University of Stirling – Ras Al Khaimah

Machine Learning Assignment

CSCU9M5

Author: Paula Vianca Cunanan

Table of Contents

Business Understanding

This project applies the CRISP-DM methodology to build a machine learning classifier that predicts whether a store will be profitable ("Good") or not ("Bad"). The client, a chain store owner, aims to use this model to enhance investment decisions by analysing historical data containing attributes such as location, demographics, competition metrics, and performance. The model helps allocate resources efficiently, reduce risks associated with unprofitable stores, and provide insights into factors driving profitability, enabling operational improvements.

Machine learning is ideal for this task due to the complexity of relationships between variables and the need to generalise predictions to new stores. Its ability to uncover non-linear patterns and interactions makes it valuable for profitability prediction. The CRISP-DM framework ensures a systematic approach from understanding business goals to deploying the model.

Terminology

- **Model**: A machine learning classifier predicting a store's profitability based on its attributes.
- **Variable**: A measurable feature or column in the dataset.
- **Target Variable**: The outcome variable, Performance, indicating profitability ("Good" or "Bad").
- **Accuracy**: Proportion of correctly classified instances in the dataset.[2]
- **Precision**: Proportion of true positives among predicted positives, focusing on minimising false positives. [2]
- **Recall**: Proportion of actual positives correctly identified, focusing on minimising false negatives. [2]
- **F1 Score**: Harmonic mean of precision and recall. [2]
- **AUC-ROC**: Measures the model's ability to distinguish between Good and Bad classes. [2]
- **MCC**: A balanced metric that evaluates classification performance, considering all elements of the confusion matrix. [2]

Business Impact

The model directly impacts investment decisions, with significant consequences for errors:

- **False Positive (FP)**: Predicting profitability when the store is not, leading to wasted resources and financial losses.

- **False Negative (FN)**: Missing profitable stores, causing lost opportunities and stunted growth.

Minimising these errors ensures the model's effectiveness in supporting data-driven decisions.

Conclusion

This binary classification task helps the client make informed investment decisions by uncovering patterns in complex variable relationships. The final model not only predicts profitability but also identifies key drivers, providing actionable insights that align with the business's long-term goals.

Data Understanding

Overview of the Data

The dataset consists of **136 instances**, **17 features**, **2 meta attributes** (*Town* and *Manager*), and a **target variable**, *Performance*. The dataset is clean, with no missing values. The features are a mix of **4 categorical variables** (*Car Park*, *Location*, *Country*, *Performance*) and **13 numeric variables** (*Staff*, *Floor Space*, *Demographic Score*, etc.). The target variable, *Performance*, categorises stores as either *Good* (profitable) or *Bad* (not profitable). Meta attributes are excluded since they do not provide direct predictive value.

Variable Types

1. Key Numeric Variables (Continuous):

- **Staff**:
    - **Type**: Numeric (Continuous).
    - **Description**: Represents the number of employees in a store.
    - **Relevance**: Included due to its operational significance; higher staff numbers may indicate greater capacity for customer service and store operations.
- **Floor Space**:
    - **Type**: Numeric (Continuous).
    - **Description**: Measures the size of the store in square units.
    - **Relevance**: Included as larger floor spaces typically allow for greater inventory and customer capacity.
- **Demographic Score**:
    - **Type**: Numeric (Continuous).

- **Description**: Reflects the attractiveness of the store's surrounding area based on factors like income levels and consumer demographics.
- **Relevance**: Included as it serves as a proxy for market potential and marketing effectiveness.

- **Clearance Space**:
  - **Type**: Numeric (Continuous).
  - **Description**: Indicates the area allocated for clearance sales.
  - **Relevance**: Included due to its potential connection with inventory management and profitability.
- **Competition Score**:
  - **Type**: Numeric (Continuous).
  - **Description**: A summary metric of competitive intensity in the store's area not just the number of competitors but also their characteristics. A higher Competition Score may indicate a more challenging or competitive environment.
  - **Relevance**: Included as it directly impacts a store's market positioning.
- **Competition Number**:
  - **Type**: Numeric (Continuous).
  - **Description**: Represents the number of competitors in the store's vicinity.
  - **Relevance**: Included, though secondary to *Competition Score*, as it adds additional context to competitive pressure. This feature represents the number of competing stores present around the store. It might be that, if it has high Competition Number there are lots of other stores nearby that offers similar products and services to the store. This can impact the store's ability to attract customers, more competition might mean fewer customer per store.
- **Store Age**:
  - **Type**: Numeric (Continuous).
  - **Description**: Indicates the age of the store in years.
  - **Relevance**: Excluded due to limited impact observed in the feature ranking and statistical analysis.

3. Categorical Variables:

- **Location**:
  - **Type**: Nominal.
  - **Description**: Classifies the type of store location (e.g., *High Street*, *Retail Park*).

    - **Relevance**: Included due to its significant influence on customer footfall and profitability, as shown in feature ranking.

- **Car Park**:
    - **Type**: Nominal.
    - **Description**: Indicates whether the store has a car park (Yes/No).
    - **Relevance**: Included as accessibility is a critical factor in customer convenience and store success.
- **Window**:
    - **Type**: Nominal.
    - **Description**: Reflects the store's display area and potential for visual merchandising.
    - **Relevance**: Included due to its moderate correlation with customer attraction and conversion rates.
- **Country**:
    - **Type**: Nominal.
    - **Description**: Specifies the store's country (e.g., UK, France).
    - **Relevance**: Excluded due to low variance, as the majority of stores are in the UK.

