Personalised Satisfaction Prediction and Mismatch Detection Using Machine Learning

FINAL REPORT

Higher Diploma in Science in Data Analytics

Bouchra KADOURI:2025

Contents

BOUCHRA KADOURI

22 August 2025

Introduction

Social media platforms and data analytics are turning into powerful tools for gathering and analysing marketing data. All businesses and organisations strive to present their best image to the public, as customer satisfaction is crucial to an organisation's success. Companies in every industry pause to consider what their clients want and utilise customer feedback to refine their business processes. Satisfied clients typically improve customer retention, loyalty, patronage, and favourable brand advertising. However, unhappy clients will stop purchasing products and services, provide negative feedback, and adversely affect the company's profits and image.

Effective segmentation helps businesses understand key drivers of satisfaction that enable informed decision-making to improve customer relations. Customer information systems, along with modern software applications, allow the processing of sophisticated customer datasets of age, income level, product or service ratings, purchase transactions, and overall satisfaction levels. Through these methods, businesses are now able to understand their customers better, improve customer satisfaction, and foster strong customer loyalty.

With the ability to systematically identify unsatisfied customers, dissatisfaction with a brand or service can be remedied. The project proposes a machine learning model that calculates the expected satisfaction score using the user's profile and actions for each customer. The customer's predicted satisfaction score is then side by side with their reported satisfaction to identify personalised mismatches. These discrepancies can identify concealed dissatisfaction and targeted areas for improvement.

Business Understanding

Most businesses traditionally assess customer satisfaction by comparing an individual's score to the average satisfaction of a group, such as all gold customers or people in a certain age range. This group-average approach assumes that everyone within the group should have similar satisfaction levels, which overlooks the personal differences that can significantly affect how satisfied someone feels. As a result, unique cases of dissatisfaction may be missed if they don't stand out against the group's average.

In contrast, this project adopts an individualised, machine learning, driven approach. Instead of relying on group averages, we use customer-specific data such as age, loyalty level, and usage patterns to predict what each customer’s satisfaction score should be. This prediction creates a personalised expectation for every customer, considering the factors most relevant to them. We then compare this predicted satisfaction to the customer’s actual reported satisfaction. If there’s a significant gap, especially when the actual satisfaction is lower than predicted, the case is flagged as a mismatch. This method helps uncover hidden problems and enables the business to target interventions more effectively, ensuring that each customer’s experience is evaluated in the context of their unique profile rather than a broad group average

Project Scope

This project aims to develop a machine learning–based system that predicts each customer's expected satisfaction score based on their profile and behaviour, including factors like age, income, loyalty level, and service quality. It then compares this predicted satisfaction score with the customer’s actual reported satisfaction to identify personalised mismatches.

Unlike traditional methods that depend on group averages, this approach offers a more accurate, customer-specific analysis. By identifying cases where expectations are not being met, the system enables businesses to uncover hidden dissatisfaction and respond more effectively.

Project Objectives

The main objectives of this project are to:

- Build a machine learning model that predicts each customer’s satisfaction score based on data such as age, income, product/service quality, and purchase frequency.
- Compare the predicted satisfaction score with the customer’s actual score to identify individuals who are less or more satisfied than expected.
- Flag customers who show a significant mismatch between predicted and actual satisfaction levels.
- Categorise these mismatches into actionable groups (e.g., “Under-Satisfied High Spender”) to help businesses prioritise interventions.

- Provide data-driven insights that companies can use to improve customer experience, increase retention, and drive loyalty.

