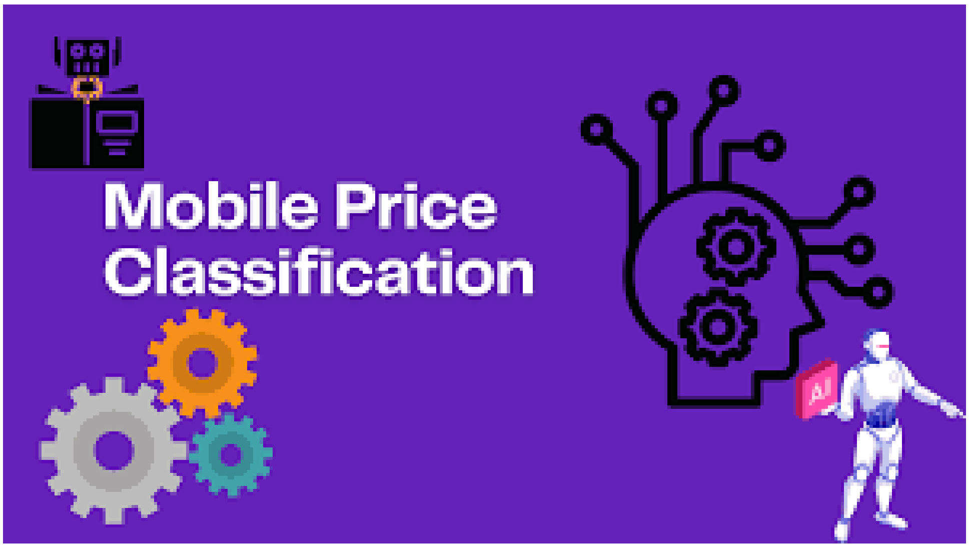
Mobile Price Classification



Foundations Of AI

ISTM 6214

Group 4 Project Report

Komal Patil, Christine Mundiya,

Selam Zegeye, Alejandra Boyd

Table of Contents

Executive Summary

Mobilitas Inc. is a mobile technology startup founded in 2023 by industry veteran Bob. The company's goal is to deliver high-quality yet affordable smartphones to compete with top brands like Apple and Samsung. To accurately price its phones and analyze the competitive mobile phone market, Mobilitas is leveraging detailed sales data on thousands of mobile phone models across leading manufacturers. By applying advanced data mining techniques, Mobilitas aims to determine the relationship between mobile features like RAM, internal memory, etc. and selling price. These insights will allow Mobilitas to optimize pricing and feature sets for its flagship and future smartphone models. While still a young firm, Mobilitas is rapidly developing its capabilities in mobile engineering and data analytics to establish itself as an innovative player disrupting the market from below on price.

Company History

In 2023, longtime tech enthusiast Bob decided to pursue his dream of starting his own mobile phone company. Drawing on over 15 years of experience working in the mobile industry, Bob founded Mobilitas Inc. with the goal of producing high-quality, competitively priced smartphones to challenge industry giants like Apple and Samsung.

Bob started Mobilitas in his garage with a small team of engineers he had worked with previously. Together, they began developing their first phone model which would compete with the latest iPhone and Galaxy models on technical specifications at a lower cost. Their focus was on sourcing top-notch yet affordable components and utilizing an efficient, user-friendly operating system.

After a year of intense research, development and testing, Mobilitas unveiled the Mobilitas One in early 2023 as their flagship model. Priced at 20% below comparable iPhone and Samsung phones, the Mobilitas One boasted features like a rapid A17 processor, crystal clear OLED display, and an advanced triple camera system. It quickly earned praise from tech critics for its combination of power, functionality and value.

Buoyed by the success of their first model, Mobilitas moved out of Bob's garage into a modern office facility downtown. More staff were hired in design, engineering and marketing as the company prepared to scale up and deliver their phones to consumers nationwide.

Today, Mobilitas continues to grow as an emerging fan-favorite within the mobile space. With Bob still at the helm, the company culture remains focused on innovation and delivering maximum usability per dollar. 2024 is set to welcome the Mobilitas Two, which Bob promises will once again "surprise and delight" customers with its fresh take on smartphone technology.

Research Purpose and Motivation

Mobilitas recognizes that pricing its products competitively is critical in the budget-conscious mobile market. However, simply undercutting the pricing of established giants like Apple or Samsung will not be sustainable long-term. Mobilitas must apply data and analytics to optimize the pricing-to-capabilities ratio of its phones. By thoroughly analyzing the market's historical pricing of mobiles with varying feature sets, Mobilitas aims to determine the sweet spot for features and pricing that will appeal to its target demographic. These data-driven insights will power Mobilitas' device roadmap and tech specs for at least the next 2-3 years as the foundation is laid for the company. Establishing razor-sharp focus here is crucial before attempting to grow market share and bootstrap operations.

Competitor Analysis

Mobilitas has targeted Apple and Samsung for disruptive competition based on their overwhelming 60%+ combined market share. Both companies have substantial brand value and loyalty given their reputation for quality, features and status. However, their premium pricing leaves a large segment of price-conscious mobile consumers underserved. Chinese manufacturers like Xiaomi and Oppo have found major success competing as value brands, signaling the market opportunity. Mobilitas will blend premium feel and user experience with behind-the-scenes optimization powered by data. Over time, Mobilitas intends to fully leverage analytics across marketing, supply chain logistics and product roadmapping to complement the lean operational efficiency required of a start-up.

Mobile Price Dataset

Mobilitas Inc gathered competitor data (Mobile Price Classification) comprising 21 variables that capture sales data for mobile phones from various companies. The competitor sales data gathered includes over 2,000 observations detailing 21 attributes of mobile phone models

from various leading manufacturers. After cleansing the raw data, there are 2,000 rows of phones with full details across the 21 columns covering technical specifications, features, dimensions, battery life, and other relevant attributes that connect to pricing.
The target variable that Mobilitas seeks to better understand is the Price Range, contained in the last column. This shows a 0 for low cost phones and 1 for high cost phones, simplified into just two categories.

Data Variables

The definition of the numeric columns in the dataset is as follows:

- **Battery Power** - Total energy a battery can store in one charge measured in milliamp hours (mAh)
- **Blue** - Boolean indicating if bluetooth is included (1 = yes, 0 = no)
- **Clock Speed** - Speed at which the phone's processor executes instructions, measured in GHz
- **Dual Sim** - Boolean indicating if dual SIM card slots are available (1 = yes, 0 = no)
- **Fc** - Front camera megapixel count
- **FourG** - Boolean indicating if 4G cellular connectivity is supported (1 = yes, 0 = no)
- **Int Memory** - Internal storage capacity, measured in gigabytes (GB)
- **M dep** - Phone depth, measured in centimeters (cm)
- **Mobile wt** - Weight of phone, measured in grams (g)
- **N cores** - Number of cores in the phone's processor
- **Pc** - Primary rear camera megapixel count
- **Px Height** - Screen pixel resolution height, typically measured in pixels
- **Px Width** - Screen pixel resolution width, typically measured in pixels
- **Ram - RAM** (random access memory) size, measured in megabytes (MB)
- **Sc h** - Screen height, measured in cm
- **Sc w** - Screen width, measured in cm
- **Talk time** - Battery talk time, measured in hours (longest voice call duration on single charge)
- **ThreeG** - Boolean indicating if 3G cellular connectivity is supported (1 = yes, 0 = no)
- **Touch Screen** - Boolean indicating if a touchscreen is present (1 = yes, 0 = no)

- **Wifi** - Boolean indicating if WiFi connectivity is included (1 = yes, 0 = no)
- **Price Range New** - This is the price range with values of 0 (low cost) and 1 (high cost).