4. Population Metrics:

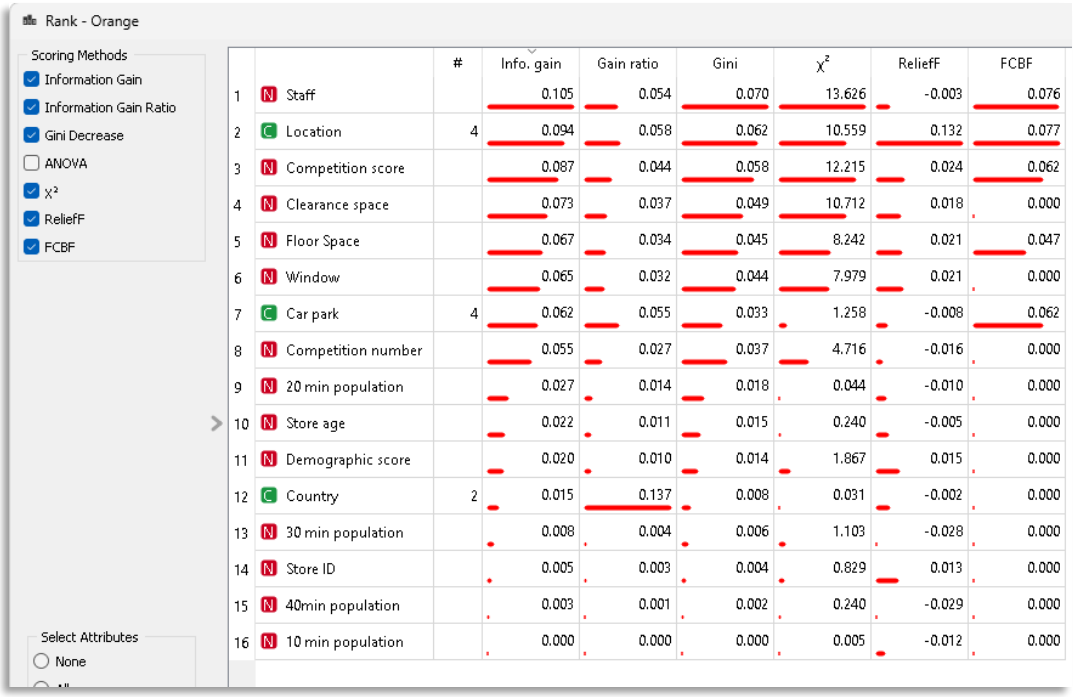
- **10 min Population**, **20 min Population**, **30 min Population**, **40 min Population**:
    - **Type**: Numeric (Continuous).
    - **Description**: Represent populations within different travel radii from the store.
    - **Relevance**: Excluded *individually* due to low predictive power, as shown by the Rank widget. These metrics do not significantly improve model performance when used independently. Combining population would be better.

5. Identifiers:

- **Store ID**:
    - **Type**: Numeric (Discrete).
    - **Description**: A unique identifier for each store.
    - **Relevance**: Excluded as it has no predictive value for profitability.

Using the Rank Widget for Feature Selection

Rank widget in Orange was used to assess the importance of features for predicting store performance (Good or Bad). By applying statistical methods such as Information Gain, Gini Index, and Chi-squared tests, the widget identified features like *Staff*, *Location*, and *Competition Score* as the most relevant for the target variable [7].



		#	Info. gain	Gain ratio	Gini	χ^2	ReliefF	FCBF
1	Staff		0.105	0.054	0.070	13.626	-0.003	0.076
2	Location	4	0.094	0.058	0.062	10.559	0.132	0.077
3	Competition score		0.087	0.044	0.058	12.215	0.024	0.062
4	Clearance space		0.073	0.037	0.049	10.712	0.018	0.000
5	Floor Space		0.067	0.034	0.045	8.242	0.021	0.047
6	Window		0.065	0.032	0.044	7.979	0.021	0.000
7	Car park	4	0.062	0.055	0.033	1.258	-0.008	0.062
8	Competition number		0.055	0.027	0.037	4.716	-0.016	0.000
9	20 min population		0.027	0.014	0.018	0.044	-0.010	0.000
10	Store age		0.022	0.011	0.015	0.240	-0.005	0.000
11	Demographic score		0.020	0.010	0.014	1.867	0.015	0.000
12	Country	2	0.015	0.137	0.008	0.031	-0.002	0.000
13	30 min population		0.008	0.004	0.006	1.103	-0.028	0.000
14	Store ID		0.005	0.003	0.004	0.829	0.013	0.000
15	40min population		0.003	0.001	0.002	0.240	-0.029	0.000
16	10 min population		0.000	0.000	0.000	0.005	-0.012	0.000

FIGURE 1 RANK WIDGET

Inputs and Outputs

- **Inputs (Features)**:
    - The retained input features for model building include:
        - **Numeric Features**: *Staff*, *Floor Space*, *Demographic Score*, *Clearance Space*, *Competition Score*, *Population Metrics*, and *Competition Number*.
        - **Categorical Features**: *Location*, *Car Park*, *Window*.
- **Output (Target)**: Performance

Data Preparation

Unique Identifier and meta attributes will be necessary because there is no predictive value. Adding these features would just add unnecessary complexity without improving model performance. For the Country feature, given that the dataset primarily focuses on the UK, Country feature has low variance and minimal contribution to predicting performance. Based on the dataset, there are only 2 stores in the France and they both perform badly. Reducing irrelevant features avoid overfitting, reduce noise, simplifies the model interpretation, and improve model efficiency without sacrificing accuracy. Even when Population Metrics individually indicate a low impact, combining them would make more sense as when there are more people, the store has better profit. Therefore, based on the relevant feature, **Select Column** widget was used to ignore the irrelevant features.

