0.9553571	0.9750000
Overall Accuracy = 0.9625				

3.3.2 Quadratic Discriminant Analysis

Table 6 and Table 7 below display the performance measures for QDA. The overall accuracy is 0.94 which is lower than both MLR and LDA. In addition, recall and precision across the classes are lower than that of MLR or LDA. In this situation, QDA does not perform as well.

Table 6: *Confusion Matrix LDA*

		True Values			
		0	**1**	**2**	**3**
Predicted Values	**0**	98	3	0	0
	1	3	100	5	0
	2	0	2	100	5
	3	0	0	6	78

Table 7: *Recall, Precision, Accuracy LDA*

	Class 0	*Class 1*	*Class 2*	*Class 3*
Recall	0.9702970	0.9523810	0.9009009	0.9397590
Precision	0.9702970	0.9259259	0.9345794	0.9285714
Overall Accuracy = 0.94				

3.3.3 Naïve Bayes

Table 8 and Table 9 below display the performance measures for the naïve Bayes model. The overall accuracy is 0.855 which is lower than LDA, QDA and MLR. The recall and precision for all of the classes is also much lower compared to the previous models. In this case naïve Bayes is not the most appropriate.

Table 8: *Confusion Matrix Naïve Bayes*

		True Values			
		0	**1**	**2**	**3**
Predicted Values	**0**	95	11	0	0
	1	6	81	18	0
	2	0	13	88	5
	3	0	0	5	78

Table 9: *Recall, Precision, Accuracy Naïve Bayes*

	Class 0	*Class 1*	*Class 2*	*Class 3*
Recall	0.9405941	0.7714286	0.7927928	0.9397590
Precision	0.8962264	0.7714286	0.8301887	0.9397590
Overall Accuracy = 0.855				

3.3.4 K-Nearest Neighbor

Table 10 and Table 11 display the KNN model performance. A variety of values for K were tested but the KNN model performed the worse than any of the other models tested. As shown below, the overall accuracy is only 0.5125 and the precision and recall for each class are very low compared to the other models tested.

Table 10: *Confusion Matrix KNN.*

		True Values			
		0	**1**	**2**	**3**
Predicted Values	**0**	65	32	7	0
	1	28	38	21	4
	2	8	25	48	25
	3	0	10	35	54

Table 11: *Recall, Precision, Accuracy Naïve Bayes*

	Class 0	*Class 1*	*Class 2*	*Class 3*
Recall	0.6435644	0.3619048	0.4324324	0.6506024
Precision	0.6250000	0.4175824	0.4528302	0.5454545
Overall Accuracy = 0.5125				

4. Discussion

Ultimately, I used multinomial logistic regression to predict the response class. I chose this model because it had the best performance based on the model accuracy, precision, and recall compared to the other models that I tried on the test dataset. Classification errors can be costly when considering data relating to price ranges, so it is important that the model had a high accuracy, precision, and recall. The model also contained all twenty explanatory variables.

A couple variables stuck out to me as being more influential and important than others in predicting price. I found that RAM was one particularity important variable, as well as mobile phone weight. Companies might consider placing a higher emphasis on these variables when creating the price range for mobile phones. Though, other variables like pixel height and pixel width were also important. There are multiple things that could go wrong in model building and it important to consider these risks when preparing a model. One issue could be correlation of explanatory variables. Problems may also arise when this a large separation between classes in which case LDA or QDA may be more appropriate. This particular dataset did not yield any major problems and constructing the models was relatively straightforward.

In the future, cross validation could be used. Using cross validation could help to improve overall model performance by increase stability and maintaining a large training data size. In addition, it may also be helpful to perform some kind of feature selection method so that explanatory variables will be systematically chosen. Chapter 3, Section 3 in *The Elements of Statistical Learning* provide background on this topic which is relevant for future reference (Haiste et al.). Implementing these things in the future could be helpful to improve model performance.