Data Preparation

As the mobile price dataset obtained from competitors was already comprehensive and relatively clean, minimal additional data preparation was required before analysis. The raw data contained complete rows for over 2,000 phone models with technical attributes fully populated across the 21 columns of interest, including detailed specs as well as pricing indicators. As no missing values or anomalies were found that could bias analysis, data preprocessing consisted mainly of subsetting the data to the variables relevant to modeling phone prices. Additionally, data types were encoded for compatibility with machine learning algorithms. Outliers did not need removal given the focus on generalized price prediction rather than individual phone models. With consistently structured data covering a breadth of phones ready for analysis, custom cleaning procedures were not necessitated.

Data Analysis

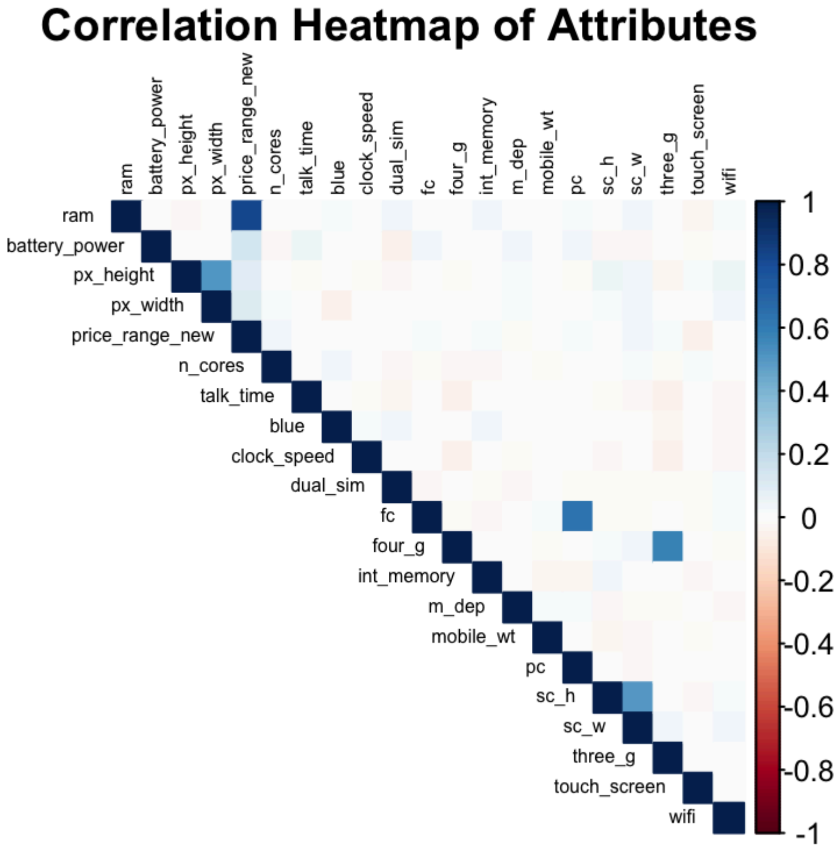


Figure 1.1 Correlation Heatmap of Attributes

The correlation heatmap was created in order to examine the connections between different mobile phone aspects, including call time, RAM, battery life, pixel size, price range, and number of processing cores. The correlation coefficients between these characteristics are shown graphically in the heatmap, where stronger positive correlations are denoted by warmer colors and stronger negative correlations by cooler colors. Interestingly, adding the "price range" attribute to the study reveals relationships with other characteristics that shed light on the variables affecting mobile phone pricing dynamics. One important finding is that RAM and battery power have a positive association, which is consistent with the hypothesis that larger RAM capacities are frequently accompanied by larger battery capacities.

The heatmap additionally sheds light on the connections between various mobile phone hardware and functional features. This visual representation provides a thorough understanding of how these attributes interact and affect mobile device pricing, making it a useful tool for spotting trends and interconnections within the dataset. More investigation and statistical analysis may yield more comprehensive insights into the intricate relationships between the characteristics of mobile phones and how much they cost.

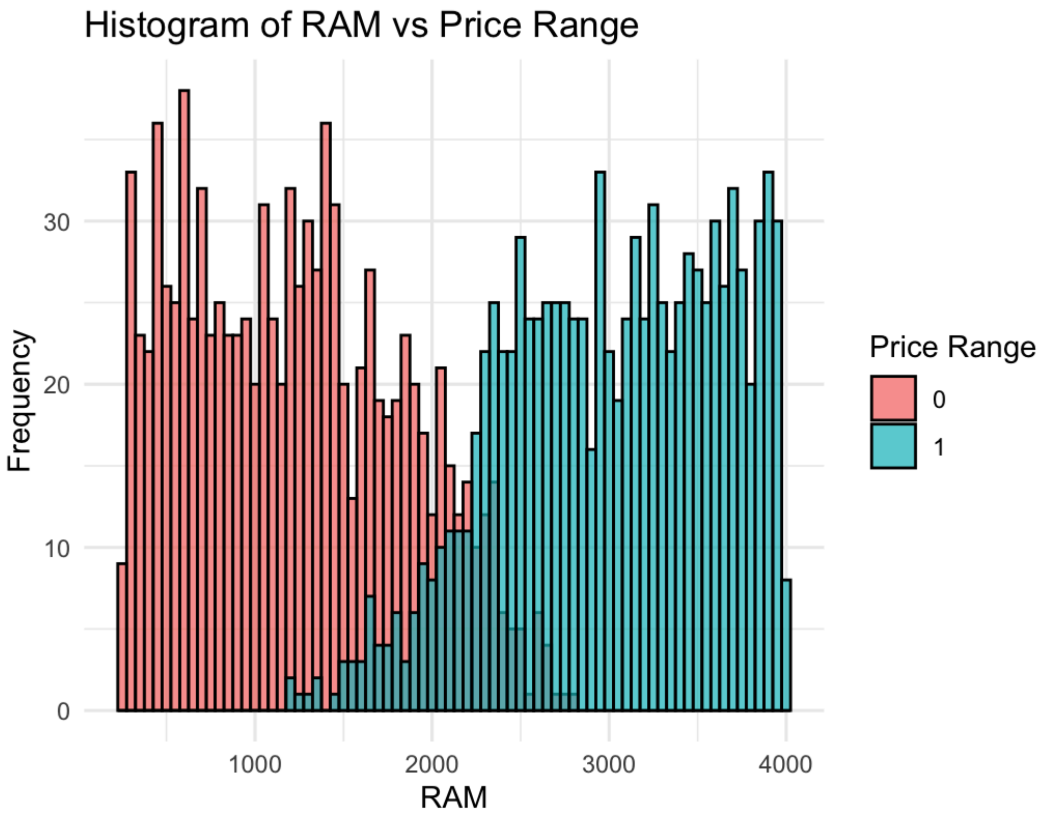


Figure 1.2 Histogram of RAM vs Price Range

This analysis of the dataset reveals interesting insights, particularly when examining the relationship between RAM and the price range of mobile phones. A histogram was constructed to visualize the distribution of RAM across different price ranges. The graph illustrates that higher RAM capacities are more prevalent in phones with elevated price ranges, suggesting a positive correlation between RAM size and device cost. This finding aligns with the general expectation that smartphones with greater RAM capabilities often command higher prices in the market. The histogram provides a clear and concise overview, enabling a quick understanding of how RAM influences the pricing dynamics within the dataset. Further exploration and statistical analyses could delve deeper into the nuanced relationships between other features and price ranges, offering valuable perspectives on market trends and consumer preferences in the realm of mobile phones.

