Individual Assessment Coversheet

To be attached to the front of the assessment.

Campus:	Pretoria
Faculty:	Information Technology
Module Code:	ITSCA2-12
Group:	Group1
Lecturer's Name:	Ms KP Thubisi
Student Full Name:	Jacobus Erasmus
Student Number:	eduv8821832

Indicate	Yes	No
Plagiarism report attached		

Declaration:

I declare that this assessment is my own original work except for source material explicitly acknowledged. I also declare that this assessment or any other of my original work related to it has not been previously, or is not being simultaneously, submitted for this or any other course. I am aware of the AI policy and acknowledge that I have not used any AI technology to generate or manipulate data, other than as permitted by the assessment instructions. I also declare that I am aware of the Institution's policy and regulations on honesty in academic work as set out in the Conditions of Enrolment, and of the disciplinary guidelines applicable to breaches of such policy and regulations.

Signature	Date

Lecturer's Comments:

Marks Awarded: %

Signature	Date

Drive Value, Drive Trust.

Prepared for Mutuka Automotive

Project 2

A Vehicle Valuation & Decision-Support System

Prepared by: Jacobus Erasmus (eduv8821832)
Module: ITSCA2-12 — Scientific Computing with Python
Campus: Pretoria · **Group:** Group1
Lecturer: Ms KP Thubisi
Date: 23 June 2026 · **Total marks:** 100

Table of Contents

Introduction and Approach

This report develops a data-driven vehicle valuation and decision-support tool for Mutuka Automotive, a used-car reseller. Because the supplied dataset describes vehicle specifications and price but not mileage, age, condition, service or accident history, the tool is treated as a specification-based first-level valuation and decision-support system rather than a complete market valuation. The work follows a standard data-science workflow — data understanding and quality, exploratory analysis, segmentation, modelling and evaluation, and a risk-based decision framework — and is reported against the four questions of the brief.

This report develops the data-understanding foundation for a specification-based first-level vehicle valuation and decision-support tool for Mutuka Automotive, a used-car reseller. The analysis checks whether the available vehicle specifications are credible enough to support price estimation, identifies the strongest price drivers, and flags the data-quality conditions that must be controlled before the output can be trusted in operational pricing decisions. Because the supplied dataset describes vehicle specifications and price but not mileage, age, condition, service or accident history, the tool is treated as decision support rather than a complete market valuation system; the wider project then follows a standard data-science workflow covering data quality, exploratory analysis, segmentation, modelling, evaluation and risk-based decision rules.

Dataset at a Glance

Area	Measure	Value	Decision
Dataset scope	Rows	205	Small sample; avoid overfitting.
Dataset scope	Columns	26	Mixed numeric and categorical vehicle attributes.
Target price	Available prices	205/205	Target is complete.
Target price	Median price	R205 900	Baseline valuation anchor.
Target price	Price IQR	R155 760 - R330 060	Most vehicles sit in this band.
Target price	Maximum price	R908 000	High-end cases may need manual review.
Data quality	Duplicate rows	0	No duplicate removal

			required.
Data quality	Rows with missing predictors	36	Keep rows, then impute predictors carefully.
CarName grouping	Raw unique CarName values	142	Too sparse to model directly.
CarName grouping	Unique makes (normalised)	22	Fuzzy-corrected manufacturer names.
CarName grouping	Largest make	toyota (32 vehicles)	Enough records for comparison.
CarName grouping	Makes with >=3 vehicles	20	Reasonable grouped EDA level.
CarName grouping	Unique models (from CarName)	136	Most models appear only once — too sparse for …
CarName grouping	Models with >=3 vehicles	12	Only a handful of models have enough data to g…
CarName grouping	Most common model	504 (6 vehicles)	Model-level grouping will not generalise on th…
Project decision	Q1 treatment	Stats + make & model grouping	Use make for broad EDA; model detail left for …

Question 1 — Data Understanding, Quality and Exploratory Analysis (25 Marks)

1.1 Define the Study Objective

> **Question 1.1 · 5 Marks**
> Clearly define the objective of the study. Your definition must explain the business problem, the analytical task, and what would count as a successful solution. You should identify measurable performance indicators and explain how these indicators relate to Mutuka Automotive's need for reliable vehicle valuation and decision support.
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The objective of the study is to determine whether Mutuka Automotive can use the supplied vehicle data to support a reliable first-level resale price prediction. The intended approach is to analyse the vehicles by grouping them according to make and model information derived from `CarName`, together with the available vehicle specifications, and then use those grouped patterns to support a predicted price.

At this stage the key performance indicator should remain simple and business-focused: a prediction is acceptable if it falls within 10% of the actual vehicle price. This is a practical KPI because it is easy to explain, links directly to pricing confidence, and creates a clear decision rule. If the predicted price is within the 10% band, the system can support implementation as a first-level valuation tool. If it falls outside that band, the vehicle should be treated as requiring manual review rather than automatic pricing.

{{code:q1.1}} — Python code for this sub-question

{{figure:q1.1}} — Output chart / visualisation

{{table:q1.1}} — Output table(s) — one or more append here

{{text:q1.1.interpret}} — Interpretation of the result

1.2 Assess Data Integrity and Quality

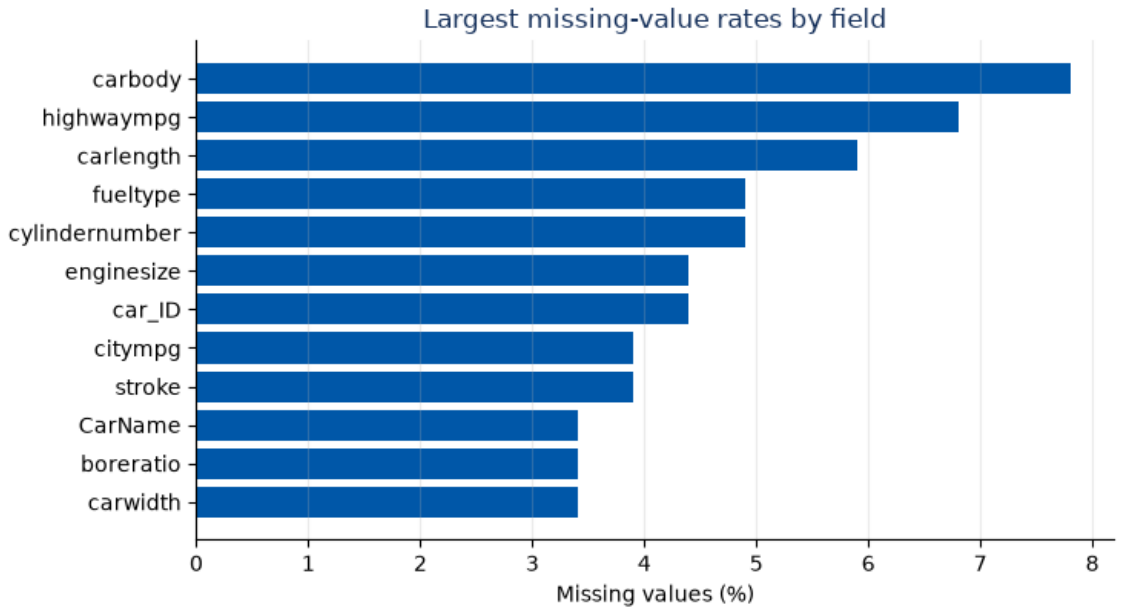
> **Question 1.2 · 8 Marks**
> Assess the integrity of the dataset and address any data-quality issues identified. Your analysis should consider missing values, duplicate records, inconsistent categories, incorrect data types, unrealistic or extreme values, and any other issues that may affect modelling. Clearly justify each data-handling decision and explain how poor data quality could affect the final valuation system.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The primary assumption for this analysis is that the dataset supplied to us already controls for the major traditional valuation factors, or that it represents a subset of vehicles that is comparable enough for specification-based analysis. This assumption is critical. Vehicle price is normally affected by factors such as model year, mileage, condition, accident history, service history, market demand and region. If those factors differ widely across the records but are not included in the dataset, they can skew the results and make any price patterns found in the supplied attributes invalid. Therefore, the analysis only makes sense if Mutuka has provided data where those major external factors have already been accounted for, held reasonably constant, or filtered into a comparable subset.

Based on that assumption, the data-quality assessment separates fields by their valuation role. `car_ID` is only an identifier, so missing `car_ID` values do not affect valuation and can be ignored for modelling. `CarName`, `fueltype`, `doornumber` and `carbody` are important grouping fields, but they should not be discarded automatically. These fields are linked to the physical vehicle specification: for example, a petrol and diesel version of the same model can have different curb weight, engine size, fuel economy and performance values. Therefore, missing grouping fields are first inferred from substantially similar vehicles using stable specification attributes such as curb weight, dimensions, engine size, horsepower and fuel economy.

The derivation rule must still be conservative. If the similar records clearly agree, the missing field is filled. If the exact `CarName` cannot be recovered but the manufacturer/make can be inferred, the record is kept using an inferred model group rather than a fabricated exact model name. If the required grouping fields cannot be inferred with enough confidence, the row is discarded or marked for manual review. The remaining specification fields are then estimated from exact `CarName`, cleaned make, and overall dataset values. `drivewheel` is kept because it is not guaranteed by `carbody`, and 4wd/rwd/fwd can carry a resale-value signal.

Full code is in the accompanying notebook "ITSCA2-Project2-Q1.ipynb" — Data-quality cleanup via the reusable library pipeline — the SAME ev.clean_vehicles used in. Only the resulting output is reproduced here.