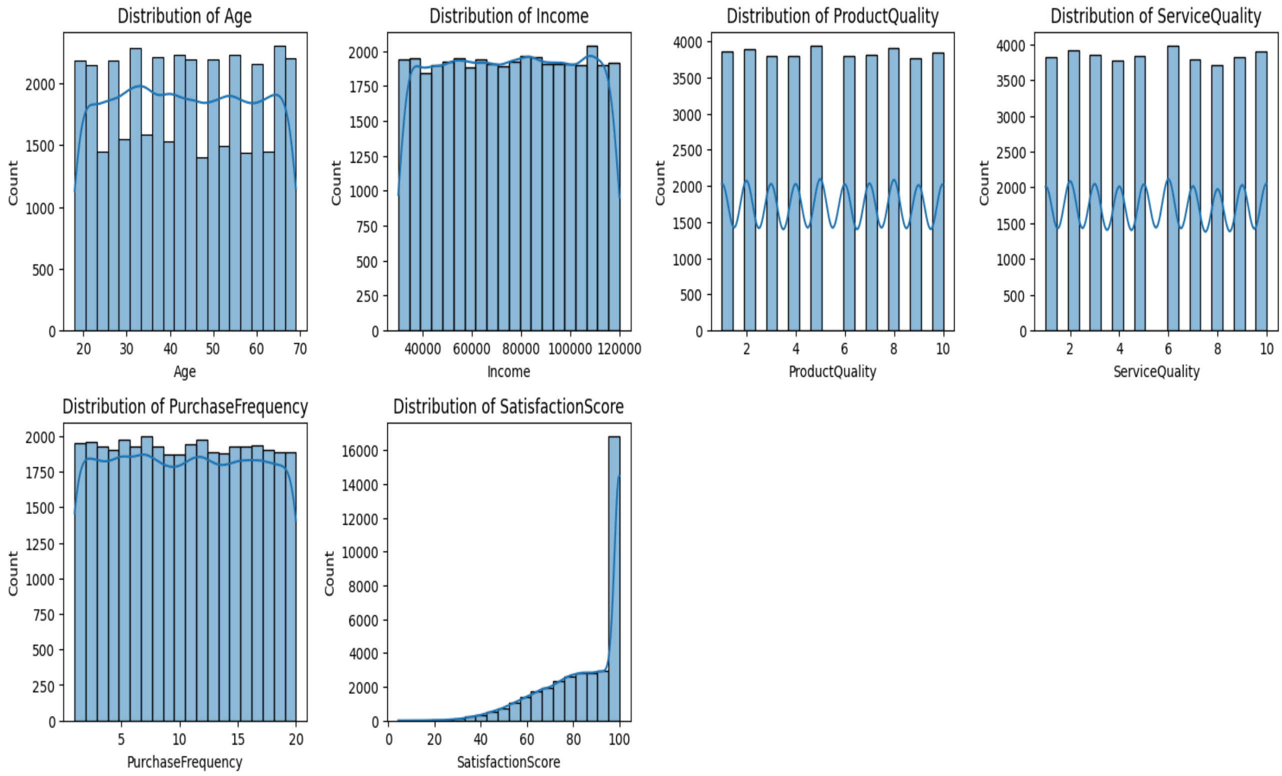
Data Understanding

https://www.kaggle.com/datasets/jahnavipaliwal/customer-feedback-and-satisfaction