In sum, the goal was to formulate a classification model to predict the class of a response variable with multiple classes. Investigation revelated that a multinomial logistic regression model was the most appropriate as it had the highest accuracy, precision, and recall compared to the other models that were tested.

5. References

T. Hastie, R. Tibshirani, and J. Friedman. (2009) *The Elements of Statistical Learning: Data Mining, Inference, and Prediction (ESL), 2nd edition, Springer*. https://hastie.su.domains/ElemStatLearn/

G. James, D. Witten, T. Hastie, and R. Tibshirani. (2021) *Intro to statistical learning with applications in R (ISLR), 2nd edition, Springer*. https://www.statlearning.com/

Statista. (2022). *Number of smartphone users in the U.S. 2010-2022 | Statista*. Statista. https://www.statista.com/statistics/201182/forecast-of-smartphone-users-in-the-us/

6. Appendix

```
#Load libraries and download the data
library(MASS) #For LDA and QDA
library(caret) #Confusion matrix and accuracy calculation
library(class) #KNN
library(e1071) #Naive Bayes
library(pROC)
library(nnet)
library(caret)
library(dplyr)
library(ggplot2)
library(corrplot)

set.seed(1)

phone <- read.csv("./Phone.csv")
train_data <- phone %>%sample_frac(0.8)
test_data <- anti_join(phone, train_data)

EDA
#change to qualatative variables to factor
attach(phone)
phone$blue <- as.factor(blue)
phone$dual_sim <- as.factor(dual_sim)
phone$four_g <- as.factor(four_g)
phone$three_g <- as.factor(three_g)
phone$touch_screen <- as.factor(touch_screen)
phone$wifi <- as.factor(wifi)
phone$price_range <- as.factor(price_range)

#Explore the data
summary(phone)
cor(phone)

#QUAN
cor_matrix <- cor(phone %>% select_if(is.numeric))
corrplot(cor_matrix, method="color")
#correlations between Pc and fc, Px width and height, Sh_h and sh_w

#battery power- shows some differences indicates it could be useful
ggplot(phone,aes(x = price_range, y =
                   battery_power,
                 fill=price_range)) + geom_boxplot() + scale_fill_brewer(palette =
                                                                           "Pastel1")+
  labs(title= "Battery Power by Price",x="Price Range", y="Battery Power",
```

```
      fill= "Price Range")+
  theme_light()
#px_height
ggplot(phone,aes(x = price_range, y = px_height,
                 fill=price_range)) +
  geom_boxplot() + scale_fill_brewer(palette ="Pastel1")+
  labs(title="Speed", x="Price Range", y="Battery Power")+theme_light()

#px wid
ggplot(phone,aes(x = price_range, y = px_width,
                 fill=price_range)) +
  geom_boxplot() + scale_fill_brewer(palette ="Pastel1")+
  labs(title="Pixel Width by Price", x="Price Range", y="Pixel Width", fill="Price Range")+
  theme_light()
#ram
ggplot(phone,aes(x = price_range, y = ram,
                 fill=price_range)) +
  geom_boxplot() + scale_fill_brewer(palette ="Pastel1")+
  labs(title="RAM by Price", x="Price Range", y="Ram", fill="Price Range")+
  theme_light()

#mobile weight - cheaper seems to be heavier
ggplot(phone,aes(x = price_range, y = mobile_wt,
                 fill=price_range)) +
  geom_boxplot() + scale_fill_brewer(palette ="Pastel1")+
  labs(title="Weight by Price", x="Price Range", y="Mobile Weight", fill="Price Range")+
  theme_light()

#internal memory
ggplot(phone,aes(x = price_range, y = int_memory,
                 fill=price_range)) +
  geom_boxplot() + scale_fill_brewer(palette ="Pastel1")+