Model and Analysis

Target Variable: Price Range

Decision Tree

Initially, five decision tree models were created to predict the price range of phones based on technical specs. The target variable selected was price range (low vs high) and predictors included battery power, RAM, resolution, and internal memory. The historical pricing data was split 80/20 into training and test data sets. Across the models, RAM was the strongest individual predictor, able to accurately classify a phone's price range with approximately 77% accuracy. Therefore, the model which used RAM as an attribute (Model 2) was chosen as it had the best model accuracy and prediction, and its attribute applied to all 4 mobile classes included in the dataset.

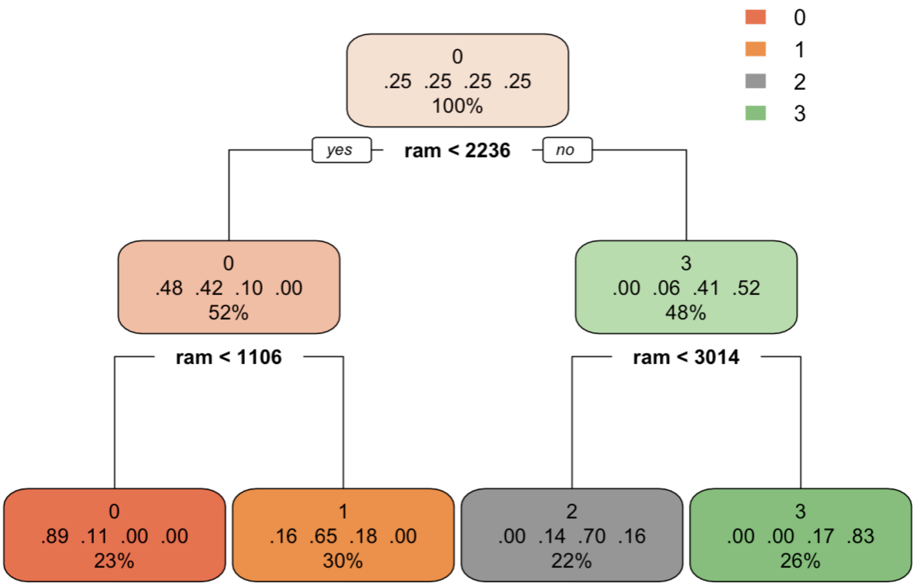
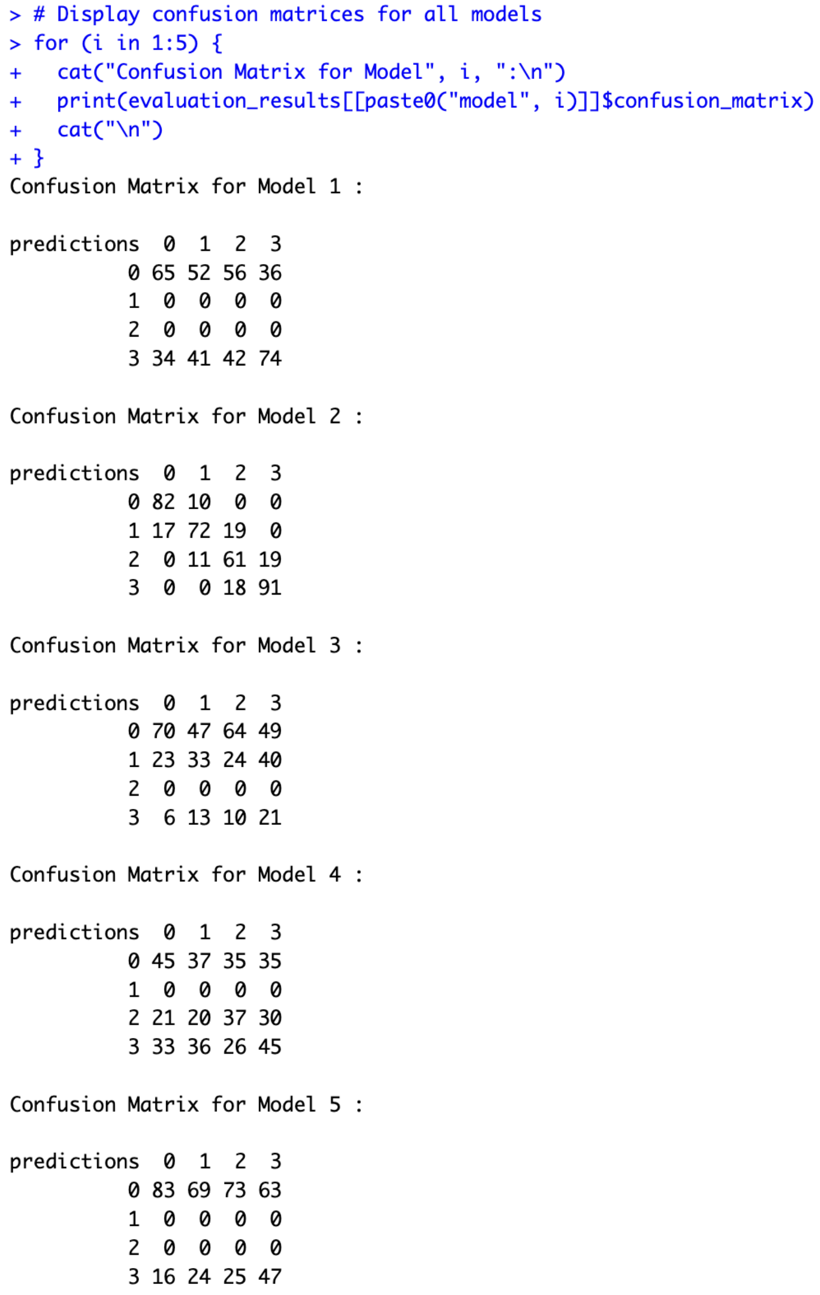