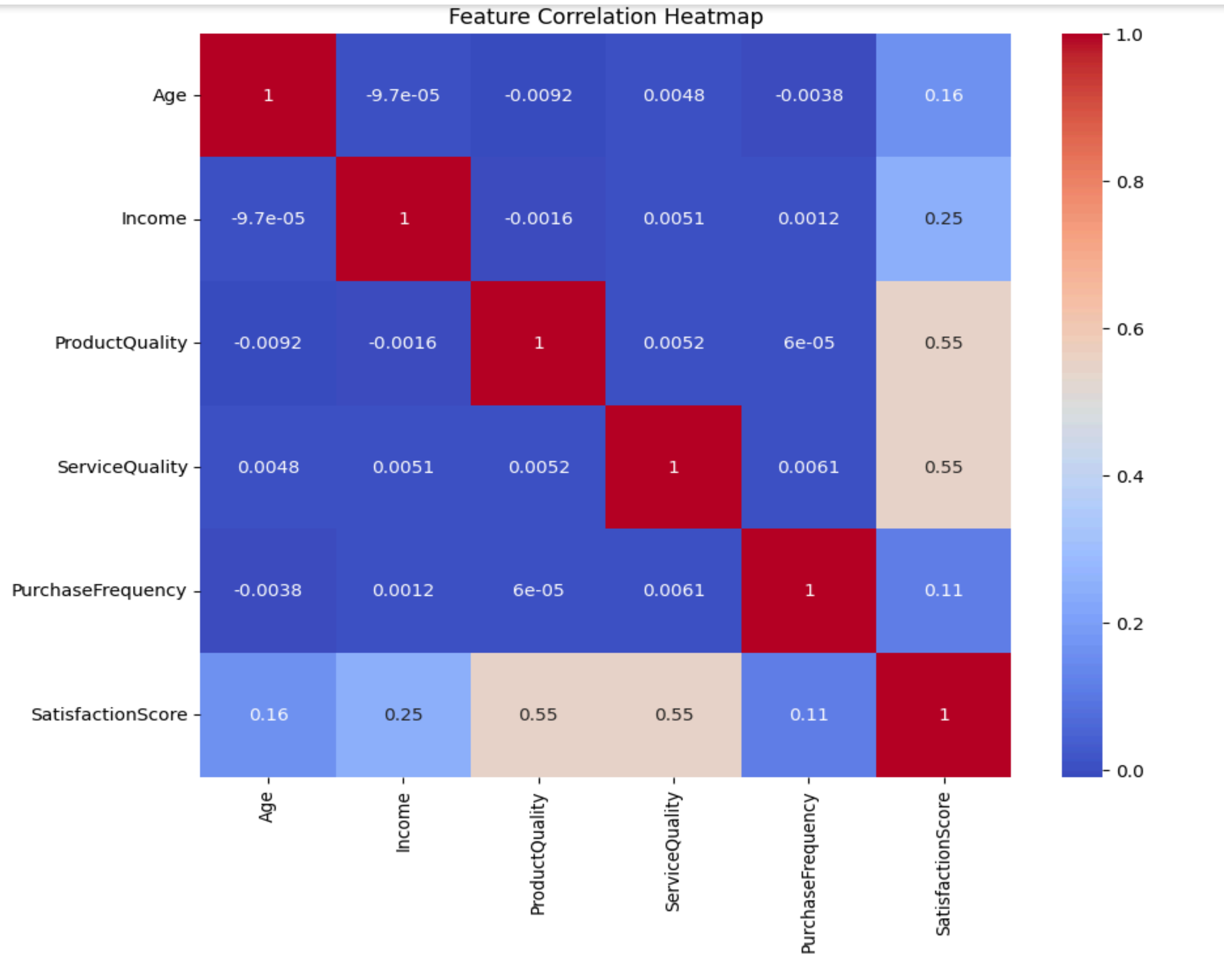
In the data understanding phase of this project, we start by thoroughly exploring the customer feedback dataset to gain insights into its structure and content. Using Python, we load the data and examine key columns, including age, loyalty level, satisfaction score, usage, and other relevant features, by displaying the first few rows and generating summary statistics.

	Age	Gender	Country	Income	ProductQuality	ServiceQuality	PurchaseFrequency	FeedbackScore	LoyaltyLevel	SatisfactionScore
0	56	Male	UK	83094	5	8	5	Low	Bronze	100.0
1	69	Male	UK	86860	10	2	8	Medium	Gold	100.0
2	46	Female	USA	60173	8	10	18	Medium	Silver	100.0
3	32	Female	UK	73884	7	10	16	Low	Gold	100.0
4	60	Male	UK	97546	6	4	13	Low	Bronze	82.0

We can understand what each variable represents and how it might influence customer satisfaction. This process helps us identify important features, such as age or loyalty, that are likely to impact how satisfied a customer feels. We also look for patterns in the data, such as whether certain loyalty levels correspond with higher satisfaction or if specific age groups tend to be less satisfied. Additionally, we assess the limitations of traditional group-based analysis, which can overlook individual differences.



To deepen our understanding, we use data visualisation techniques. We plot the distribution of satisfaction scores to reveal how customer sentiment is spread across the dataset and to spot any unusual clusters or outliers.