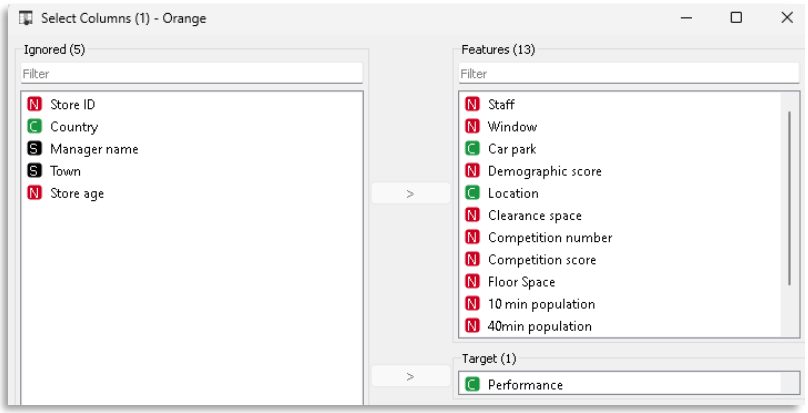


FIGURE 2 SELECT COLUMN WIDGET

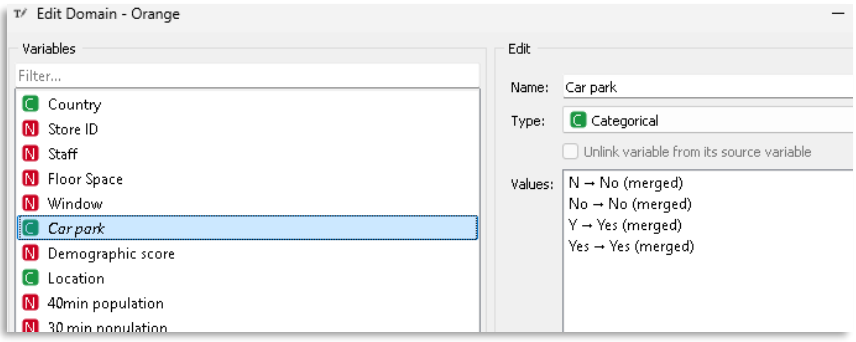


FIGURE 3 EDIT DOMAIN WIDGET

Furthermore, during the data preparation process, standardising the **Car Park** variable using Edit Domain widget was done. The original values "Y" and "N" were merged into the more descriptive

categories "Yes" and "No," respectively. This change was made to enhance the readability of the data and maintain consistency in categorical variables.

Handle Missing Data

Based on Feature Statistics and Data Table, the dataset does not have any missing data. The impute widget was considered but not applied due to the absence of missing data in the dataset.

Handling Outlier and Visualisation

Outliers can distort scaling, reduce model accuracy, and affect relationships between features therefore, **Scatter Plot** widget was used to visually identify any outliers in the numeric features. There are questionable inputs in the Staff Feature such as the -2, 300, and 600. -2 is not logically correct as number of Staff in a store and the hundred employees does not also correlate with the floor size. Therefore, these were removed interpreted as wrong input. Lastly, the widget **Outlier** was used for automatically removing the outliers.