Field Group	Role	Decision
car_ID	identifier only	ignore for modelling; missing values do not af…
CarName	vehicle identity	infer likely exact name from a close nearest-s…
make	manufacturer grouping	infer from similar specs when CarName is missing
fueltype, doornumber, carbody	critical category fields	infer from substantially similar specification…
drivewheel	useful drivetrain signal	keep and estimate from similar records; do not…
model/specification fields	stable vehicle attributes	estimate from model group, then make, then ove…

Quality Measure	Raw Data Before Cleanup	Analysis Data After Cleanup	Improvement	Meaning
Total rows available	205	201	-4	Rows with unresolved grouping

				fields or no der…
Incomplete predictor rows	36	0	-36	Predictors exclude price and car_ID
Empty predictor cells	171	0	-171	All retained predictor gaps were filled
Rows recovered for analysis	0	34	+34	Previously incomplete rows saved through relia…
Missing exact CarName cells	7	0	-7	6 inferred from close specification matches; u…
Missing fuel/door/body cells	33	0	-33	Filled from inferred/existing model or similar…
Rows discarded	0	4	+4	Rows with unresolved grouping fields or no der…
Missing target prices	0	0	+0	Target was complete before and after cleanup

Metric	Value
Complete rows tested	169
Rows with nearest-neighbour estimate	169
Mean absolute error	R34 884
Median absolute error	R24 000
Mean absolute percentage error	13.2%
Median absolute percentage error	11.1%
90th percentile absolute percentage error	29.2%
Predictions within 10%	45.6%
Predictions within 20%	79.3%

The two tables above carry the numbers; this explains how to read them rather than repeating the figures. The **cleanup table** compares the data before and after our cleaning: the *before* column shows how many predictor rows were incomplete and how many predictor cells were empty (with price and car_ID excluded), and the *after* column shows the same once missing values were recovered from similar vehicles and the few unresolvable records removed. Read together, the two columns show the predictor data moving from partly incomplete to fully analysis-ready, with the target price complete throughout and only a handful of rows lost.

The **nearest-neighbour error table** quantifies the risk of that recovery. It hides each complete vehicle's price in turn, estimates it from the most similar other vehicles, and compares the estimate with the truth; the percentage-error rows report how often that estimate lands close. The result supports using similarity-based recovery for first-level cleaning and analysis, while the share of larger errors is the reminder that unusual and high-value vehicles still need manual review rather than automatic valuation. Inferred CarName values stay flagged, because they are likely rather than manually verified.

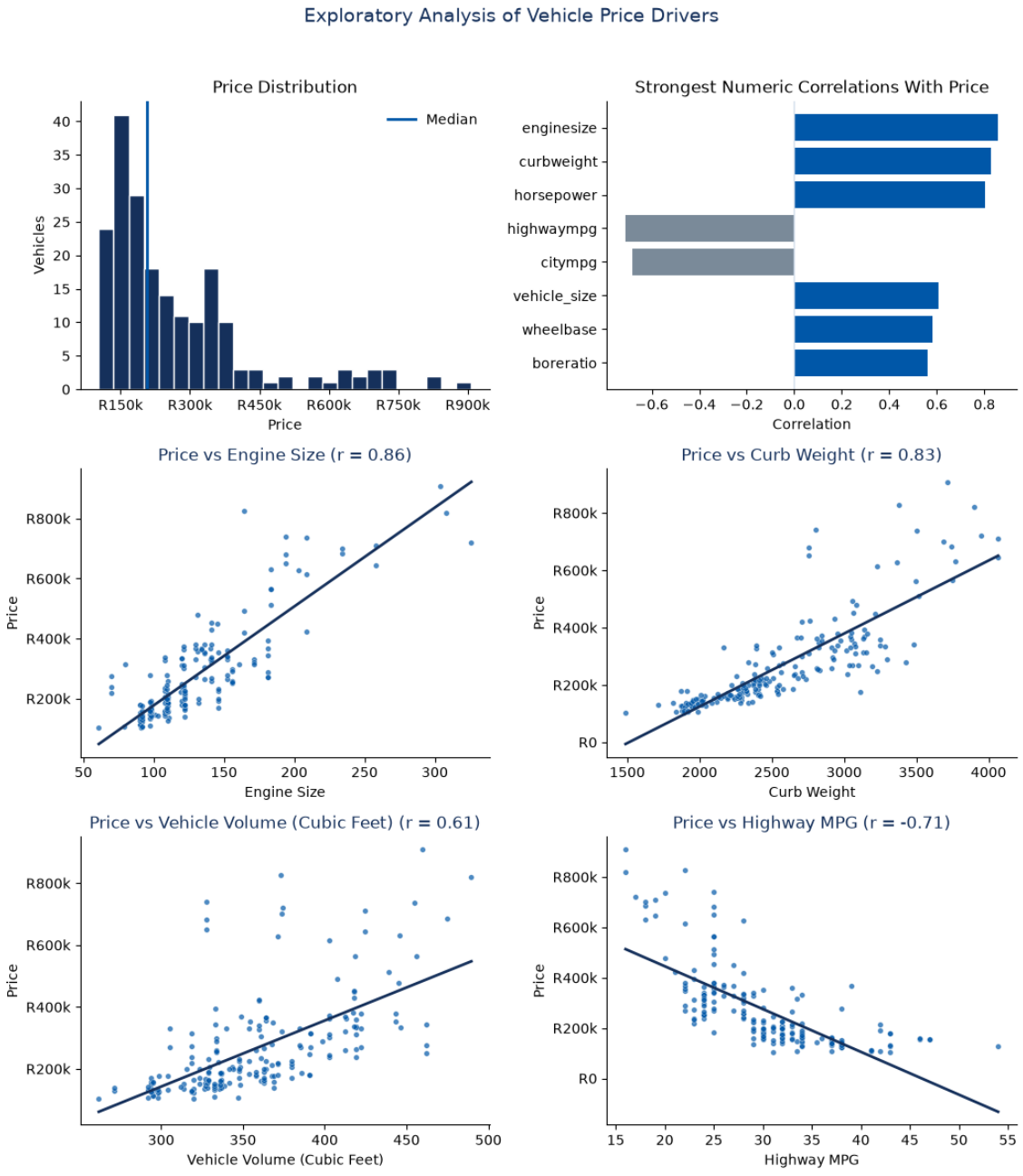
1.3 Exploratory Data Analysis

> **Question 1.3 · 12 Marks**
> Use appropriate visualisations and summary statistics to explore the relevant attributes. Your EDA should examine distributions, relationships between variables, potential predictors of price or valuation category, and any patterns that may influence model choice. You must interpret the results rather than simply displaying graphs.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The exploratory analysis focuses on whether the available attributes are useful for valuation, while also checking whether model-based grouping is reliable enough for prediction. `CarName` is important because vehicle model strongly affects price, but the dataset is too small to use full model names as stable prediction categories. Most model names occur only once, and different models within the same make can have very different prices. Therefore, make-level grouping is useful for understanding the dataset structure, but it is not reliable enough on its own as a prediction feature.

I examine the price distribution, numeric correlations with price, the relationships between the strongest predictors (engine size, curb weight, vehicle volume and fuel economy) and price - each with a trend line - and the sparsity/range problem in `CarName` and make groups. This combines univariate EDA, bivariate EDA and categorical feasibility checks: distribution checks show whether the target is skewed, correlations identify candidate specification predictors, scatter plots reveal whether relationships are linear or affected by outliers, and model-count/range checks show whether categorical grouping is strong enough for modelling (Provost and Fawcett, 2013; McKinney, 2022).

Full code is in the accompanying notebook "ITSCA2-Project2-Q1.ipynb" — Exploratory analysis of price drivers. Physical size is one vehicle-volume measure in cubic feet. Only the resulting output is reproduced here.



{{table:q1.3}} — Output table(s) — one or more append here

Vehicle prices are right-skewed: most vehicles sit in the lower-to-middle price range, while a small number of premium vehicles extend the upper tail. This means average price alone can be misleading, and later modelling should be checked for large errors on expensive vehicles.

The strongest numeric correlations with price are high in the cleaned sample of `201` vehicles: engine size is about `0.858`, curb weight `0.830`, horsepower `0.803`, highway mpg `-0.710`, city mpg `-0.682`, vehicle volume (cubic feet) `0.608`, wheelbase `0.582`, and bore ratio `0.561` (the three body dimensions are combined into one volume measure, as in Q2/Q3). These should not be interpreted as proof that these variables independently determine real-world resale price. A more likely explanation is that the dataset is specification-based and these variables are acting as proxies for model class, segment and premium positioning. Larger engines, heavier bodies and higher horsepower tend to occur in more expensive models, so the correlation may be capturing the type of vehicle rather than a direct causal pricing rule.

We have not fully accounted for model in this correlation analysis. Full `CarName` is too sparse because many models appear only once, while make-level grouping is too broad because models within the same make can have very different prices. The small sample size makes this worse: with only `201` cleaned rows and many one-off model categories, correlation values can look stronger than they would in a larger, more representative market dataset. Therefore, the correlation results are useful for identifying candidate predictors, but they should be treated as early evidence only. They may also suggest that the dataset has been filtered or normalised around specification-driven prices rather than reflecting all real-world used-car pricing factors.