Furthermore, we generate a feature correlation heatmap, which visually summarises the relationships between all numerical features, including the satisfaction score. This heatmap helps us quickly identify which variables are most strongly associated with satisfaction and whether any features are highly correlated with each other, guiding our feature selection for modelling. By combining these exploratory and visual analyses, we lay a strong foundation for building a machine learning model that predicts personalised satisfaction scores and detects mismatches between expected and actual satisfaction, supporting the project's goal of uncovering hidden dissatisfaction and enabling targeted business improvements.

Data preparation

Data preparation is a step-by-step process that transforms raw data into a clean, structured format suitable for modelling. Here's how it is done in this project.

1: Remove Irrelevant Columns

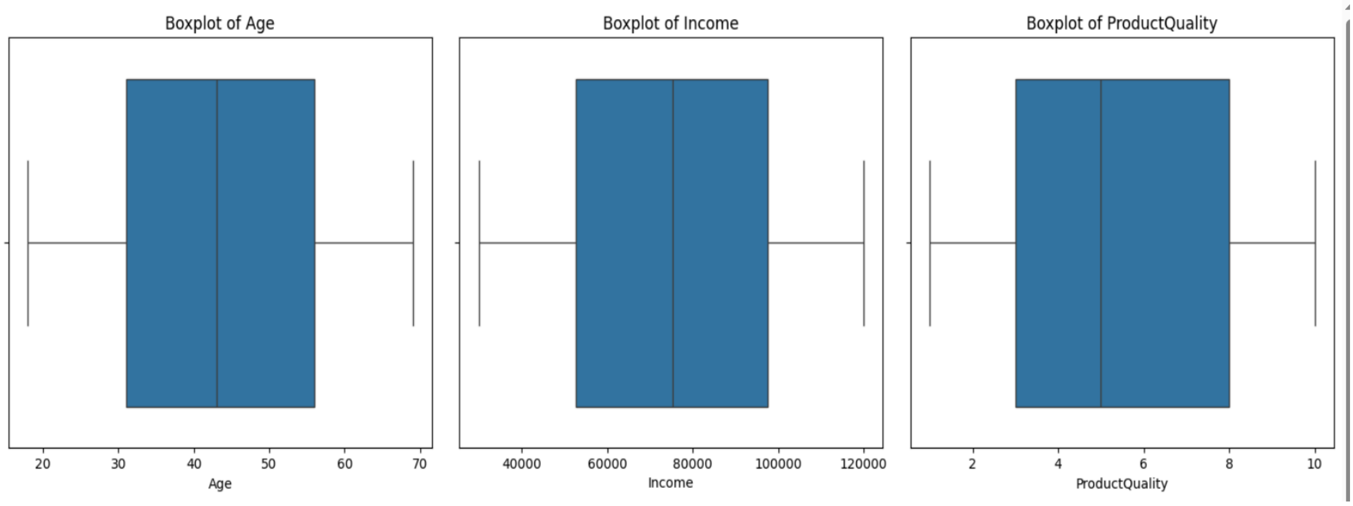
We start by dropping columns that do not contribute to the prediction, such as CustomerID. This reduces noise and focuses the analysis on meaningful information.
You can add a screenshot of the dataset before and after dropping irrelevant columns to show this change.