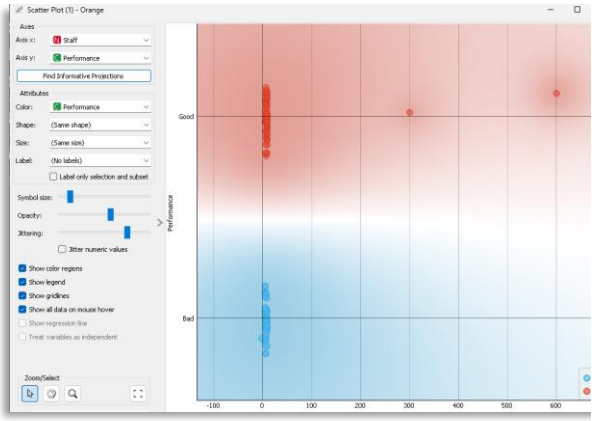


FIGURE 4: STAFF WITH OUTLIERS

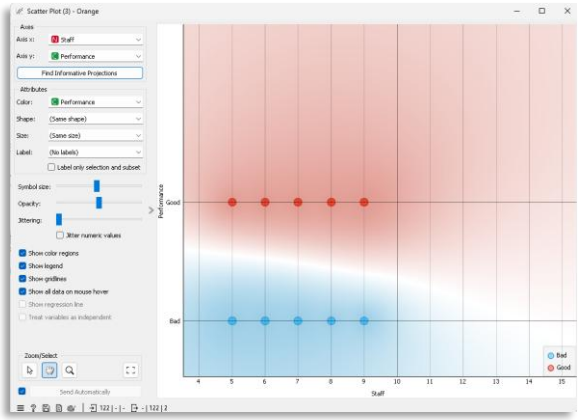
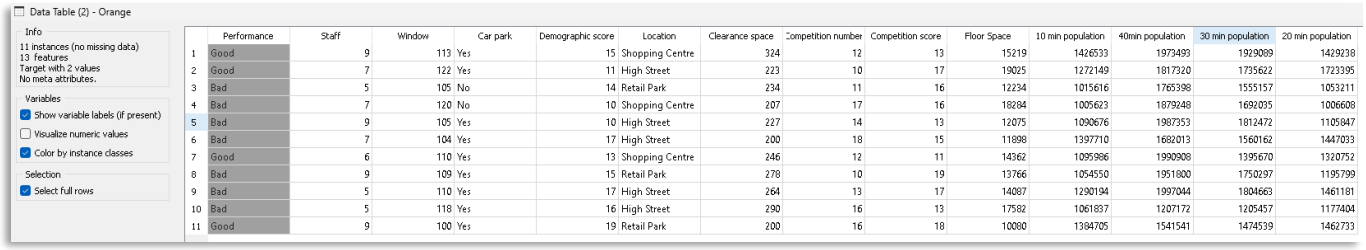


FIGURE 5 STAFF WITHOUT OUTLIERS

The **Staff** relationship with Performance suggests that staffing levels may play a role in store performance. A higher number of staff likely supports better customer service, operational efficiency, and overall store management, which can contribute to profitability. On *figure 5*, there are no more visible outliers in the Staff feature after removing the extreme values (-2, 300, 600). The distribution now appears cleaner, with the values for both "Good" and "Bad" categories falling within reasonable ranges. Table below are the outliers detected using the **Outlier** widget.



	Performance	Staff	Window	Car park	Demographic score	Location	Clearance space	Competition number	Competition score	Floor Space	10 min population	40min population	30 min population	20 min population
1	Good	9	113	Yes	15	Shopping Centre	324	12	13	15219	1426533	1973493	1929089	1429238
2	Good	7	122	Yes	11	High Street	223	10	17	19025	1272149	1817320	1735622	1723395
3	Bad	5	105	No	14	Retail Park	234	11	16	12234	1015616	1765398	1555157	1053211
4	Bad	7	120	No	10	Shopping Centre	207	17	16	18284	1005623	1879248	1692035	1006608
5	Bad	9	105	Yes	10	High Street	227	14	13	12075	1090676	1987353	1812472	1105847
6	Bad	7	104	Yes	17	High Street	200	18	15	11898	1397710	1682013	1560162	1447033
7	Good	6	110	Yes	13	Shopping Centre	246	12	11	14362	1095986	1990908	1395670	1320752
8	Bad	9	109	Yes	15	Retail Park	278	10	19	13766	1054550	1951800	1750297	1195799
9	Bad	5	110	Yes	17	High Street	264	13	17	14087	1290194	1997044	1804663	1461181
10	Bad	5	118	Yes	16	High Street	290	16	13	17582	1061837	1207172	1205457	1177404
11	Good	9	100	Yes	19	Retail Park	200	16	18	10080	1384705	1541541	1474539	1462733

FIGURE 6 DETECTED OUTLIERS FROM OUTLIER WIDGET

Feature Engineering

Feature engineering plays a critical role in improving the predictive power and interpretability of machine learning models. By aggregating and deriving meaningful features, the model can better capture the underlying relationships in the data. For instance, calculating the PopulationTotal by weighting populations from different time zones (e.g., 40-minute, 30-minute, etc.) provides a single, interpretable metric that reflects the potential customer base. Closer zones, such as 10 minutes, are weighted higher to indicate a higher likelihood of customers visiting the store. This aggregation simplifies the relationships that the model needs to learn and directly aligns with business objectives by quantifying customer density in a meaningful way.

Similarly, the Floor/Window ratio represents store layout efficiency or customer attraction capability. A larger window relative to floor space could indicate better visibility and customer appeal, while a smaller ratio might suggest less emphasis on external attractiveness. This derived feature reduces multicollinearity that might arise when Floor_Space and Window are treated independently, enabling the model to make clearer and more interpretable decisions. Overall, feature engineering ensures that the model captures relevant insights, simplifies relationships, and aligns the data with real-world contexts, ultimately improving both performance and interpretability. The screenshot below shows that **Formula** widget was used for combing the features and making new feature for modelling. **Floorwindow** and **PopulationTotal** was added as a new feature.

FIGURE 7 FORMULA WIDGET FOR FEATURE ENGINEERING

Data Scaling and Normalisation

To ensure consistent feature scaling and improve model performance, normalisation was applied to specific continuous numeric features [1]. The features requiring normalisation are the Numerical features as these variables exhibit significant variability in their ranges. Normalisation ensures that these features are scaled to a uniform range, preventing larger magnitude features from dominating smaller-scale features during the modelling process [1]. This step is critical for algorithms like SVM and Gradient Boosting which are sensitive to differences in feature magnitudes. By normalising these features, the model achieves better balance, avoids feature dominance, and ensures faster and more stable convergence during training [1]. The normalisation to the numeric variables was done with **Standardize to μ=0, σ^2=1.** This step is necessary for models that are sensitive to feature scaling. This will improve the performance metrics for models such as the SVM and Gradient Boosting. Categorical variables (e.g., Location, Car Park) and target labels were excluded from normalisation as they do not require scaling.

Below is the distribution of Floor_Space and the Window individually data before using the **Preprocess** widget for normalizing the data.