Figure 1.3 Decision Tree for RAM vs Price Range

Confusion Matrix

Model Evaluation was carried out for all five models using confusion matrices, which quantify predictions vs actuals by price class. In a Confusion Matrix, higher diagonals indicate greater class accuracy. The target variable to be estimated was Price Range. Model one used Battery Power as a predictor, Model 2 used RAM, Model 3 used Pixel Height, Model 4 used Internal Memory, and Model 5 used Pixel Width as an attribute to predict the target variable. The following are the confusion matrices for all five decision tree models.



```
> # Display confusion matrices for all models
> for (i in 1:5) {
+   cat("Confusion Matrix for Model", i, ":\n")
+   print(evaluation_results[[paste0("model", i)]]$confusion_matrix)
+   cat("\n")
+ }
Confusion Matrix for Model 1 :

predictions  0  1  2  3
          0 65 52 56 36
          1  0  0  0  0
          2  0  0  0  0
          3 34 41 42 74

Confusion Matrix for Model 2 :

predictions  0  1  2  3
          0 82 10  0  0
          1 17 72 19  0
          2  0 11 61 19
          3  0  0 18 91

Confusion Matrix for Model 3 :

predictions  0  1  2  3
          0 70 47 64 49
          1 23 33 24 40
          2  0  0  0  0
          3  6 13 10 21

Confusion Matrix for Model 4 :

predictions  0  1  2  3
          0 45 37 35 35
          1  0  0  0  0
          2 21 20 37 30
          3 33 36 26 45

Confusion Matrix for Model 5 :

predictions  0  1  2  3
          0 83 69 73 63
          1  0  0  0  0
          2  0  0  0  0
          3 16 24 25 47
```

Interpretation

Model 1:

- Class 0: Correctly predicted (True Positives): 65
- Class 1: Incorrectly predicted as 0 (False Negatives): 52
- Class 2: Incorrectly predicted as 0 (False Negatives): 56

- Class 3: Incorrectly predicted as 0 (False Negatives): 36

Model 2:

- Class 0: Correctly predicted (True Positives): 82
- Class 1: Incorrectly predicted as 0 (False Negatives): 10, and as 2: 0
- Class 2: Incorrectly predicted as 1 (False Negatives): 11, and as 3: 19
- Class 3: Incorrectly predicted as 2 (False Negatives): 18

Model 3:

- Class 0: Correctly predicted (True Positives): 70
- Class 1: Incorrectly predicted as 0 (False Negatives): 47
- Class 2: Incorrectly predicted as 0 (False Negatives): 64
- Class 3: Incorrectly predicted as 0 (False Negatives): 49

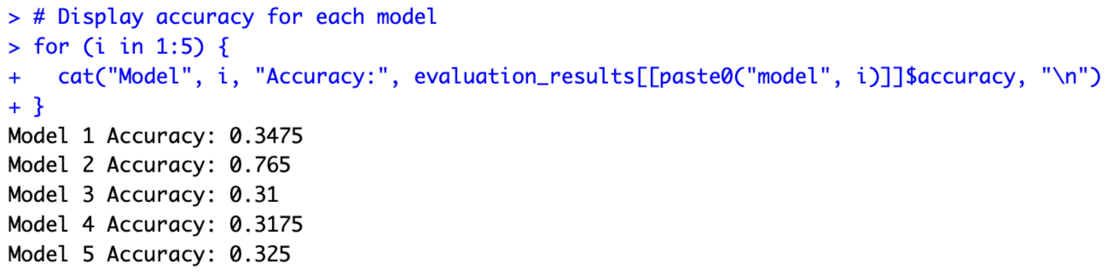
Model 4:

- Class 0: Correctly predicted (True Positives): 45
- Class 1: Incorrectly predicted as 0 (False Negatives): 37
- Class 2: Incorrectly predicted as 0 (False Negatives): 35, and as 3: 30
- Class 3: Incorrectly predicted as 0 (False Negatives): 35, and as 1: 36, and as 2: 26

Model 5:

- Class 0: Correctly predicted (True Positives): 83
- Class 1: Incorrectly predicted as 0 (False Negatives): 69
- Class 2: Incorrectly predicted as 0 (False Negatives): 73
- Class 3: Incorrectly predicted as 0 (False Negatives): 63

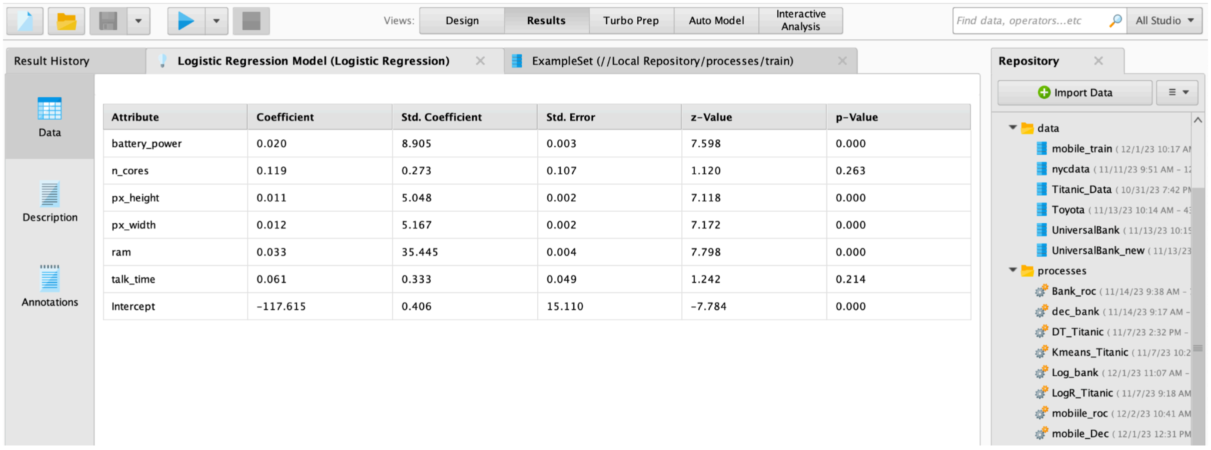
Model Accuracy for Each Model



```
> # Display accuracy for each model
> for (i in 1:5) {
+   cat("Model", i, "Accuracy:", evaluation_results[[paste0("model", i)]]$accuracy, "\n")
+ }
Model 1 Accuracy: 0.3475
Model 2 Accuracy: 0.765
Model 3 Accuracy: 0.31
Model 4 Accuracy: 0.3175
Model 5 Accuracy: 0.325
```

Model 2 performed best and had the best overall model accuracy of **0.765%**. Other models showed weaker diagonals and more cross-class errors. For example, Model 1 incorrectly labeled 52 False Negatives. Model 5 had significant errors labeling mid-tier classes. In summary, Model 2's confusion matrix demonstrated it most accurately making it the optimal model for Mobilitas' pricing needs.