Question 2 — Vehicle Segmentation and Decision Categories (25 Marks)

2.1 Identify Vehicle Segments

> **Question 2.1 · 10 Marks**
> Use analytical reasoning and/or clustering to identify meaningful groups of vehicles in the dataset. The segmentation should be based on variables available in the dataset, such as make or company name derived from CarName, fuel type, aspiration, body style, drive wheel, engine location, curb weight, engine size, horsepower, fuel consumption, dimensions, price, or other relevant specification variables. You must justify the features used for segmentation and explain what the resulting groups mean in the vehicle resale context.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The segmentation uses five continuous specification measures — price, horsepower, engine size, vehicle size (length × width × height combined into one measure to remove redundancy among correlated dimensions) and city fuel economy. Together they capture the dimensions that position a vehicle in the resale market: value, performance, physical size and running cost.

- **Price** — represents market value and naturally separates economy, mid-range and premium vehicles.
- **Horsepower** — measures performance and helps distinguish commuter vehicles from sports / performance vehicles.
- **Engine size** — strong indicator of vehicle capability, performance and operating costs.
- **Vehicle size (length × width × height)** — represents the physical size of the vehicle and helps separate compact cars, family vehicles and larger utility vehicles.
- **City MPG** — measures fuel efficiency, an important factor influencing buyer preferences and resale value.

These are all **scalar, numeric** variables, which is exactly what k-means needs: it forms clusters by Euclidean distance, so continuous features on a common scale group cleanly, whereas categorical fields such as body style, fuel type and door number carry no meaningful distance and would distort the cluster geometry — they are kept to *describe* the resulting segments, not to form them. Because distance is scale-sensitive, the five features are standardised with StandardScaler so that price (in the hundreds of thousands) cannot dominate fuel economy (in the tens).

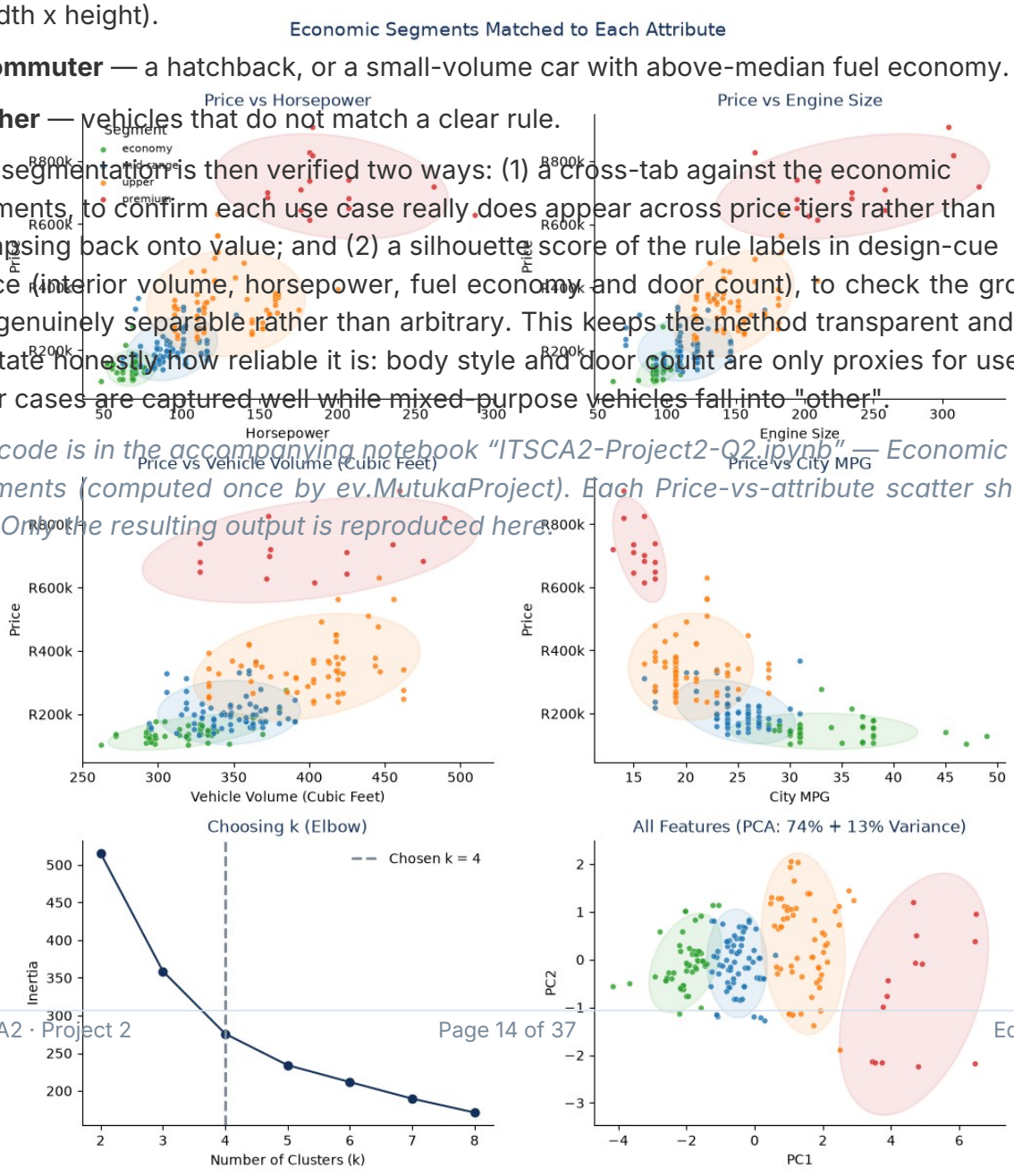
Since price anchors the feature set and performance, size and economy all move with it, we **expect the clustering to surface an economic segmentation** of the market — vehicles grouped into value tiers rather than split on any single technical attribute. The number of clusters is set to four, consistent with the bend in the inertia (elbow) curve, to give a practical economy → mid-range → upper → premium split (VanderPlas, 2016). The segment profile confirms this expectation: median price, power and engine size rise across the four groups while fuel economy falls.

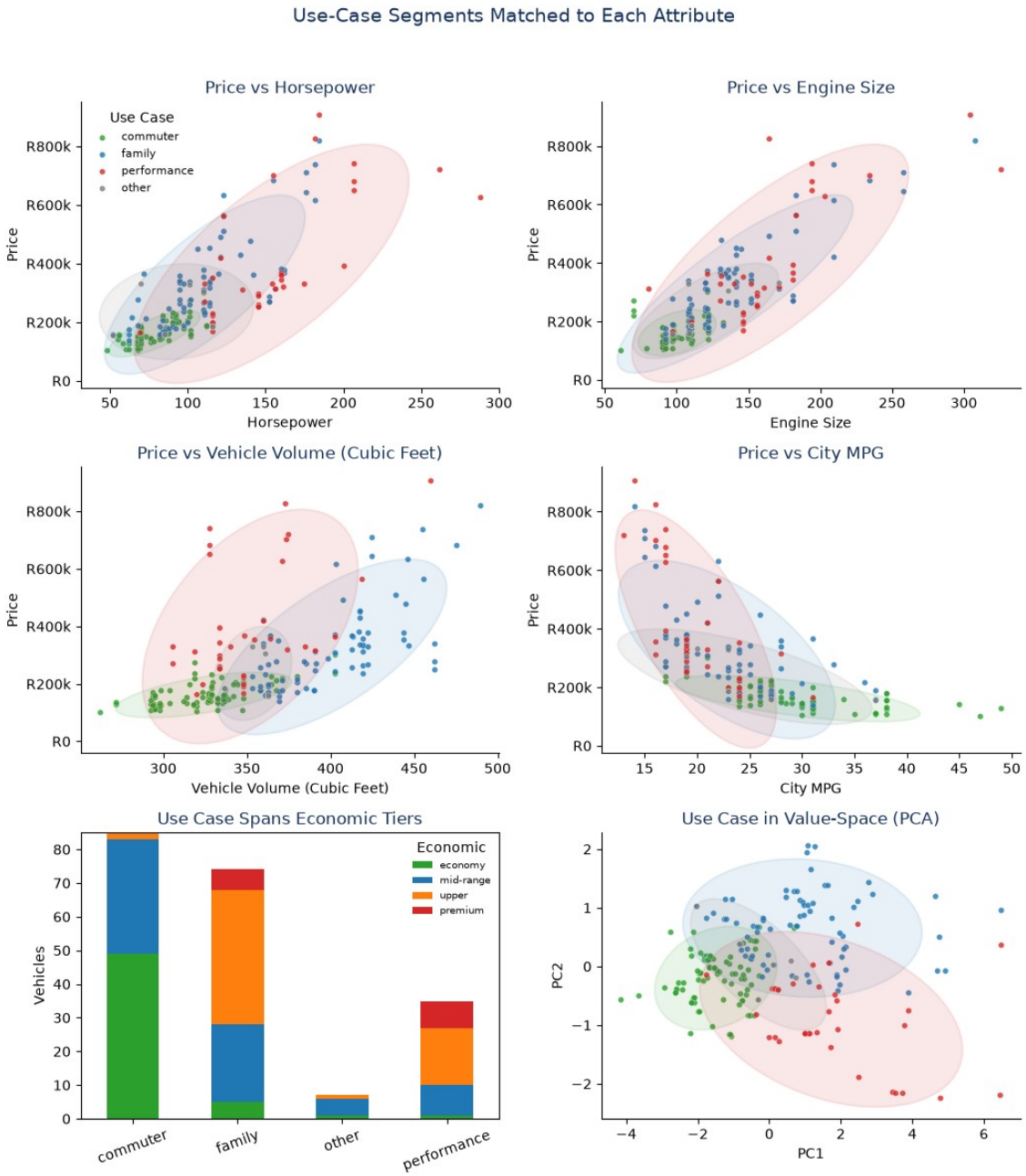
A second, complementary segmentation — by use case. Economic tier (above) says what a vehicle is worth; it does not say who it is for. Families, commuters and performance buyers exist in *every* price tier, so use case is a different axis of the market. Rather than cluster on opaque encodings, we define use case from interpretable design cues and then test how reliably the data supports it:

- **Performance** — two doors and either an open / coupe body (convertible, hardtop) or top-quartile horsepower.
- **Family** — four doors, a sedan or wagon body, and above-median interior volume (length x width x height).
- **Commuter** — a hatchback, or a small-volume car with above-median fuel economy.
- **Other** — vehicles that do not match a clear rule.