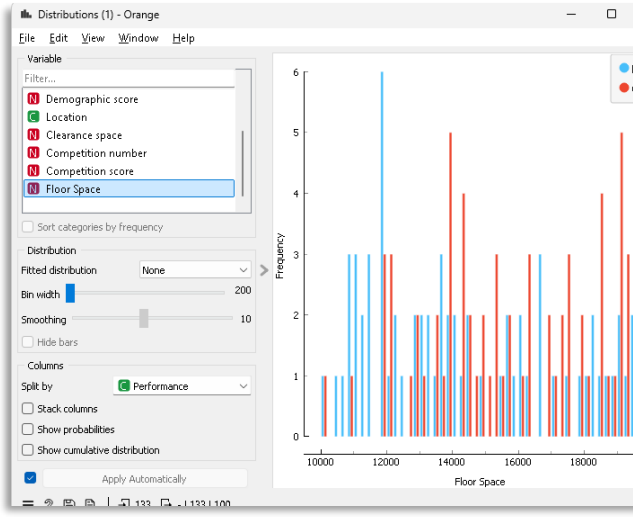


FIGURE 8 FLOOR_SPACE BEFORE NORMALIZING

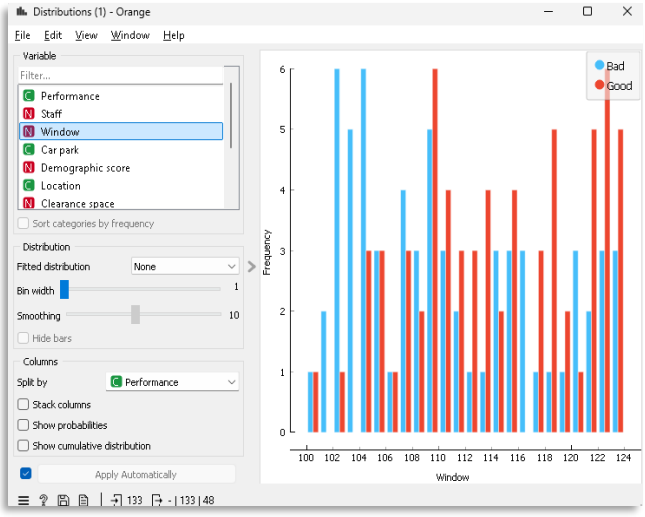


FIGURE 9 WINDOW BEFORE NORMALIZING

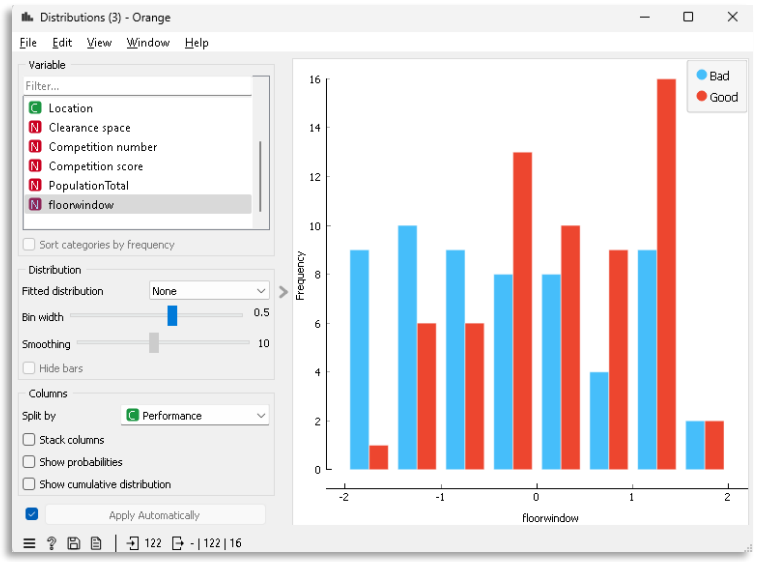


FIGURE 10 FLOORWINDOW NORMALIZED

After normalising and using the feature engineering, the distribution now looks like the table below. It is better for interpretation and shows more balanced data.

Modelling

After the feature engineering and normalising, the **Select Column** was used again for removing the individual population metrics, floor_space, and window. Then, PopulationTotal and floorwindow was added.

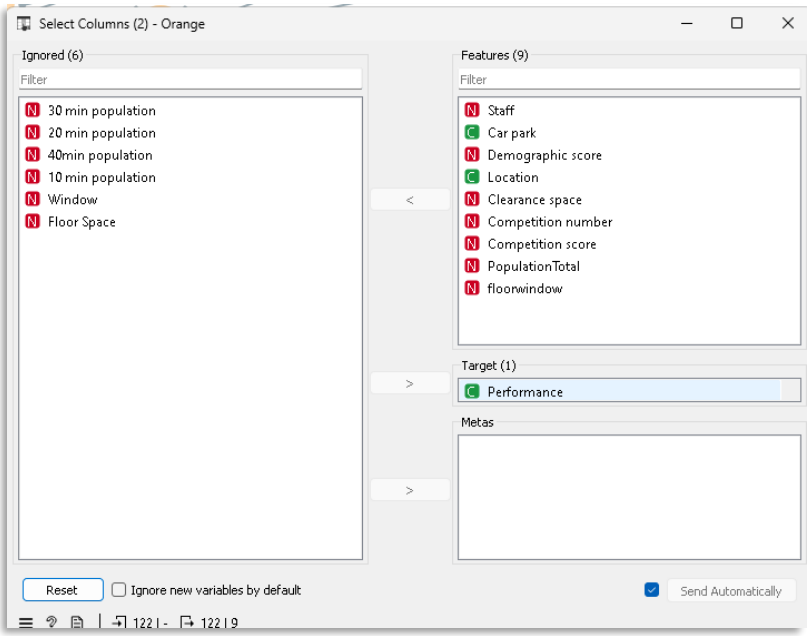


FIGURE 11 SELECT COLUMN AFTER PREPROCESSING

Data Testing and Split

For the modelling process 70% of the data was used to train the models and 30% was reserved for testing and evaluation.