Logistic Regression

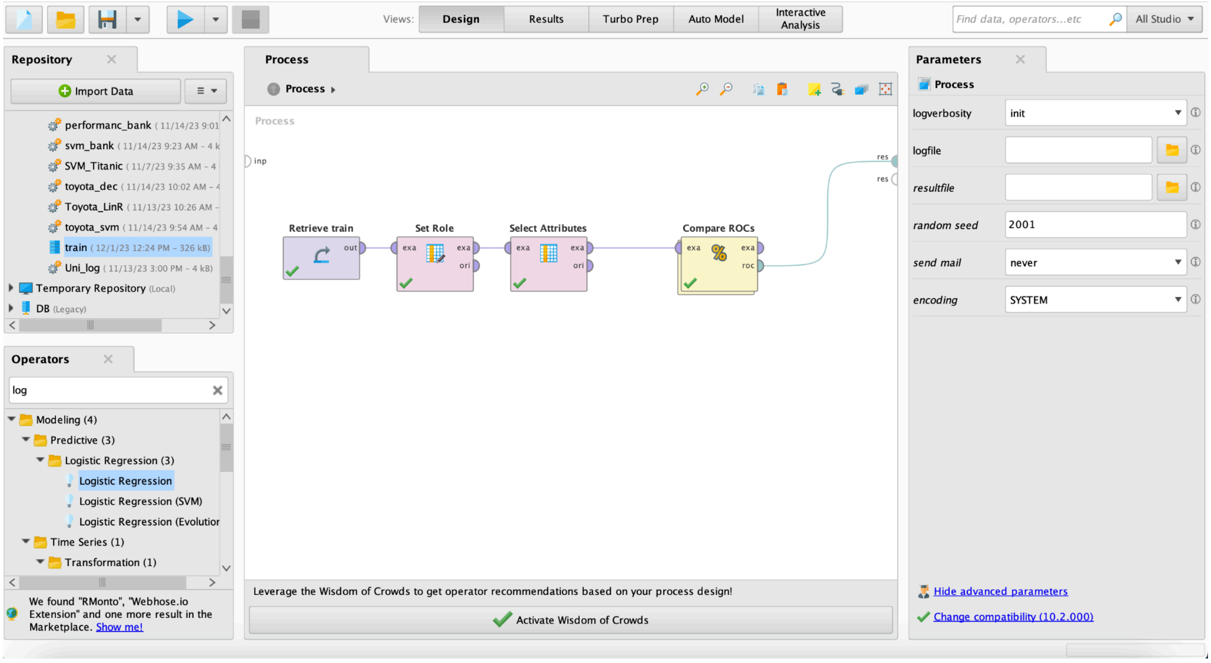

Views:
Design
Results
Turbo Prep
Auto Model
Interactive Analysis
Find data, operators...etc
All Studio
Result History
Logistic Regression Model (Logistic Regression)
ExampleSet (//Local Repository/processes/train)
Data
Description
Annotations
Repository
Import Data
data
mobile_train (12/1/23 10:17 A
nycdata (11/11/23 9:51 AM – 1
Titanic_Data (10/31/23 7:42 P
Toyota (11/13/23 10:14 AM – 4
UniversalBank (11/13/23 10:1
UniversalBank_new (11/13/23
processes
Bank_roc (11/14/23 9:38 AM –
dec_bank (11/14/23 9:17 AM –
DT_Titanic (11/7/23 2:32 PM –
Kmeans_Titanic (11/7/23 10:2
Log_bank (12/1/23 11:07 AM –
LogR_Titanic (11/7/23 9:18 AM
mobiile_roc (12/2/23 10:41 AM
mobile_Dec (12/1/23 12:31 PM

Attribute	Coefficient	Std. Coefficient	Std. Error	z-Value	p-Value
battery_power	0.020	8.905	0.003	7.598	0.000
n_cores	0.119	0.273	0.107	1.120	0.263
px_height	0.011	5.048	0.002	7.118	0.000
px_width	0.012	5.167	0.002	7.172	0.000
ram	0.033	35.445	0.004	7.798	0.000
talk_time	0.061	0.333	0.049	1.242	0.214
Intercept	-117.615	0.406	15.110	-7.784	0.000

Figure 1.4 Logistic Regression

The logistic regression model is utilized to forecast the target variable, price_range, employing independent variables such as battery_power, n_cores, px_width, px_height, ram, and talk_time. The coefficients in the model indicate the impact of each independent variable on the likelihood of the target outcome, with positive coefficients suggesting an increase in this likelihood. By examining the p-values, we can identify the significant attributes, which include battery_power, n_cores, px_width, px_height, and ram. Notably, Ram stands out as the most influential attribute, as evidenced by its highest coefficient of 0.033. Given that the Ram attribute's z-value is 7.798, it is an effective prediction for the mobile price.

ROC


Views:
Design
Results
Turbo Prep
Auto Model
Interactive Analysis
Find data, operators...etc
All Studio
Repository
Import Data
performanc_bank (11/14/23 9:01
svm_bank (11/14/23 9:23 AM – 4 k
SVM_Titanic (11/7/23 9:35 AM – 4
toyota_dec (11/14/23 10:02 AM – 4
Toyota_LinR (11/13/23 10:26 AM –
toyota_svm (11/14/23 9:54 AM – 4
train (12/1/23 12:24 PM – 326 kB)
Uni_log (11/13/23 3:00 PM – 4 kB)
Temporary Repository (Local)
DB (Legacy)
Operators
log
Modeling (4)
Predictive (3)
Logistic Regression (3)
Logistic Regression
Logistic Regression (SVM)
Logistic Regression (Evolutior
Time Series (1)
Transformation (1)
Process
Process
inp
res
res
Retrieve train
out
Set Role
exa
exa
ori
Select Attributes
exa
exa
ori
Compare ROCs
exa
exa
roc
Parameters
Process
Hide advanced parameters
Change compatibility (10.2.000)
Activate Wisdom of Crowds

logverbosity: init

logfile:

resultfile:

random seed: 2001

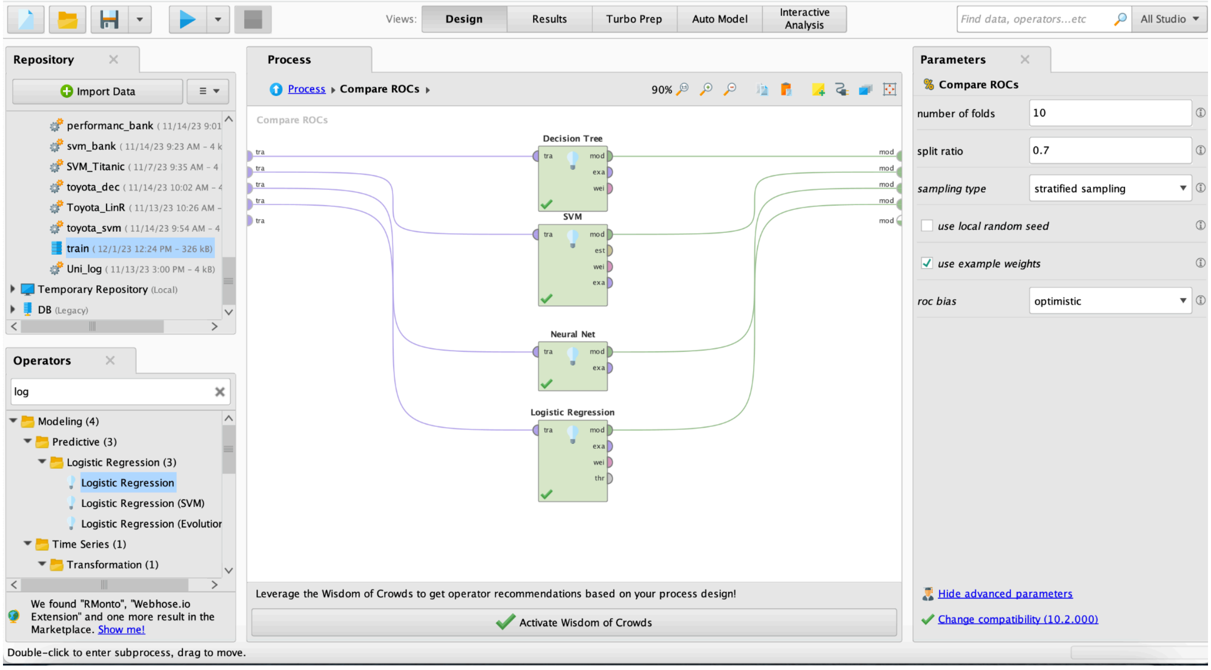
send mail: never

encoding: SYSTEM

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

We found "RMonto", "Webhose.io Extension" and one more result in the Marketplace. Show me!