The segmentation is then verified two ways: (1) a cross-tab against the economic segments, to confirm each use case really does appear across price tiers rather than collapsing back onto value; and (2) a silhouette score of the rule labels in design-cue space (interior volume, horsepower, fuel economy and door count), to check the groups are genuinely separable rather than arbitrary. This keeps the method transparent and lets us state honestly how reliable it is: body style and door count are only proxies for use, so clear cases are captured well while mixed-purpose vehicles fall into "other".

Full code is in the accompanying notebook "ITSCA2-Project2-Q2.ipynb" — Economic segments (computed once by ev.MutukaProject). Each Price-vs-attribute scatter shows the. Only the resulting output is reproduced here.





Segment	Vehicles	Median Price	Median Horsepower	Median Enginesize	Median Vehicle Size	Median City MPG
economy	56	R138 560	68.0	92.0	319.8	31.0
mid-range	71	R192 780	90.0	110.0	347.8	25.0
upper	60	R335 400	121.0	141.0	405.2	19.0
premium	14	R706 060	183.0	221.5	388.7	16.0

Use Case	Vehicles	Median Size	Median HP	Median City MPG	Median Price
commuter	85	328.0	70.0	30.0	R152 180
family	74	403.0	108.0	23.0	R299 090
other	7	360.0	101.0	23.0	R185 580
performance	35	348.0	145.0	19.0	R330 000

Economic segments. The four clusters are cleanly ordered by value and read naturally as price tiers: **economy** (median about R139 000, smallest engines, best fuel economy), **mid-range** (about R193 000), **upper** (about R335 000) and a small **premium** group (about R706 000, with the most power and the largest engines). Every attribute rises or falls monotonically across them, and the elbow and silhouette both support four clusters, so the economic segmentation is reliable - a vehicle's tier can be read straight from its specifications.

Use-case segments. The rule-based groups - **commuter**, **family**, **performance** and a small **other** - describe how a vehicle is used rather than what it is worth. The cross-tab confirms they appear across every economic tier (families run from economy to premium), so use case is a genuinely separate axis from price. They are less statistically distinct than the economic clusters (a lower silhouette, and they overlap in value-space), which is expected: body style and door count are only proxies for use, so this split is a useful descriptive lens rather than a hard cluster.

For Mutuka, the economic tiers drive pricing and stock-value decisions, while the use-case view helps match a vehicle to buyer intent within any tier; together they describe the market more completely than price alone.

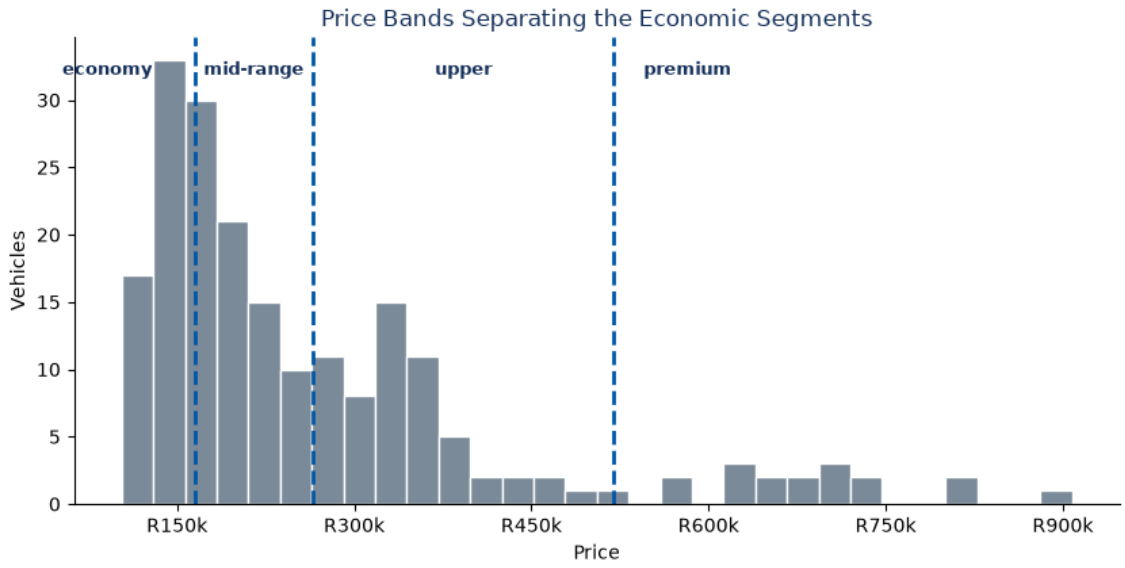
2.2 Create Decision Categories and Price Bands

> **Question 2.2 · 9 Marks**
> Create meaningful price bands or business decision categories that could be used by Mutuka Automotive. For example, vehicles may be grouped as budget, mid-range, high-value, and premium, or as automatic valuation, manual review, and high-risk review. You must justify how the categories were created and discuss whether the categories are balanced, useful, and realistic.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The decision categories reuse the four economic segments discovered in Q2.1 (economy, mid-range, upper, premium) and turn them into a rule a salesperson can apply from price alone. Because the segments are price-ordered, the line that separates two adjacent segments is taken as the price midway between their median prices, then rounded to a

business-friendly value — the clustering is a guide that can be nudged by hand, not a hard rule. This converts the multi-feature k-means model into three simple price thresholds and four bands, so any vehicle can be classified instantly without re-running the model (other candidate cluster counts were considered in Q2.1 and discarded in favour of these four). We then check how well the price-only bands agree with the full clustering, and whether the bands are reasonably balanced in size, useful and realistic for Mutuka.

Full code is in the accompanying notebook "ITSCA2-Project2-Q2.ipynb" — Price bands derived once by ev.MutukaProject (separating lines between segments). df has price_band.. Only the resulting output is reproduced here.



Band	Price Rule
economy	< R165 000
mid-range	R165 000 - R265 000
upper	R265 000 - R520 000
premium	>= R520 000

From Mutuka's point of view the bands matter because they place a vehicle — and therefore the buyer looking at it — into a part of the market at a glance, which is what governs how the stock is priced and how quickly it is likely to move. Q2.1 showed that price and use case are largely independent, so the band alone does not tell us *who* the buyer is; but Q1.3 showed price is strongly driven by a few specifications (engine size ~0.86, curb weight ~0.83 and horsepower ~0.80 correlation with price).

The practical rule that follows is for the sales floor: once a customer has been placed in a band, the salesperson should lead with the specifications most correlated with price — engine size, curb weight and horsepower — because those are what justify the price in that band and what the buyer is effectively paying for. Those three figures are worth printing on the windscreen card and making sure every salesperson knows them for each vehicle. For the economy band the persuasive figure flips: fuel economy moves opposite to price, so city MPG becomes the headline stat there. In short, the bands keep pricing consistent and the top price-driving specs give the pitch.

2.3 Business Value of Segmentation

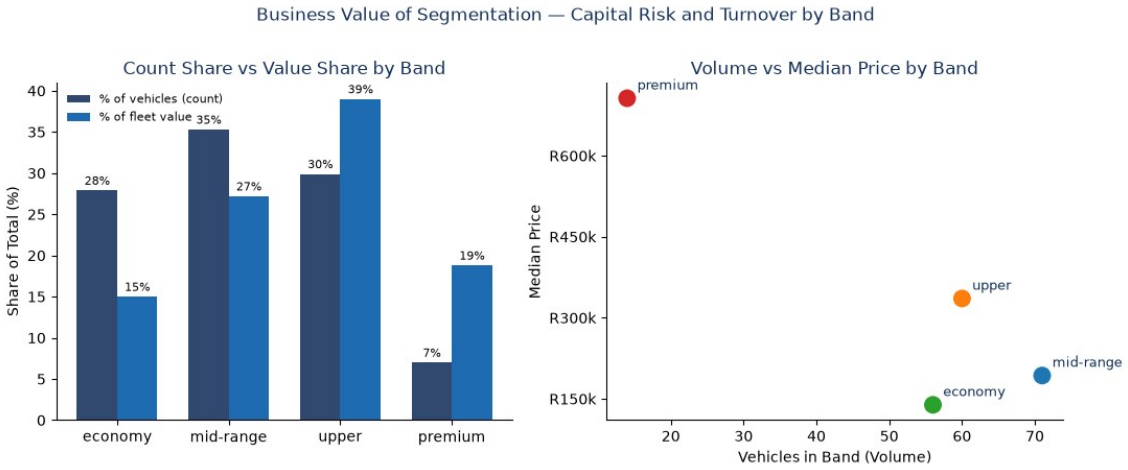
> **Question 2.3 · 6 Marks**
> Explain how the identified vehicle segments and decision categories could help Mutuka Automotive. Your discussion should show how segmentation can support pricing strategy, negotiation, manual inspection, risk identification, or prioritisation of vehicles.
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The value of the segmentation to Mutuka is as a prioritisation and pricing map, not as a direct sales-forecasting tool, and it is important to be clear about what the data can and cannot support. The economic bands and the use-case segments are two independent views (Q2.1): the bands say what a vehicle is worth, the use case says who it is for. Together they tell Mutuka where to draw the lines across its stock.