Choosing Model

Three machine learning models were selected for this task:

1. **Logistic Regression**: A simple and interpretable linear model, ideal for binary classification tasks like predicting store performance ("Good" or "Bad"). It works well with balanced datasets and provides insights into feature influence [4].
2. **Support Vector Machine (SVM):** Effective for both linear and non-linear relationships [5], SVM is robust with small datasets like this (136 instances). Its ability to optimise class separation makes it a strong candidate.
3. **Gradient Boosting:** Known for modelling complex, non-linear relationships, Gradient Boosting combines weak learners to enhance prediction accuracy [3]. Its ability to capture feature interactions, such as demographic and competition scores, makes it suitable for this dataset.

Hyperparameter Exploration

To optimise model performance, hyperparameter tuning was applied to all selected algorithms: Logistic Regression, Support Vector Machine (SVM), and Gradient Boosting. A Grid Search approach with 10-fold cross-validation with stratification was used to evaluate a range of parameter combinations and select the best-performing configuration for each model.

Logistic Regression

- **Regularisation Type**:
    - **Result of Regularization Type**: L2 ridge was chosen in this scenario for Logistic Regression because it demonstrated slightly better performance across key evaluation metrics compared to L1 Lasso.

Regularization Type	AUC	CA	F1	Precision	Recall	MCC
L1, C = 1	0.914	0.814	0.814	0.814	0.814	0.627
L2, C=1	0.9	0.814	0.814	0.814	0.814	0.627

- **Regularisation Strength (C)**: A range of values for C was tested to balance model complexity and generalisation. The optimal configuration was achieved with C = 1.0, providing a suitable trade-off between flexibility and robustness.

Support Vector Machine (SVM)

- **Kernel**: A linear kernel was selected based on the dataset's characteristics and the need for interpretability, avoiding unnecessary complexity from non-linear kernels. Other kernels like Polynomial, Sigmoid, and RBF was considered but linear shows better performance across the evaluation metric. The best kernels were Linear and RBF which is compared on the table below.

Kernel	AUC	CA	F1	Precision	Recall	MCC
Linear	0.902	0.826	0.826	0.826	0.826	0.651
RBF	0.811	0.744	0.743	0.745	0.744	0.487

- **Cost (C)**: C = 1.0 provided the best balance between maximising the margin and tolerating misclassifications.

- **Iteration Limit**: Adjusted numerical tolerance and iteration limits to ensure stable and efficient convergence during training. Iteration Limit = 100

Extreme Gradient Boosting (xgboost)

- **Method:** Several Methods were considered for Gradient Boosting such as Extreme Gradient Boosting (xgboost) and scikit-learn. The method that performed the best is the xgboost.

Method	AUC	CA	F1	Precision	Recall	MCC
Xgboost	0.879	0.779	0.778	0.780	0.779	0.557
Scikit-learn	0.663	0.628	0.623	0.629	0.628	0.252

- **Number of Trees**: Configurations with varying numbers of trees were tested, with 100 trees providing the optimal balance between model capacity and computational efficiency.

- **Learning Rate**: Set to 0.3 to enable gradual model updates and reduce the risk of overfitting.

- **Maximum Tree Depth**: Limited to a depth of 6 to control overfitting while maintaining model interpretability.