Figure 1.5 ROC


Views:
Design
Results
Turbo Prep
Auto Model
Interactive Analysis
Find data, operators...etc
All Studio
Repository
Import Data
performanc_bank (11/14/23 9:01
svm_bank (11/14/23 9:23 AM – 4 k
SVM_Titanic (11/7/23 9:35 AM – 4
toyota_dec (11/14/23 10:02 AM – 4
Toyota_LinR (11/13/23 10:26 AM –
toyota_svm (11/14/23 9:54 AM – 4
train (12/1/23 12:24 PM – 326 kB)
Uni_log (11/13/23 3:00 PM – 4 kB)
Temporary Repository (Local)
DB (Legacy)
Operators
log
Modeling (4)
Predictive (3)
Logistic Regression (3)
Logistic Regression
Logistic Regression (SVM)
Logistic Regression (Evolutior
Time Series (1)
Transformation (1)
Process
Process
Compare ROCs
90%
Compare ROCs
tra
tra
tra
tra
tra
Decision Tree
tra
mod
exa
wei
SVM
tra
mod
est
wei
exa
Neural Net
tra
mod
exa
Logistic Regression
tra
mod
exa
wei
thr
mod
mod
mod
mod
mod
Parameters
Compare ROCs
Hide advanced parameters
Change compatibility (10.2.000)
Activate Wisdom of Crowds

number of folds: 10

split ratio: 0.7

sampling type: stratified sampling

□ use local random seed

☑ use example weights

roc bias: optimistic

Leverage the Wisdom of Crowds to get operator recommendations based on your process design!

We found "RMonto", "Webhose.io Extension" and one more result in the Marketplace. Show me!

Double-click to enter subprocess, drag to move.

Figure 1.6 Comparing ROCs (Decision Tree, SVM, Neural Net and Logistic Regression)

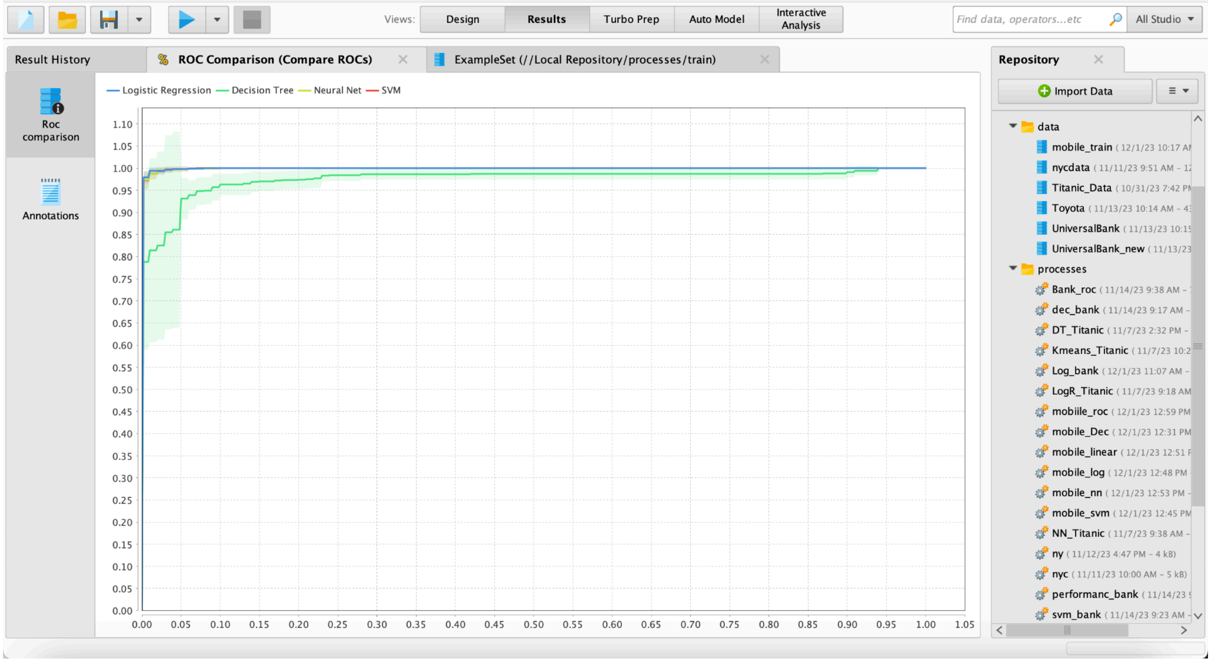
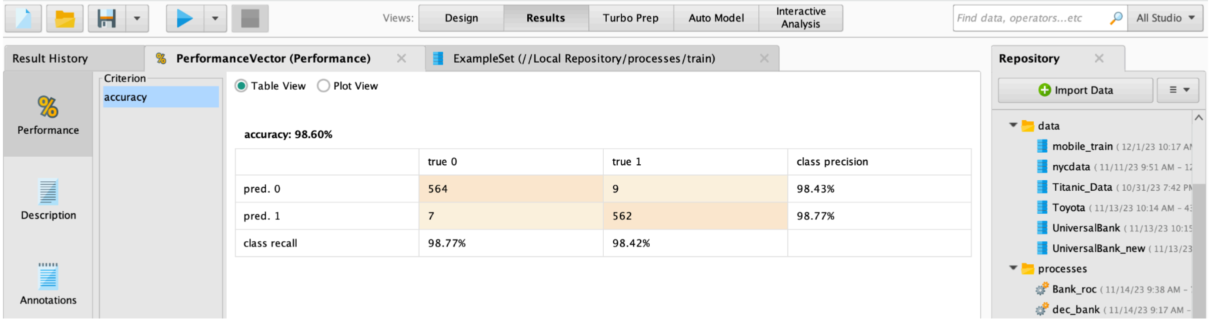


Figure 1.7 ROC curve

Figure 1.7 features an ROC curve generated through RapidMiner, illustrating a comparative analysis of the decision tree model, logistic regression model, neural network model, and SVM model. The ROC curve highlights that the logistic regression model outperforms the others, exhibiting superior accuracy in both true positives and false positives. In contrast, the decision tree model performs the least effectively, evident in its smaller AUC.