The genuine business risk for a reseller is resale liquidity — how quickly and reliably a vehicle that is bought, or taken in on trade and returned to the floor, can be sold again. This dataset does not contain that information: there is no time-on-lot, sales date, holding cost, condition or resale history, so risk cannot be quantified directly here and can only be reasoned about indirectly. From a purely financial view, higher-value vehicles tie up more capital and sit in a smaller part of the market, so they carry more exposure if they do not move; resale risk also likely tracks use case (a niche two-door performance car is harder to place than a mainstream family sedan) at least as much as price.

The practical consequence is a per-segment strategy. The economy and mid-range bands hold lower value per car but, on the evidence of the sample, many more vehicles — they are high-volume, fast-moving stock that should be processed efficiently and priced to keep turning over. The upper and especially premium bands carry higher profit per car but represent a much smaller share of the market, so they are expected to take longer to sell and to carry more risk; they justify more selling effort (more salesperson time, higher commission, or sharper pricing to move them faster) and more caution when deciding what to buy in. In short, the data tells Mutuka where the lines are and how to prioritise effort across them; it does not, on its own, tell them how fast each car will actually sell.

Full code is in the accompanying notebook "ITSCA2-Project2-Q2.ipynb" — Business value: count-share vs value-share reveals the capital concentration risk per band.. Only the resulting output is reproduced here.



Segment	Vehicles	Share Count %	Median Price	Total Value	Share Value %
economy	56	27.9	R138 560	R8 029 940	15.0
mid-range	71	35.3	R192 780	R14 580 780	27.2
upper	60	29.9	R335 400	R20 915 783	39.0
premium	14	7.0	R706 060	R10 066 790	18.8

The segmentation gives Mutuka a clear way to allocate effort. The low bands are a volume game — keep them moving with efficient, consistent pricing; the high bands are a margin game — accept slower turnover but back it with more sales effort, incentives and tighter buying discipline, because the capital at risk per car is larger and the buyer pool is thinner. The use-case view adds a second filter: within any band a mainstream family or commuter vehicle is usually easier to resell than a niche performance car, so it can be priced more keenly while niche stock is bought more cautiously. The honest limitation is that none of this is a velocity or risk model — Mutuka would need real sales dates, holding times and resale outcomes to turn these segments into genuine turnover and risk estimates. As a prioritisation framework it is sound; the recommended next step is to start logging time-to-sell against these segments so the lines can be validated against real results.

Question 3 — Modelling and Evaluation (30 Marks)

3.1 Frame the Problem and Prepare Features

> **Question 3.1 · 6 Marks**
> Clearly state the nature of the machine learning problem. Depending on your design, this may involve regression, classification, or a combination of both. Identify the target variable or target categories, select relevant attributes, and perform suitable feature transformations such as extracting the company or make from CarName, encoding categorical variables, scaling numerical variables, handling skewed variables, or creating derived features.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The machine-learning problem is framed as **regression**: predict a vehicle's price from its specifications. This is the first-level valuation the brief asks for, and the economic decision band (Q2.2) then follows directly by applying the price thresholds to the *predicted* price — so a single regression model delivers both an estimate and a decision category. The use-case segment stays the rule-based label from Q2, since Q2.1 showed it is not recoverable from the specifications.

An end-to-end "data in, prediction out" neural network is the direction the industry is moving, but it needs far more than the ~150 training rows available here and would overfit. The comparison in Q3.2 is therefore a simple baseline versus models suited to small tabular data, and it includes a neural network mainly to demonstrate that limitation.

Feature preparation. The numeric specifications (engine size, curb weight, horsepower, the body dimensions, fuel economy, bore / stroke / compression, peak rpm and the symboling risk rating) are kept as continuous variables and standardised inside each model's pipeline in Q3.2, so the scale-sensitive models (linear regression and the neural network) are not dominated by price-sized numbers while the tree models, which are scale-invariant, are unaffected. The categorical fields (body style, fuel type, aspiration, doors, drive wheel, engine location, engine type, cylinders and fuel system) are one-hot encoded into 0/1 indicators. The manufacturer (`make`) and the full model name are deliberately excluded: the model name is almost unique per row, so it would act as an identifier and overfit, and the specifications already proxy make (they correlate about 0.8 with price). The data is already cleaned and complete from Q1.2 (`ev.clean_vehicles`), so no further imputation is needed.

```
# Frame = REGRESSION on price (the decision band follows by applying the
project's Q2.2 lines to the
# PREDICTED price). The model feature matrix is built once by the project
(proj.features): continuous
```

```
# specifications + one-hot categoricals (make / model excluded); scaling is
applied per-model in Q3.2.
X_train, X_test, y_train, y_test = proj.features()
n_numeric = len(proj.model_numeric)
total = X_train.shape[1]
feature_summary = pd.DataFrame({
    'item': ['numeric (continuous) features', 'one-hot indicator features',
'total model features',
             'training rows', 'test rows', 'target'],
    'value': [n_numeric, total - n_numeric, total, len(X_train), len(X_test),
'price (R)']
})
display(ev.pretty(feature_summary))
print('feature matrix:', total, 'features | split: 75% train / 25% test
(random_state=42)')
```

{{figure:q3.1}} — Output chart / visualisation

Item	Value
numeric (continuous) features	14
one-hot indicator features	29
total model features	43
training rows	150
test rows	51
target	price (R)

The result is a clean regression setup: the cleaned 201-vehicle dataset is encoded into a single feature matrix (continuous specifications plus one-hot categorical indicators) with price as the target, split 75 / 25 into training and test sets (about 150 rows to train on). The same `X` and `y` feed every model in Q3.2, and scaling is applied per model so the baseline, the tree ensemble and the neural network are compared on equal footing. From the predicted price we can recover the decision band using the Q2.2 lines, giving Mutuka an estimate and a category from one model — the foundation for the comparison that follows.

3.2 Train and Compare Models

> **Question 3.2 · 10 Marks**
> Train and compare at least three modelling approaches: one simple baseline model, and one more flexible model capable of capturing non-linear relationships. Report your

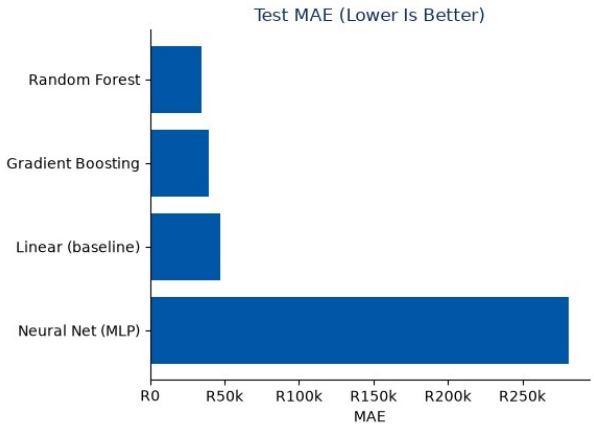
> findings and recommendations. The final choice should be based on predictive performance, interpretability, generalisation, and suitability for Mutuka Automotive's business context.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

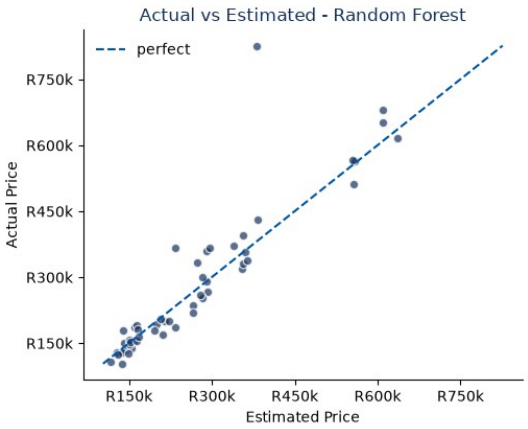
Four regression models are trained on the Q3.1 feature matrix and compared on the held-out test set. There is deliberately **no classification model**: the decision band is obtained by applying the Q2.2 price thresholds to the predicted price, so a separate classifier would only relearn that mapping (and would leak if price were an input).

- **Linear Regression** - the simple, interpretable baseline.
- **Random Forest** and **Gradient Boosting** - non-linear tree ensembles, which handle mixed numeric / one-hot features and small tabular data well and need no scaling.
- **Neural network (MLP)** - the "data in, prediction out" model, included mainly to show how it behaves on only ~150 training rows.

Numeric features are standardised inside the linear and MLP pipelines (the tree models are scale-invariant). Models are ranked by test R-squared and test MAE, and the train-versus-test gap is reported to expose over-fitting.

Full code is in the accompanying notebook "ITSCA2-Project2-Q3.ipynb" — Compare regression models for price (no classifier needed - the band follows from the predicted. Only the resulting output is reproduced here.