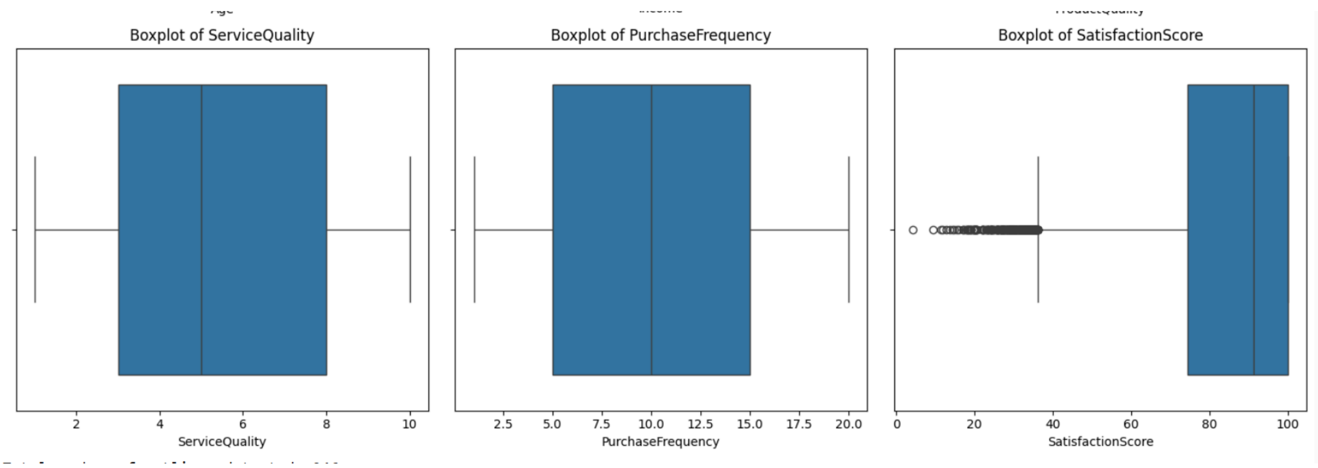
2: Handle Missing Values

After checking the dataset, I confirmed that there are no missing values in any of the features.

3: Detect and Handle Outliers

In the data preparation stage, it's important to identify and handle outliers, values that are unusually high or low compared to the rest of the data. Outliers can distort the training of machine learning models, especially when predicting customer satisfaction, because they may be the result of data entry errors or rare, unrepresentative cases.





1. Visual Detection with Boxplots**:**

   First, we use boxplots to inspect each numerical feature, including Satisfaction Score visually. A boxplot shows the spread of the data and highlights any points that fall far outside the normal range. Outliers appear as dots or stars beyond the “whiskers” of the boxplot. This helps us quickly spot which features might have extreme values.
2. Statistical Detection with Z-Score:

   Next, we use the z-score method to statistically flag outliers. The z-score measures how many standard deviations a value is from the mean. If a value’s z-score is greater than 3 (or less than -3), it’s considered an outlier.

```
Total number of outliers detected: 246
      Age  Gender Country  Income  ProductQuality  ServiceQuality  \
142    31  Female     USA   67042               1               1
162    21    Male  Canada   64783               1               1
714    36    Male  France   32050               3               1
845    18    Male  Canada   55374               1               1
1415   25    Male  France  111566               1               1

      PurchaseFrequency FeedbackScore LoyaltyLevel  SatisfactionScore
142                   9          High       Silver              26.67
162                   2           Low       Bronze              29.03
714                  10           Low       Bronze              30.52
845                  11           Low       Bronze              20.11
1415                  1          High       Bronze              32.45
```

3. flagged outliers

In my dataset, the target is customer satisfaction scores, so the model needs reliable data. Outliers, like unusual satisfaction scores, can mislead the model and harm prediction accuracy. By identifying these extremes, I ensure the model focuses on realistic patterns, leading to more trustworthy predictions for each customer.

```
Outliers flagged successfully!
   IsOutlier  Age  Income  ProductQuality  ServiceQuality  PurchaseFrequency  \
0          0   56   83094               5               8                  5
1          0   69   86860              10               2                  8
2          0   46   60173               8              10                 18
3          0   32   73884               7              10                 16
4          0   60   97546               6               4                 13

   SatisfactionScore
0              100.0
1              100.0
2              100.0
3              100.0
4               82.0
Total flagged as outliers: 246
```

4: Encode Categorical Variables

To prepare the data for machine learning, I identified categorical features like gender, country, loyalty level, and feedback score. I applied one-hot encoding to convert these categories into separate binary columns, ensuring the algorithms could process the data without assuming a natural order. After encoding, I checked the shape and previewed the first few rows of the new Data Frame to confirm the transformation was successful and the data was ready for modelling.