accuracy: 98.60%

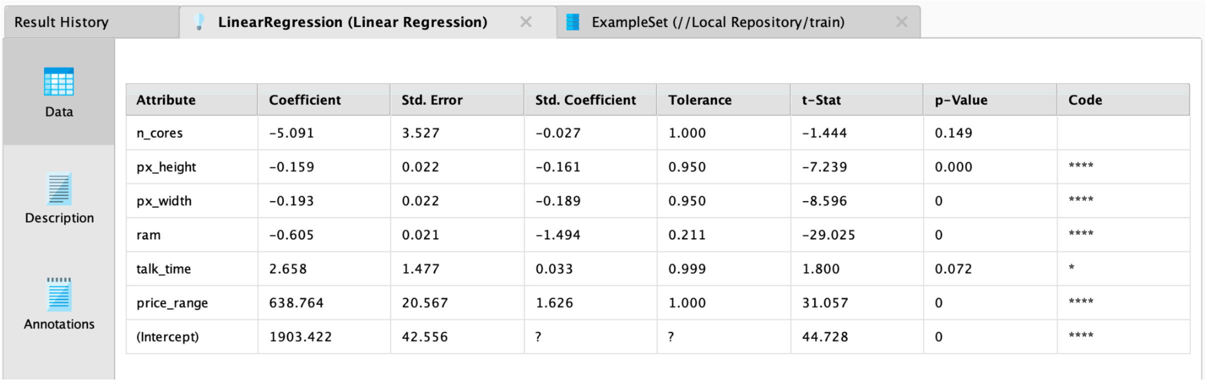
	true 0	true 1	class precision
pred. 0	564	9	98.43%
pred. 1	7	562	98.77%
class recall	98.77%	98.42%	

Figure 1.8 Confusion Matrix for Logistics

Logistic regression has the largest AUC, indicating that their accuracy is high at 98.60% and the class recall is 98.77%.

Target Variable: Battery Power

Linear Regression

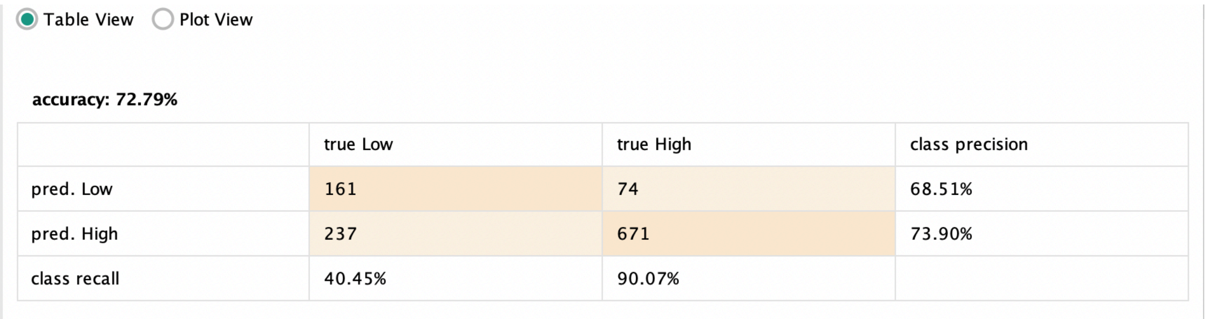

Result History
LinearRegression (Linear Regression)
ExampleSet (//Local Repository/train)
Data
Description
Annotations

Attribute	Coefficient	Std. Error	Std. Coefficient	Tolerance	t-Stat	p-Value	Code
n_cores	-5.091	3.527	-0.027	1.000	-1.444	0.149	
px_height	-0.159	0.022	-0.161	0.950	-7.239	0.000	****
px_width	-0.193	0.022	-0.189	0.950	-8.596	0	****
ram	-0.605	0.021	-1.494	0.211	-29.025	0	****
talk_time	2.658	1.477	0.033	0.999	1.800	0.072	*
price_range	638.764	20.567	1.626	1.000	31.057	0	****
(Intercept)	1903.422	42.556	?	?	44.728	0	****

Figure 1.9 Linear Regression

The linear regression analysis reveals that the coefficient for the price range is statistically significant with a p-value less than 0.05, indicating its influence on the target variable, battery power. However, the coefficients for the number of cores, pixel height, pixel width, and talk time are not statistically significant, as their respective p-values exceed the conventional threshold of 0.05. The intercept also demonstrates statistical significance. Specifically, the coefficient for the price range implies a certain change in battery power for a unit change in the price range. These findings highlight the importance of the price range as a significant predictor of battery power in the context of this regression model, while the other examined variables do not exhibit statistical significance.

Confusion Matrix for LinearRegression

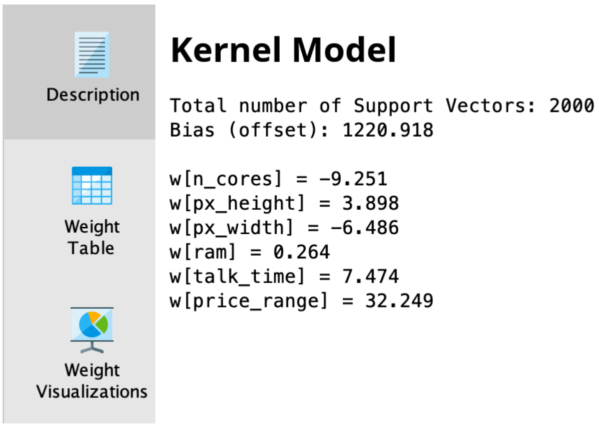


accuracy: 72.79%

	true Low	true High	class precision
pred. Low	161	74	68.51%
pred. High	237	671	73.90%
class recall	40.45%	90.07%	

The confusion matrix reveals a model accuracy of 72.79%, showcasing better performance in predicting true high-class instances (90.07% recall) compared to true low-class instances (40.45% recall).

Support vector Machine



Kernel Model

```
Total number of Support Vectors: 2000
Bias (offset): 1220.918

w[n_cores] = -9.251
w[px_height] = 3.898
w[px_width] = -6.486
w[ram] = 0.264
w[talk_time] = 7.474
w[price_range] = 32.249
```

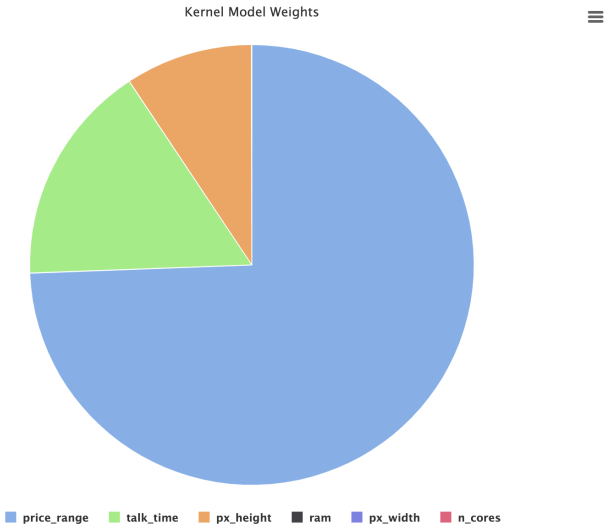
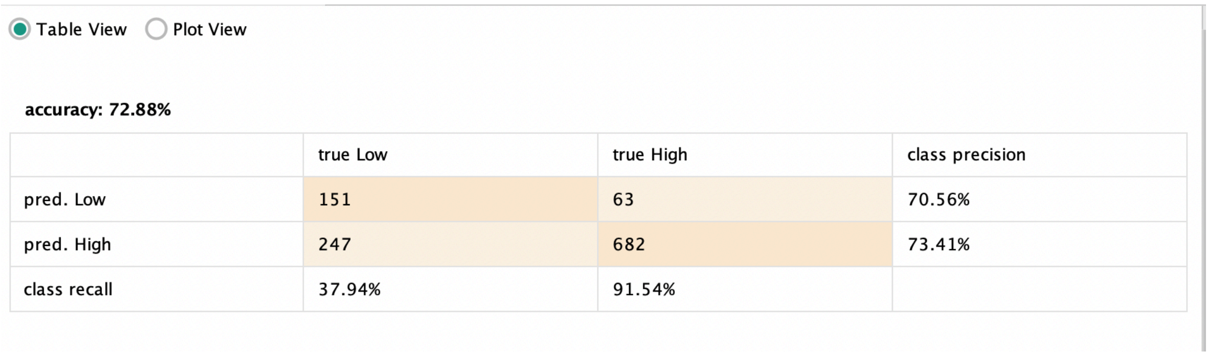