  labs(title="Internal Memory by Price", x="Price Range", y="Internal Memory", fill="Price Range")+
  theme_light()


#categorical variables
#
#BLUEtooth maybe
ggplot(phone,aes(x = price_range, fill = blue)) +
  geom_bar(position = "dodge") +
  labs(title = "Bluetooth by Price", x = "Price Range", y = "Count",
       fill ="Bluetooth") + scale_fill_brewer(palette = "Pastel1",
                                              labels=c("No","Yes"))+theme_light()

#dual sim-
```

```
ggplot(phone,aes(x = price_range, fill = dual_sim)) +
  geom_bar(position = "dodge") +
  labs(title = "Dual Sim Capability by Price", x = "Price Range", y = "Count",
       fill ="Dual Sim") + scale_fill_brewer(palette = "Pastel1",
                                     labels=c("No","Yes"))+theme_light()

#4g-
ggplot(phone,aes(x = price_range, fill = four_g)) +
  geom_bar(position = "dodge") +
  labs(title = "4G by Price", x = "Price Range", y = "Count",
       fill ="4G") + scale_fill_brewer(palette = "Pastel1",
                                     labels=c("No","Yes"))+theme_light()

#wifi-
ggplot(phone,aes(x = price_range, fill = wifi)) +
  geom_bar(position = "dodge") +
  labs(title = "Wifi by Price", x = "Price Range", y = "Count",
       fill ="Wifi") + scale_fill_brewer(palette = "Pastel1",
                                       labels=c("No","Yes"))+theme_light()

MLR

mlr <- multinom(price_range~., data=train_data,
                family=multinomial)
summary(mlr)
mlr_pred <- predict(mlr, newdata=test_data, type="class")


cm_mlr <- table(predicted_values = mlr_pred, true_values=test_data$price_range)
cm_mlr <- confusionMatrix(cm_mlr)
cm_mlr
cm_mlr
precision <- cm_mlr$byClass[, "Precision"]
recall<- cm_mlr$byClass[,"Recall"]
precision
recall

LDA

LDA <- lda(price_range~., data=train_data)
LDApred <- predict(LDA, newdata=test_data)
LDA_class <- LDApred$class

cm_lda <- table(predicted_values = LDA_class, true_values=test_data$price_range)

cm_lda <- confusionMatrix(cm_lda)
```

```
cm_lda
cm_lda$predicted_values <- as.numeric(cm_lda$predicted_values)

precision <- cm_lda$byClass[, "Precision"]
recall<- cm_lda$byClass[,"Recall"]
precision
recall

QDA

QDA <- qda(price_range~., data=train_data)
QDApred <- predict(QDA, newdata=test_data)
QDA_class <- QDApred$class
cm_qda <- table(predicted_values = QDA_class, true_values=test_data$price_range)
cm_qda <- confusionMatrix(cm_qda)

cm_qda$predicted_values <- as.numeric(cm_qda$predicted_values)

precision <- cm_qda$byClass[, "Precision"]
recall<- cm_qda$byClass[,"Recall"]
precision

recall

KNN

train_labels <- train_data$price_range
test_labels <- test_data$price_range

k <- 7

knn_pred <- knn(train=scale(train_data[,-21]), test=scale(test_data[,-21]),
            cl=train_labels, k=k)

cm_knn <- table(predicted_values = knn_pred, true_values=test_labels)

cm_knn <- confusionMatrix(cm_knn)


cm_knn$predicted_values <- as.numeric(cm_knn$predicted_values)
cm_knn
precision <- cm_knn$byClass[, "Precision"]
recall<- cm_knn$byClass[,"Recall"]
```

```
precision
recall
```

NAÏVE BAYES

```
NB <- naiveBayes(price_range~., data=train_data)

NB


NB_class <- predict(NB, newdata=test_data)
cm_NB <- table(predicted_values=NB_class, true_values=test_data$price_range)
cm_NB <- confusionMatrix(cm_NB)

cm_NB$predicted_values <- as.numeric(cm_NB$predicted_values)
cm_NB
precision <- cm_NB$byClass[, "Precision"]
recall<- cm_NB$byClass[,"Recall"]
precision
recall
```