Summary Result

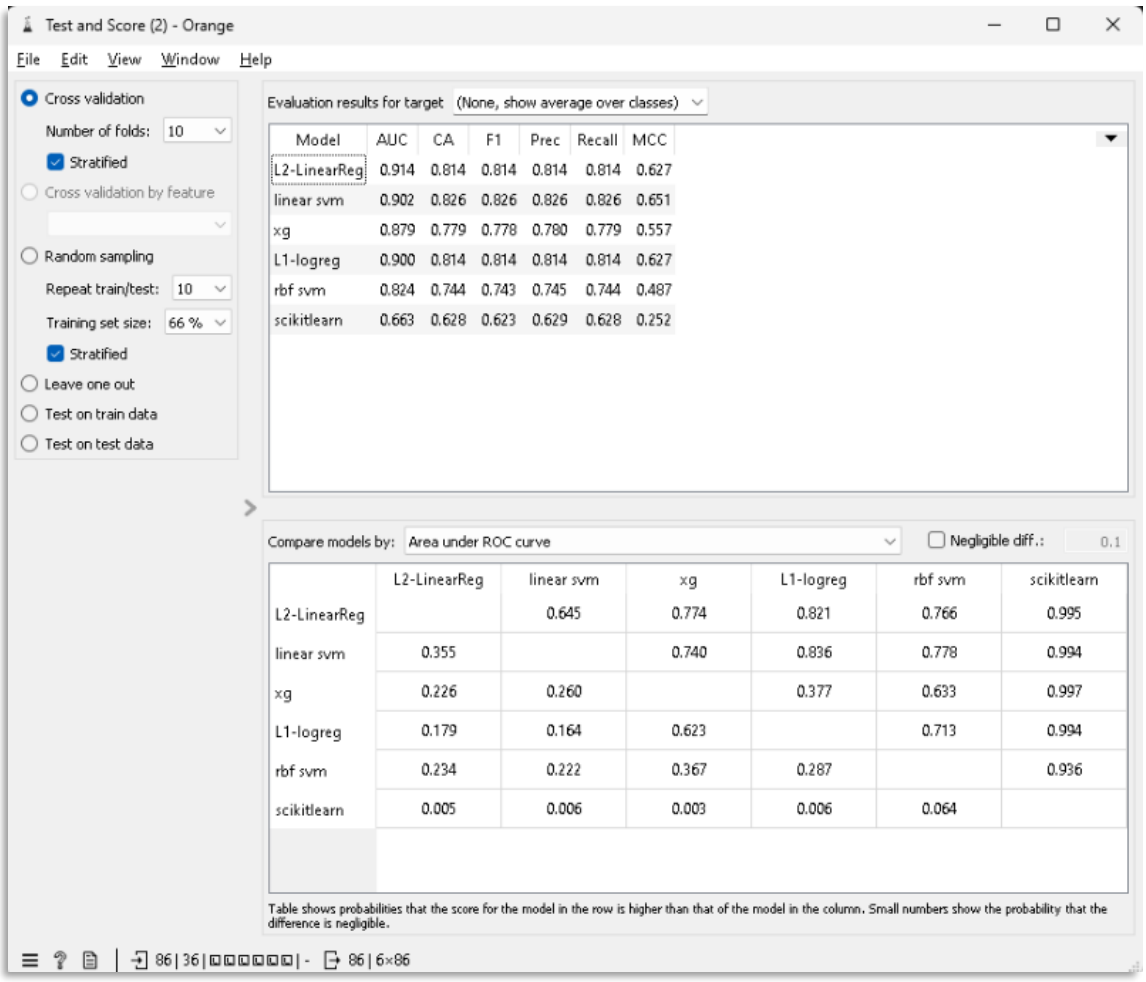


FIGURE 12 EVALUATION METRICS TABLE

The evaluation compares the performance of three machine learning models—SVM, Logistic Regression, and Gradient Boosting—on key metrics: AUC, CA, F1 score, Precision, Recall, and MCC. These metrics provide insights into the models' ability to classify the target variable accurately and consistently. Using 10-fold cross-validation ensured robust and unbiased performance evaluation. Based on the evaluation metrics, **L2-Regularised Logistic Regression (L2-LinearReg)** with **C = 1** is the best-performing model with the highest AUC (0.914), CA (0.814), F1 score (0.814), Precision (0.814), Recall (0.814), and MCC (0.645). It effectively balances False Positives and False Negatives, aligning with the business need to minimise misallocated resources and missed opportunities. Logistic Regression is also highly interpretable, making it ideal for understanding the influence of variables on profitability. Its performance and simplicity suit the dataset's relatively linear structure and small size (136 instances), making it the most suitable choice for predicting store performance.

Final Model Evaluation

The final model, **Logistic Regression with L2 regularisation (Ridge)** and C=1, was trained on the clean dataset to optimise predictive performance. Its generalisation was validated using test data and 10-fold stratified cross-validation.

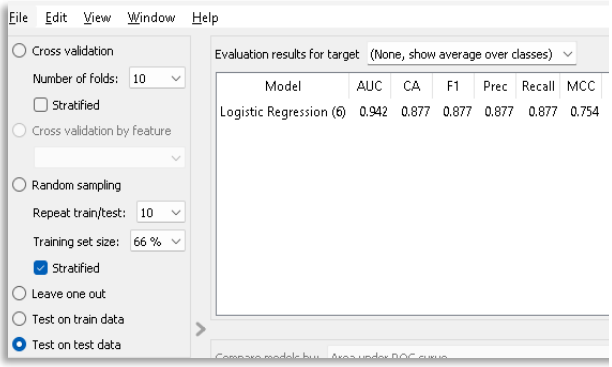


FIGURE 13 TEST ON TEST DATA

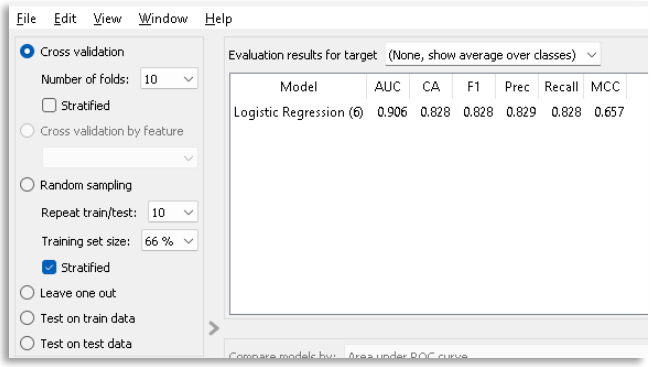


FIGURE 14 TEST USING CROSS VALIDATION

Cross-validation provided robust and unbiased evaluation by testing the model across multiple splits, while test data offered a reliable estimate of real-world performance. The model demonstrated strong predictive capabilities, with consistent performance metrics across methods. This confirms that the Logistic Regression model is robust, stable, and well-suited for deployment, making it a reliable choice for the task. The testing on data on final model's balanced precision (0.877) and recall (0.877) indicate its reliability for real-world deployment, with strong performance across all areas of the data.