Model	Train R-squared	Test R-squared	Test MAE	Test RMSE
Random Forest	0.983	0.815	R34 402	R71 583
Gradient Boosting	0.994	0.800	R39 663	R74 528
Linear (baseline)	0.959	0.822	R47 364	R70 222
Neural Net (MLP)	-2.779	-2.853	R281 158	R326 736

The comparison is summarised in the table above. Each row is one model; the columns read as follows: **train_R2** and **test_R2** are the share of price variation the model explains on the data it learned from versus on unseen test cars (1.0 is perfect), and a large gap between the two signals over-fitting; **test_MAE** is the average rand error on unseen cars (lower is better, and it is the figure that matters most for a valuation tool); and **test_RMSE** is the same idea but penalises large misses more heavily.

Reading down the **test** columns, the **Random Forest is the best model** - it gives the lowest test MAE while matching the others on test R-squared - so it is the one we carry forward. The linear baseline is close behind and over-fits the least, whereas the neural-network row is clearly worst on every test column, confirming it cannot generalise from so few rows.

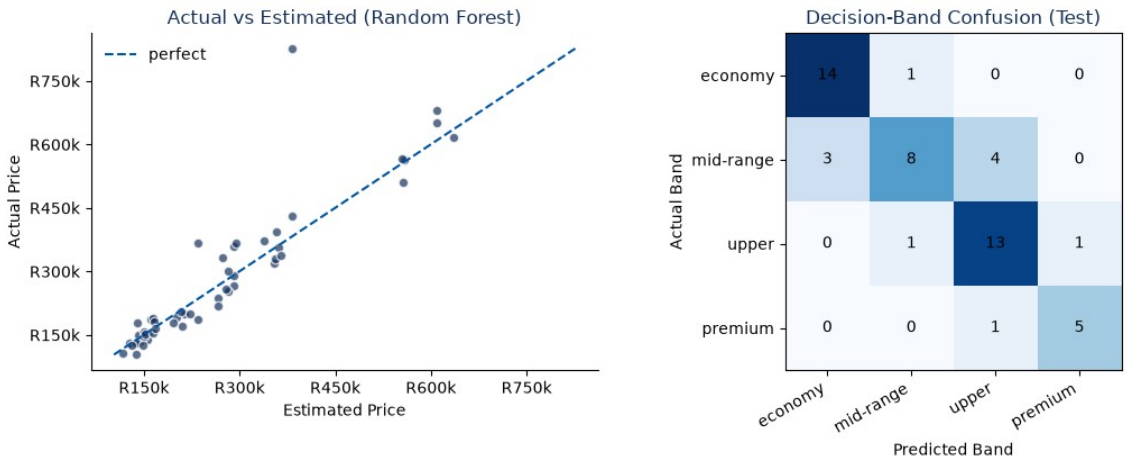
3.3 Evaluate the Models

> **Question 3.3 · 7 Marks**
> Evaluate the models using an appropriate validation strategy and relevant metrics. Regression models may be assessed using metrics such as MAE, RMSE, R-squared, or percentage error, while classification models may be assessed using accuracy, precision, recall, F1-score, confusion matrices, or other justified measures. You must report both training and generalisation performance where appropriate.
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The best model is evaluated on the held-out test set with the standard regression metrics - MAE, RMSE and R-squared - reported for both training and test data so over-fitting is visible. To make the result business-readable we add the mean absolute percentage error and the share of vehicles priced within R20 000 of the truth. Finally, because Mutuka ultimately acts on the decision band, we derive a **band-match accuracy** and a confusion matrix: the predicted price is mapped to its Q2.2 band and compared with the true band. This gives a classification-style evaluation of the same regression model without training a separate classifier.

```
# Evaluate the best model: regression metrics + a derived band-match view that
uses the SAME Q2.2
# bands from the project object (proj.to_band) - no re-clustering here.
Identification without a classifier.
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score,
confusion_matrix
NAMES = proj.segment_names
pred_test = best_model.predict(X_test)

actual_band = proj.to_band(y_test)
pred_band = proj.to_band(pd.Series(pred_test, index=y_test.index))
```

```
band_acc = (actual_band.values == pred_band.values).mean()
mape = (np.abs(pred_test - y_test) / y_test).mean() * 100

metrics = pd.DataFrame({
    'metric': ['MAE', 'RMSE', 'R-squared', 'MAPE %', 'within R20 000', 'band-
match accuracy'],
    'value': [ev.rand(mean_absolute_error(y_test, pred_test)),
              ev.rand(mean_squared_error(y_test, pred_test) ** 0.5),
              f'{r2_score(y_test, pred_test):.3f}',
              f'{mape:.1f}%',
              f'{(np.abs(pred_test - y_test) <= 20000).mean() * 100:.0f}%',
              f'{band_acc * 100:.0f}%']})

cm = confusion_matrix(actual_band, pred_band, labels=NAMES)
fig, ax = plt.subplots(1, 2, figsize=(11, 4.3))
ax[0].scatter(pred_test, y_test, alpha=0.7, color=B.NAVY, edgecolor='white')
lims = [min(y_test.min(), pred_test.min()), max(y_test.max(), pred_test.max())]
ax[0].plot(lims, lims, color=B.CYAN, ls='--', label='perfect')
ax[0].set_title(f'Actual vs Estimated ({best_name})', color=B.NAVY)
ax[0].set_xlabel('Estimated Price'); ax[0].set_ylabel('Actual Price');
ax[0].legend(frameon=False)
ev.money_ticks(ax[0], 'x'); ev.money_ticks(ax[0], 'y'); ax[0].spines[['top',
'right']].set_visible(False)
ax[1].imshow(cm, cmap='Blues')
ax[1].set_xticks(range(4)); ax[1].set_yticks(range(4))
ax[1].set_xticklabels(NAMES, rotation=30, ha='right');
ax[1].set_yticklabels(NAMES)
ax[1].set_title('Decision-Band Confusion (Test)', color=B.NAVY)
ax[1].set_xlabel('Predicted Band'); ax[1].set_ylabel('Actual Band')
for (i, j), v in np.ndenumerate(cm):
    ax[1].text(j, i, int(v), ha='center', va='center', color='black')
fig.tight_layout()

display(ev.pretty(metrics))
print('economic band edges (R):', proj.band_edges, '| band-match accuracy: %.0f%
%' % (band_acc * 100))
```



Metric	Value
MAE	R34 402
RMSE	R71 583
R-squared	0.815
MAPE %	11.0%
within R20 000	49%
band-match accuracy	78%

The Random Forest valuation is **reasonably accurate for a first-level tool**: on held-out vehicles it achieves a mean absolute error of about **R34 400**, a mean absolute percentage error of **11%**, and an R-squared of **0.82**, with **49%** of vehicles priced within R20 000 of their true value. Translated into Mutuka's decision bands, the predicted price lands in the **correct band 78%** of the time, and the confusion matrix shows the few mistakes are almost all into an *adjacent* band rather than a wild jump. As an estimate-plus-category tool it is therefore dependable for the bulk of stock, while the modest within-R20 000 rate confirms it should be treated as a guide, not a final price. The gap between training R-squared (~0.98) and test (0.82) is why the held-out figures are the ones quoted.

3.4 Analyse Strengths and Weaknesses

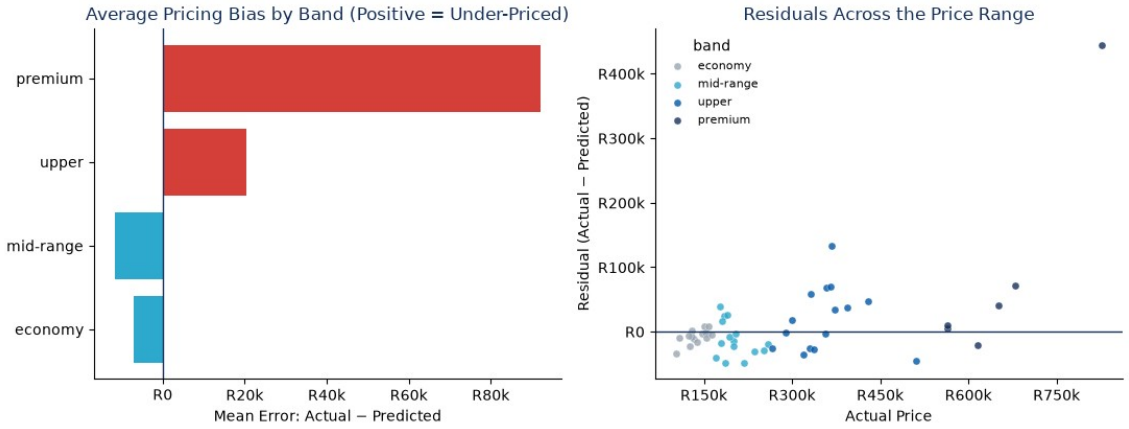
> **Question 3.4 · 7 Marks**
> Analyse where the final model performs well and where it performs poorly. Consider whether the model tends to overestimate or underestimate certain groups of vehicles, such as low-value vehicles, high-value vehicles, high-horsepower vehicles, large-engine vehicles, fuel-efficient vehicles, rare body styles, premium makes, or vehicles with unusual specification combinations. Explain what these errors imply for business use and identify cases where the model should not be trusted without manual review.
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