	Age	Income	ProductQuality	ServiceQuality	PurchaseFrequency	SatisfactionScore	IsOutlier	Gender_Male
0	56	83094	5	8	5	100.0	0	True
1	69	86860	10	2	8	100.0	0	True
2	46	60173	8	10	18	100.0	0	False
3	32	73884	7	10	16	100.0	0	False
4	60	97546	6	4	13	82.0	0	True

Country_France	Country_Germany	Country_UK	Country_USA	FeedbackScore_Low	FeedbackScore_Medium	LoyaltyLevel_Gold	LoyaltyLevel_Silver
False	False	True	False	True	False	False	False
False	False	True	False	False	True	True	False
False	False	False	True	False	True	False	True
False	False	True	False	True	False	True	False
False	False	True	False	True	False	False	False

5: Scale Numerical Features

To get the data ready for modelling, I first converted text categories into numbers for features like gender, loyalty level, and feedback score. This is important because machine learning algorithms work with numbers, not text, and for features with a natural order, like loyalty levels, this approach preserves their ranking. After encoding, I scaled the numerical features such as age, income, product quality, and service quality. Scaling ensures all variables are on a similar scale, which is especially helpful for algorithms that are sensitive to the size of the numbers.

6: Select Relevant Features

Finally, we retain only the columns that are most likely to influence customer satisfaction, such as age, income, product and service quality, purchase frequency, and satisfaction score.

Modelling

In the modelling part of this project, the main goal was to predict what each customer's satisfaction score should be, based on their personal and behavioural data, like age, income, loyalty level, and shopping habits, instead of just looking at the feedback they gave. This approach helps us create a personalised satisfaction estimate for every customer.

I tried out several popular machine learning models:

Let me explain how we trained and evaluated the models:

1. Split the Data:
   I divided the dataset into two parts: a training set to teach the model and a test set to see how well the model predicts satisfaction for new customers.
2. Train the Models:
   Each model was trained using the training data, learning the connections between customer features and their satisfaction scores.
3. Make Predictions:
   After training, the models predicted satisfaction scores for the test set, which we can compare to the actual scores customers gave.
4. Evaluate Performance:
   To see how well each model worked, I used metrics like Root Mean Squared Error (RMSE), which shows how far off the predictions are from the real scores (lower is better), and R^2 (R-squared), which shows how much of the variation in satisfaction the model can explain (higher is better).

Regression Model Comparison for Predicting Customer Satisfaction

To predict each customer's satisfaction score using their personal and behavioural data, I trained and evaluated five different machine learning models: Decision Tree, Random Forest, Gradient Boosting, Support Vector Regression (SVR), and a Deep Learning neural network. Here's how they performed:

1. Accuracy (R^2 Score):

- Gradient Boosting and SVR led the way, each explaining about 78.5% of the variation in satisfaction scores ($R^2 \approx 0.785$).
- Random Forest followed closely with an R^2 of 0.768, showing strong predictive power.
- Decision Tree was less effective, capturing only about 54.4% of the variation.
- The Deep Learning model showed moderate performance with an R^2 of around 58.5%.

2. Prediction Errors (RMSE & MAE):

- Both Gradient Boosting and SVR had the lowest errors, with RMSE around 7.76 and MAE between 5.5 and 5.7, meaning the average prediction was within 5 to 6 points of the actual score.
- Random Forest had slightly higher errors (RMSE = 8.07, MAE ≈ 5.68) but remained quite accurate.
- Decision Tree had larger errors (RMSE = 11.31, MAE = 7.62), meaning predictions were less reliable.
- The Deep Learning model had moderate errors (MAE ≈ 8.4), showing it was less precise than the top models.

3. Consistency (Pearson Correlation):

- The top three models, Gradient Boosting, SVR, and Random Forest, all showed strong positive correlations (around 0.87 to 0.89) with actual satisfaction scores, indicating they closely followed real trends.
- The Decision Tree's correlation was lower (0.78), showing more mismatch between predictions and reality.
- The Deep Learning model had a correlation of about 0.82, which is decent but not as strong as the best-performing models.