Figure 10.0 Support Vector Machine

The SVM model employs a kernel with 2000 support vectors and a bias of 1220.918 for classification. Feature weights indicate their impact on the decision boundary. A negative weight for "n_cores" suggests fewer cores increase the likelihood of a specific classification. Positive weights for "px_height" and "px_width" indicate their higher values contribute to classification. "Ram" has a moderate positive influence, while "talk_time" positively impacts classification with a notable weight. "Price_range" holds substantial weight, signifying significant influence on the decision-making process. Overall, these parameters collectively shape the SVM's effective decision boundary for classification.

Confusion Matrix for SVM



accuracy: 72.88%

	true Low	true High	class precision
pred. Low	151	63	70.56%
pred. High	247	682	73.41%
class recall	37.94%	91.54%	

The confusion matrix indicates a classification model with 72.88% accuracy. It shows that the model performs well in predicting true high-class instances (91.54% recall) but less effectively for true low-class instances (37.94% recall).

Conclusion

In this project, we aimed to leverage AI and data analytics to categorize mobile phone prices, aligning with Mobilitas Inc.'s objective to optimize their product offerings. Our comprehensive analysis, employing methodologies such as logistic regression and support vector machines, revealed significant relationships between phone features and their pricing categories. Notably, features like RAM, battery power, and screen resolution emerged as key determinants of price range.

While our methodologies provided robust insights, we acknowledge the limitations due to dataset scope and recommend further research incorporating broader market data and consumer behavior analysis. These findings have vital implications for Mobilitas Inc., offering a data-driven foundation for strategic decision-making in product development and pricing. Additionally, our research contributes valuable insights to the mobile phone industry, highlighting the importance of feature-based pricing strategies in a highly competitive market.

Appendixes

Appendix A: R Code for Decision trees

```
# Loading libraries
library(rpart)
library(rpart.plot)
library(caret)
library(lattice)

#Reading the Data
data = read.csv("train.csv")

# Specifying target variable
target <- "price_range"

# Defining subsets of predictors
subset1 <- c("battery_power")
subset2 <- c("ram")
subset3 <- c("px_height")
subset4 <- c("int_memory")
subset5 <- c("px_width")

View(data)

# Building decision tree for subset 1
model1 <- rpart(formula = as.factor(price_range) ~ .,
             data = data[, c(subset1, target)],
             control = rpart.control(minsplit = 20, minbucket = 7, cp = 0.01))

# Plotting tree for subset 1
rpart.plot(model1)

# Building decision tree for subset 2
model2 <- rpart(formula = as.factor(price_range) ~ .,
             data = data[, c(subset2, target)],
             control = rpart.control(minsplit = 20, minbucket = 7, cp = 0.01))

# Plotting tree for subset 2
rpart.plot(model2)

# Building decision tree for subset 3
model3 <- rpart(formula = as.factor(price_range) ~ .,
               data = data[, c(subset3, target)],
               control = rpart.control(minsplit = 20, minbucket = 7, cp = 0.01))

# Plotting tree for subset 3
rpart.plot(model3)


# Building decision tree for subset 4
model4 <- rpart(formula = as.factor(price_range) ~ .,
               data = data[, c(subset4, target)],
               control = rpart.control(minsplit = 20, minbucket = 7, cp = 0.01))

# Plotting tree for subset 4
rpart.plot(model4)

# Building decision tree for subset 5
model5 <- rpart(formula = as.factor(price_range) ~ .,
               data = data[, c(subset5, target)],
               control = rpart.control(minsplit = 20, minbucket = 7, cp = 0.01))

# Plotting tree for subset 5
rpart.plot(model5)


#MODEL EVALUATION

set.seed(123)  # Setting seed for reproducibility
train_indices <- sample(1:nrow(data), 0.8 * nrow(data))  # 80% for training, 20% for testing

train_data <- data[train_indices, ]
test_data <- data[-train_indices, ]


# Function to make predictions and calculate accuracy
evaluate_model <- function(model, test_data) {
  predictions <- predict(model, newdata = test_data, type = "class")
  confusion_matrix <- table(predictions, test_data$price_range)
  accuracy <- sum(diag(confusion_matrix)) / sum(confusion_matrix)
  return(list(predictions = predictions, accuracy = accuracy, confusion_matrix =
confusion_matrix))
}

# Evaluating each model
evaluation_results <- list()
```

```
evaluation_results$model1 <- evaluate_model(model1, test_data)
evaluation_results$model2 <- evaluate_model(model2, test_data)
evaluation_results$model3 <- evaluate_model(model3, test_data)
evaluation_results$model4 <- evaluate_model(model4, test_data)
evaluation_results$model5 <- evaluate_model(model5, test_data)


# Displaying accuracy for each model
for (i in 1:5) {
  cat("Model", i, "Accuracy:", evaluation_results[[paste0("model", i)]]$accuracy, "\n")
}


# Displaying confusion matrix for each model
for (i in 1:5) {

  cm <- evaluation_results[[paste0("model", i)]]$confusion_matrix

  cat("Confusion Matrix for Model", i, ":\n")
  print(cm)
  cat("\n")
}
```

Works Cited

1. Abhishek Sharma. (2017). Mobile Price Classification Data Set. Kaggle. https://www.kaggle.com/datasets/iabhishekofficial/mobile-price-classification/data (Accessed October 10, 2022)
2. Burger, S.V. (2018). Introduction to Machine Learning with R: Rigorous Mathematical Analysis. O'Reilly Media.
3. Provost, F., & Fawcett, T. (2013). Data Science for Business: What You Need to Know about Data Mining and Data-analytic Thinking. O'Reilly Media.
4. RapidMiner. (2022). Machine Learning for Predictive Maintenance. https://rapidminer.com/blog/machine-learning-predictive-maintenance/ (accessed October 15, 2022).
5. Vijay Kotu, Bala Deshpande. (2014). Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner.
6. Peng, R. D. (2016). R programming for data science. Leanpub.
7. Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. IEEE Intelligent Systems and their Applications, 13(4), 18-28. https://doi.org/10.1109/5254.708428 (Accessed November 28, 2022)