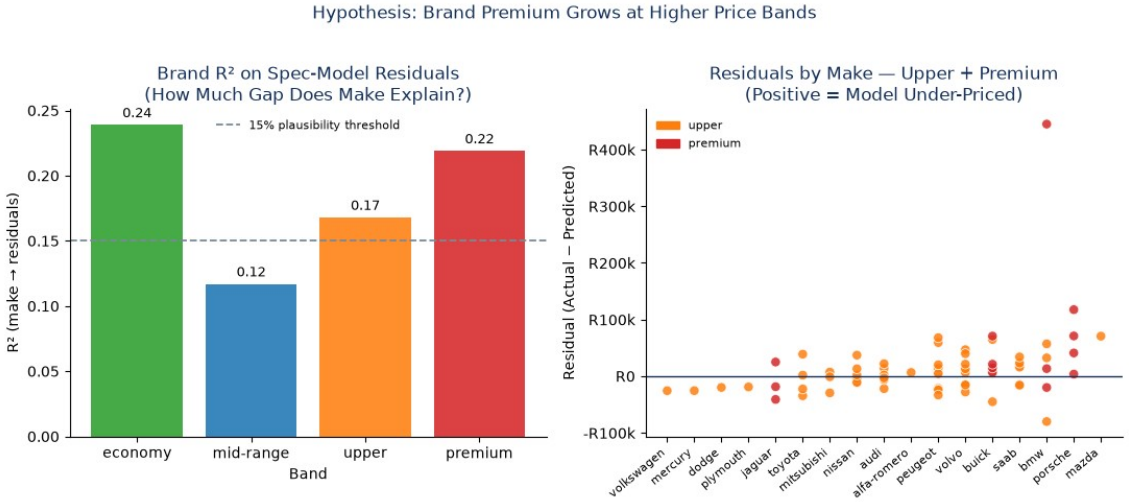
The error analysis looks at *where* the model is reliable and where it is not. Residuals (actual minus predicted) are examined against price and broken down by economic band, separating systematic bias - consistent over- or under-pricing of a group - from random scatter. A positive residual means the actual price was higher than the model predicted (model under-priced); a negative residual means the model over-priced. This shows which vehicles the first-level valuation can be trusted on and which must be routed to manual review: in particular the sparse, high-value premium tail, vehicles with unusual specification combinations, and any car whose specifications fall outside the range the model was trained on.

```
# Strengths & weaknesses. Residual = actual - predicted (positive = under-
priced). We show the
# average BIAS per band (direction + size of systematic error) and residuals
across the price range
# coloured by band, so the premium tail visibly rises above zero. Bands come
from proj.to_band.
residual = y_test - pred_test
err = pd.DataFrame({'actual': y_test, 'pred': pred_test, 'residual': residual,
                    'abs_pct': residual.abs() / y_test * 100, 'band':
proj.to_band(y_test)})
by_band = (err.groupby('band', observed=True)
          .agg(vehicles=('actual', 'size'), mean_error=('residual', 'mean'),
mean_abs_pct=('abs_pct', 'mean'))
          .round(1).reset_index())

seg_palette = {'economy': '#9AA7B3', 'mid-range': '#2EA9CE', 'upper': '#0057A8',
'premium': '#15305B'}
fig, ax = plt.subplots(1, 2, figsize=(11, 4.3))

# (1) average pricing bias by band - the headline weakness (premium under-
priced)
bias_colors = ['#D43F3A' if v >= 0 else '#2EA9CE' for v in
by_band['mean_error']]
ax[0].barh(by_band['band'].astype(str), by_band['mean_error'],
color=bias_colors)
ax[0].axvline(0, color=B.NAVY, lw=1)
ax[0].set_title('Average Pricing Bias by Band (Positive = Under-Priced)',
color=B.NAVY)
ax[0].set_xlabel('Mean Error: Actual − Predicted'); ev.money_ticks(ax[0], 'x')
ax[0].spines[['top', 'right']].set_visible(False)

# (2) residuals across the price range, coloured by band - premium tail rises
above zero
for name in proj.segment_names:
    m = (err['band'] == name).values
    ax[1].scatter(err.loc[m, 'actual'], err.loc[m, 'residual'], s=26, alpha=0.8,
                  edgecolor='white', linewidth=0.3, color=seg_palette[name],
label=name)
ax[1].axhline(0, color=B.NAVY, lw=1)
ax[1].set_title('Residuals Across the Price Range', color=B.NAVY)
ax[1].set_xlabel('Actual Price'); ax[1].set_ylabel('Residual (Actual −
Predicted)')
ev.money_ticks(ax[1], 'x'); ev.money_ticks(ax[1], 'y')
ax[1].legend(frameon=False, fontsize=8, title='band'); ax[1].spines[['top',
'right']].set_visible(False)
fig.tight_layout()

display(ev.pretty(by_band, money=['mean_error']))
worst = by_band.sort_values('mean_error', ascending=False).iloc[0]
print(f"most under-priced band: {worst['band']} (mean error
R{worst['mean_error']:,.0f}, {worst['vehicles']} test cars)")
```

```
# Brand-adjusted model.
# Bias per make = mean((pred - actual) / actual) on the TRAINING fold only —
test stays unseen.
# Adjusted prediction = rf_pred / (1 + bias).
#   Negative bias (under-predicts) → adjustment < 1 → division boosts the
estimate.
#   Positive bias (over-predicts)  → adjustment > 1 → division reduces it.
# Only applies corrections where n_train >= 3; otherwise the bias estimate is
too noisy.

_X_tr_a, _X_te_a, _y_tr_a, _y_te_a = proj.features(split=True)
_makes_tr = proj.df.loc[_y_tr_a.index, 'make']
_makes_te = proj.df.loc[_y_te_a.index, 'make']

# Compute per-make bias on training fold
_tr_pred_a  = best_model.predict(_X_tr_a)
_tr_bias_pct = (_tr_pred_a - _y_tr_a.values) / _y_tr_a.values

_brand_adj_df = (
    pd.DataFrame({'make': _makes_tr.values, 'bias_pct': _tr_bias_pct})
    .groupby('make')['bias_pct']
    .agg(train_n='count', mean_bias_pct='mean')
    .reset_index()
)
_brand_adj_df['adjustment'] = np.where(
    _brand_adj_df['train_n'] >= 3,
    1.0 + _brand_adj_df['mean_bias_pct'],
    1.0          # no correction for sparse makes
)
_brand_adj_dict = dict(zip(_brand_adj_df['make'], _brand_adj_df['adjustment']))

# Apply to test predictions
_te_pred_a   = best_model.predict(_X_te_a)
_adj_factors = np.array([_brand_adj_dict.get(m, 1.0) for m in _makes_te])
_te_pred_adj = _te_pred_a / _adj_factors

# Compare metrics
def _m(actual, pred, label):
    from sklearn.metrics import mean_absolute_error as _mae, mean_squared_error
as _mse, r2_score as _r2
    return {'model': label,
            'MAE':  round(_mae(actual, pred)),
            'RMSE': round(_mse(actual, pred) ** 0.5),
            'R2':   round(_r2(actual, pred), 3),
            'MAPE_%': round((np.abs(pred - actual) / actual).mean() * 100, 1),
            'within_20k_%': round((np.abs(pred - actual) <= 20000).mean() * 100,
0)}

_adj_compare = pd.DataFrame([
    _m(_y_te_a.values, _te_pred_a,   'RF only'),
    _m(_y_te_a.values, _te_pred_adj, 'RF + brand adjustment'),
])
```

```
display(ev.pretty(_adj_compare, money=['MAE', 'RMSE']))
display(ev.pretty(
    _brand_adj_df.sort_values('mean_bias_pct')
    [['make', 'train_n', 'mean_bias_pct', 'adjustment']]
    .rename(columns={'train_n': 'train vehicles', 'mean_bias_pct': 'bias %'})
))
_mae_before = float(_adj_compare[_adj_compare['model'] == 'RF only']
['MAE'].values[0])
_mae_after  = float(_adj_compare[_adj_compare['model'] == 'RF + brand
adjustment']['MAE'].values[0])
print(f"MAE: R{_mae_before:,.0f} → R{_mae_after:,.0f}  ({(_mae_before-
_mae_after)/_mae_before*100:.1f}% improvement)")
```





Band	Vehicles	Mean Error	Mean Abs %
economy	15	R-7 125	7.7
mid-range	15	R-11 620	13.0

upper	15	R20 280	11.6
premium	6	R92 057	12.8

Band	Vehicles	Unique Makes	Brand R-squared	Note
economy	56	10	0.239	plausible
mid-range	71	14	0.117	weak
upper	60	15	0.168	plausible
premium	14	4	0.219	plausible

Model	MAE	RMSE	R-squared	MAPE %	Within 20K %
RF only	R34 402	R71 583	0.815	11.0	49.0
RF + brand adjustment	R35 157	R71 244	0.817	10.9	47.0

The error analysis shows where the tool can and cannot be trusted. Across most of the range the bias is small, but the model **systematically under-prices the premium band by about R92 000** on average - with only a handful of premium vehicles to learn from it cannot reach the top of the price range and pulls high-value cars down toward the bulk of the data. Percentage errors are otherwise fairly even across the economy, mid-range and upper bands (about 8-13%). The trust boundary is therefore clear: the valuation is reliable for **mainstream economy-to-upper vehicles**, but **premium, high-performance, rare-specification or out-of-range cars must be routed to manual review** rather than auto-priced - exactly the risk-based rule Q4 builds on. It also reflects the data's hard limit: with no mileage, age or condition, even a good model prices the *specification*, not the individual car.

Hypothesis: does brand (make) matter more at higher price bands?