Confusion Matrix of Final Model: Logistic Regression

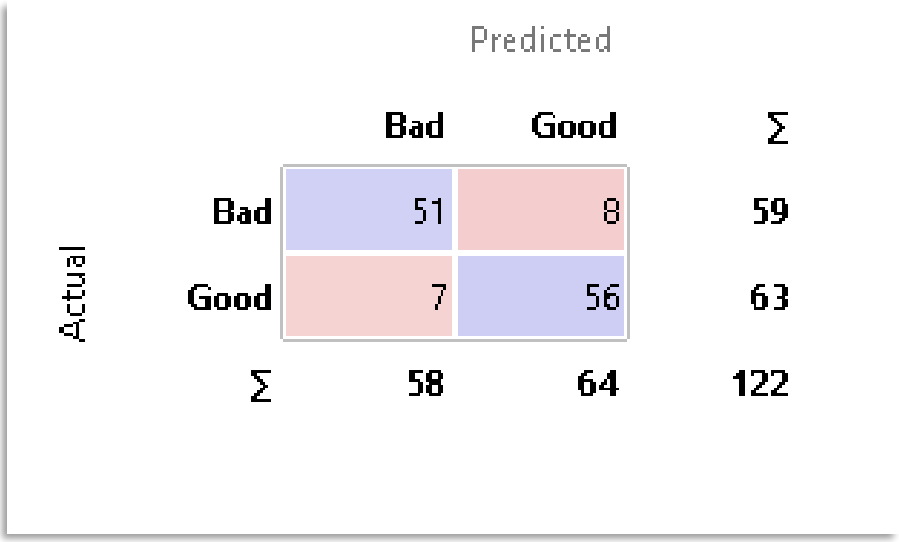


FIGURE 15 CONFUSION MATRIX OF FINAL MODEL

Results and Errors

The final showed strong performance overall but revealed certain limitations.

1. Confusion Matrix Results:

    - **True Positives (TP):** Correctly identified 59 "Good" stores.
    - **True Negatives (TN):** Correctly identified 51 "Bad" stores.
    - **False Positives (FP):** Misclassified 8 "Bad" stores as "Good," potentially leading to wasted investments.
    - **False Negatives (FN):** Misclassified 7 "Good" stores as "Bad," causing missed profitable opportunities.

2. Model Challenges:

    - Difficulty distinguishing stores near decision boundaries, such as those with average competition scores and demographic scores, where class overlap is evident.
    - Features like staff size, clearance space, and population total show overlapping distributions for "Good" and "Bad" stores, contributing to misclassifications.

3. Performance Trends:

    - The model performs well on distinct feature ranges (e.g., high or low competition scores) but struggles with intermediate values where "Good" and "Bad" stores are similar.

This analysis highlights that while the model is robust, addressing overlapping feature distributions through advanced feature engineering or additional data may further improve classification accuracy. Distribution plots were utilised to analyse the model's errors by examining the overlap between correctly and incorrectly classified instances across key features. These plots provided insights into areas where the model struggles, particularly for stores near decision boundaries, such as those with average competition scores and demographic scores.

Conclusion

The Logistic Regression model with L2 regularisation (C=1) demonstrated strong and consistent performance, achieving an AUC of 0.942 and balanced metrics across precision, recall, and F1 score. The model effectively distinguishes between profitable ("Good") and unprofitable ("Bad") stores, with a low error rate indicated by minimal false positives and negatives. Error analysis using distribution plots highlighted challenges in classifying instances near decision boundaries, such as stores with average competition scores or demographic scores, suggesting potential areas for refinement.

Overall, the model is robust and interpretable, making it a reliable tool for the client to make informed investment decisions. Recommendations include further fine-tuning on features like competition and demographic scores and exploring alternative models to address borderline cases for improved performance.

References

1. Orange Data Mining, "Data Preparation for Machine Learning," Accessed: Nov. 22, 2024. [Online]. Available: https://orangedatamining.com/blog/data-preparation-for-machine-learning/

2. Orange3 Documentation, "Test and Score," Accessed: Nov. 23, 2024. [Online]. Available: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/evaluate/testandscore.html

3. Orange Data Mining, "Gradient Boosting," Accessed: Nov. 22, 2024. [Online]. Available: https://orangedatamining.com/widget-catalog/model/gradientboosting/

4. Orange Data Mining, "Logistic Regression," Accessed: Nov. 22, 2024. [Online]. Available: https://orangedatamining.com/widget-catalog/model/logisticregression/

5. Orange Data Mining, "Support Vector Machine (SVM)," Accessed: Nov. 22, 2024. [Online]. Available: https://orangedatamining.com/widget-catalog/model/svm/

6. Orange3 Documentation, "Confusion Matrix," Accessed: Nov. 24, 2024. [Online]. Available: https://orange3.readthedocs.io/projects/orange-visual-programming/en/latest/widgets/evaluate/confusionmatrix.html

7. Orange Data Mining, "Rank - Orange Widgets Documentation," *Orange3 Documentation*, 2024. [Online]. Available: https://orange3.readthedocs.io/projects/orange-visual-programming/en/master/widgets/data/rank.html. [Accessed: 22-Nov-2024].