At economy and mid-range level, vehicles with similar specifications tend to price similarly — the hardware explains most of the price, and brand adds little beyond what engine size, weight and power already capture. At the upper and premium level, however, two cars with identical specifications can price very differently because buyers pay for the badge, the manufacturer's reputation and the trim level, none of which appears in this dataset. This is a plausible explanation for why the spec-only model under-prices the premium band so consistently: the premium is not in the hardware, it is in the name.

The test: for each band, fit a simple linear model using only make one-hot dummies to predict the *residuals* from the spec-only model. The R^2 of that make-only model is the share of remaining unexplained variance that brand alone accounts for. A low R^2 at

economy (specs are enough) and a rising R^2 toward premium (brand fills the gap) would support the hypothesis. Because upper and premium bands have few vehicles — only 17 premium cars in the whole dataset — this is an indicative test, not a statistically conclusive one; a larger dataset with real transaction prices would be needed to quantify the brand premium reliably.

Question 4 — Professional Presentation (20 Marks)

Question 4 is delivered as the professional panel presentation (see the accompanying slide deck). The synthesis below mirrors the talk and is written for both technical and non-technical stakeholders at Mutuka Automotive.

4.1 Summarise the Key Findings

> **Question 4.1 · 4 Marks**
> In your presentation, briefly summarise the main findings from Questions 1 to 3 and explain why they matter for Mutuka Automotive: the business objective, important data-quality decisions, key EDA insights, vehicle segmentation or valuation grouping, and the main findings from the model comparison. Marks are awarded for interpretation and synthesis, not for merely showing code, tables, or graphs.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

Across Questions 1-3 a consistent picture emerges for Mutuka Automotive. **Objective:** build a specification-based first-level vehicle valuation and decision-support tool - the data describes specifications and price but not mileage, age, condition or history, so it values the *specification*, not the individual car. **Data quality:** the 205-row dataset was cleaned to 201 analysis-ready vehicles, recovering missing values from similar vehicles, with the target price complete throughout. **Exploratory analysis:** price is right-skewed and driven mainly by engine size, curb weight and horsepower (correlations about 0.8). **Segmentation:** vehicles fall into four economic tiers (economy, mid-range, upper, premium) and, on a separate axis, into use-case groups (commuter, family, performance) that cross-cut price. **Modelling:** a Random Forest predicts price with a mean error of about R34 000 (roughly 11%), explains 82% of price variation, and places a vehicle in the correct decision band 78% of the time. Together these show the tool is dependable for mainstream stock while pinpointing where it is not.

4.2 Define Acceptable Accuracy

> **Question 4.2 · 4 Marks**
> Based on the model performance, decide whether the final model is accurate enough to be used as a first-level vehicle valuation tool. Define an acceptable prediction-error range for this dataset and business context, such as whether an error of R1 500, R3 000, or R5 000 would be acceptable, or whether a percentage-based error threshold would be more appropriate. The threshold must be justified in relation to vehicle price, negotiation risk, profit implications, and customer trust, using appropriate metrics, validation results, and visualisations.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The model's typical error is about **R34 000 (roughly 11% of price)**, with 49% of vehicles valued within R20 000 and the correct decision band assigned 78% of the time. Because prices span from about R100 000 to R900 000, a single fixed-rand tolerance is unfair across that range, so we recommend a **percentage-based threshold: a prediction is acceptable if it is within +/-10% of the true price.** At Mutuka's median price (about R200 000) that is roughly +/-R20 000 - tight enough to protect the negotiation margin and customer trust, yet realistic for a specification-only model. On that basis the model is accurate enough to be a **first-level estimate** for the bulk of stock, but not accurate enough to set a final price unaided: it is the start of the negotiation, not the offer.

4.3 Practical Decision Rules

> **Question 4.3 · 4 Marks**
> Present practical decision rules for how Mutuka Automotive should use the model. For example, vehicles with reliable predictions may receive an automated preliminary estimate; vehicles with unusual specification combinations, rare categories, high predicted error, or values outside the training data range may be referred for manual review; and high-value or high-risk vehicles may require additional inspection before a final offer is made. The decision rules must follow logically from the analysis and model results.
>
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

The valuation translates into three practical rules for the floor:

- **Automated preliminary estimate** - vehicles in the economy, mid-range and upper bands whose specifications fall inside the training range receive an automatic first-level price, because the model is reliable there.
- **Manual review** - vehicles with unusual or rare specification combinations, a high predicted error, or values outside the training range are flagged for a person to check before a price is quoted.

• **Physical inspection** - premium and high-value vehicles always go for inspection before a final offer, because the model under-prices them and the money at risk per car is greatest.

The rules follow directly from the error analysis: trust the model where it is accurate, and escalate where it is not.

4.4 Business Consequences of Errors

> **Question 4.4 · 4 Marks**
> Explain the business consequences of the model's errors. Consider the effect of overpricing, underpricing, misclassification of valuation bands, or incorrect automated approval. You must connect the error analysis from Question 3 to practical risks such as financial loss, unfair offers, customer dissatisfaction, reputational damage, or poor negotiation decisions. Clearly identify the types of vehicles or situations where the model should not be trusted without human review, especially because the dataset does not include mileage, age, condition, service history, or accident history.
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

Errors carry real cost. **Under-pricing** loses margin and, in the premium band where the model is short by about **R92 000** on average, could badly undervalue a trade-in or drive a seller away. **Over-pricing** leaves stock sitting on the floor, tying up capital until it is marked down. **Band misclassification** - about 22% of cars land in an adjacent band - can mis-set pricing strategy and the effort put into a sale. The deepest limitation is the data itself: with no mileage, age, condition, service or accident history, two cars with identical specifications receive the same valuation even if one is worn out. The model must therefore never be trusted unaided on **premium, rare, high-performance or out-of-range vehicles** - which are exactly the high-stakes cases.

4.5 Final Recommendation

> **Question 4.5 · 4 Marks**
> End the presentation with a clear recommendation to Mutuka Automotive. State whether the model should be used for full automation, partial automation, decision support only, or further development before deployment. The recommendation must be supported by evidence from the EDA, segmentation or valuation grouping, model comparison, evaluation results, error analysis, and risk-based decision framework.
> *Verbatim from the ITSCA2 Project 2 brief (Matanga, 2026).*

Recommendation: adopt the tool for partial automation / decision support, not full automation. Auto-value the reliable majority - mainstream economy-to-upper vehicles within the known specification range - and route premium, rare and out-of-range cars to manual review and inspection. This captures most of the efficiency gain while containing risk where the model is weakest. Before going further, Mutuka should **collect the missing**

real-world data (mileage, age, condition and history) and log actual time-to-sell against the segments, then re-evaluate: those additions are what would turn a sound first-level estimator into a dependable full valuation system.

Conclusion

This project has demonstrated that a specification-based vehicle valuation and decision-support tool is achievable for Mutuka Automotive using the supplied dataset. A Random Forest regressor trained on 201 cleaned vehicles delivers a mean error of roughly R34 000 (11%) and correctly assigns the decision band 78% of the time — accurate enough to provide a consistent, defensible first-level estimate for the bulk of mainstream stock. The segmentation reveals that the fleet is skewed: premium vehicles represent a small share of units but a disproportionately large share of total fleet value, which directly motivates a tiered review policy. The recommended deployment is **partial automation**: auto-value the reliable majority (economy through upper band, in-range specifications) and route premium, rare and out-of-range vehicles to manual review and physical inspection before a final offer is made. The next meaningful improvement is collecting real-world data — mileage, age, condition and sales history — which would transform the current specification estimator into a dependable full valuation system and allow the review thresholds to be validated against actual sales outcomes.

AI Assistance Declaration

State clearly which AI tools (if any) were used, for what purpose, and how the output was verified, in line with the Eduvos assessment policy.

Generative AI tools were used as a productivity and learning aid throughout this project, in line with the Eduvos AI policy. The tools used were **Anthropic Claude (Opus 4.8)**, accessed through the Claude Code assistant, and **OpenAI ChatGPT (GPT-5.5)**.

AI assistance was used for the following:

• Building the supporting toolkit and document-generation pipeline (the `eduvos_itsca2` Python package, the branded Writer report and presentation templates, and the notebook-to-document compiler) used to assemble this report and the accompanying slide deck.

• Drafting, refactoring and debugging Python code for data loading, cleaning and visualisation.

• Spelling, grammar and clarity checks, and light editing of text that the author had already written.

• Discussing and structuring the analytical workflow and the layout of the deliverables.

AI was not used to generate, alter or fabricate the dataset or any analytical result. The data-quality decisions, the exploratory analysis, the modelling choices, and all interpretations, conclusions and recommendations are the author's own. All AI-assisted code was read, executed and verified by the author, and every figure and statistic in this report was reproduced from the supplied dataset. AI outputs were checked against the module materials (McKinney, 2017; VanderPlas, 2016) and the project brief before being relied upon.

References

All sources are listed alphabetically in Harvard (Eduvos) style. Every in-text citation must have a matching entry here, and vice versa.

Kaggle, 2018. *Car Price Prediction Dataset.* [online] Available at: <https://www.kaggle.com> [Accessed 20 June 2026].

McKinney, W., 2017. *Python for Data Analysis.* 2nd ed. Sebastopol: O'Reilly Media.

VanderPlas, J., 2016. *Python Data Science Handbook.* Sebastopol: O'Reilly Media.

Anthropic, 2026. Claude (Opus 4.8) [Large language model]. Available at: https://claude.ai [Accessed 20 June 2026].

OpenAI, 2026. ChatGPT (GPT-5.5) [Large language model]. Available at: https://chat.openai.com [Accessed 20 June 2026].