During hyperparameter tuning, Gradient Boosting and SVR continued to outperform others, each explaining about 78.6% of the score variation with low prediction errors. Random Forest also performed well, with an R^2 of 77.3%. In contrast, Decision Tree and Deep Learning models struggled to capture the complexity of the data, delivering less accurate predictions.

Classification

After performing regression to predict individual satisfaction scores, I used classification models to support the second goal of my project: **Mismatch Detection**. This step helped identify whether a customer is **satisfied or unsatisfied**, enabling a binary categorisation for easier interpretation and action. I evaluated multiple models, such as Decision Trees, Random Forests, and Gradient Boosting, using cross-validation to ensure robust results. The metrics from the classification report, **accuracy, sensitivity, specificity, and AUC**, allowed me to compare how well each model detects satisfaction correctly. For instance, high sensitivity ensures that most truly satisfied customers are correctly identified, while specificity highlights the model's ability to detect unsatisfied ones. The confusion matrix and heatmap visualisations further clarified the model's performance. This classification step directly supports my project goal of **detecting mismatches** between predicted and actual satisfaction, making it easier for businesses to take **personalised corrective actions**

Evaluation

A	B	C	D	E	F	G	H	I	J	K	L	M
	GLM	SD+-	DL	SD+-	DT	SD+-	RF	SD+-	GBT	SD+-	SVM	SD+-
root mean squared error	9.323	0.201	7.677	0.216	8.778	0.209	9.544	0.236	7.864	0.191	7.96	0.111
absolute error	7.428	0.145	5.361	0.16	6.143	0.174	7.391	0.161	5.521	0.14	5.538	0.062
relative error lenient	9%	0.20%	7%	0.20%	8%	0.20%	9%	0.20%	7%	0.20%	7%	0.20%
squarred error	86.948	3.741	58.977	3.304	77.095	3.683	91.124	4.496	61.874	2.997	63.368	1.771
correlation	0.837	0.007	0.894	0.006	0.859	0.008	0.864	0.008	0.888	0.006	0.889	0.004

Using AI Studio (Altair), I compared several regression models after training and tuning. Gradient Boosting achieved the best overall performance, with the lowest RMSE (7.864), a low absolute error (5.521), and a high correlation (0.888) to actual scores. SVM and Deep Learning followed closely, both with RMSE around 7.96 and 7.677, correlations of 0.889 and 0.894, respectively. Models like GLM (RMSE = 9.323) and Random Forest (RMSE = 9.544) had higher errors, while Decision Tree performed moderately (RMSE = 8.778). These results show that Gradient Boosting captured satisfaction patterns most accurately in the AI Studio benchmark.

	DL	SD+-	DT	SD+-	RF	SD+-	GBT	SD+-	SVM	SD+-
root mean squared error	15.89	6.97	11.306	0.195	8.04	0.1	7.75	0.096	7.749	0.1
absolute error	13.2	6.47	7.623	0.154	5.64	0.082	5.75	0.066	5.523	0.082
relative error lenient	10.27%		11%		8%	0.10%	8%	0.10%	8%	0.10%
squarred error	301.29	275.26	127.82	4.36	64.74	1.73	60.018	1.49	60.042	1.59
correlation	0.84	Nan	0.77	0.16	0.882	0.004	0.889	0.003	0.887	0.004

After developing and testing several regression models in Python, including Decision Tree, Random Forest, Gradient Boosting, Support Vector Machine, and Deep Learning, I used AI Studio to benchmark their performance. This platform provided a detailed comparison using key metrics in Python, Gradient Boosting emerged as the top performer (RMSE = 7.75, Absolute Error = 5.75, Correlation = 0.889), closely followed by SVM and Random Forest. Deep Learning showed a strong correlation (0.84) but higher errors (RMSE = 15.89), while Decision Tree performed less accurately (RMSE = 11.31, Corr = 0.77). These results show that Gradient Boosting predicted customer satisfaction most accurately after optimisation.

goal.

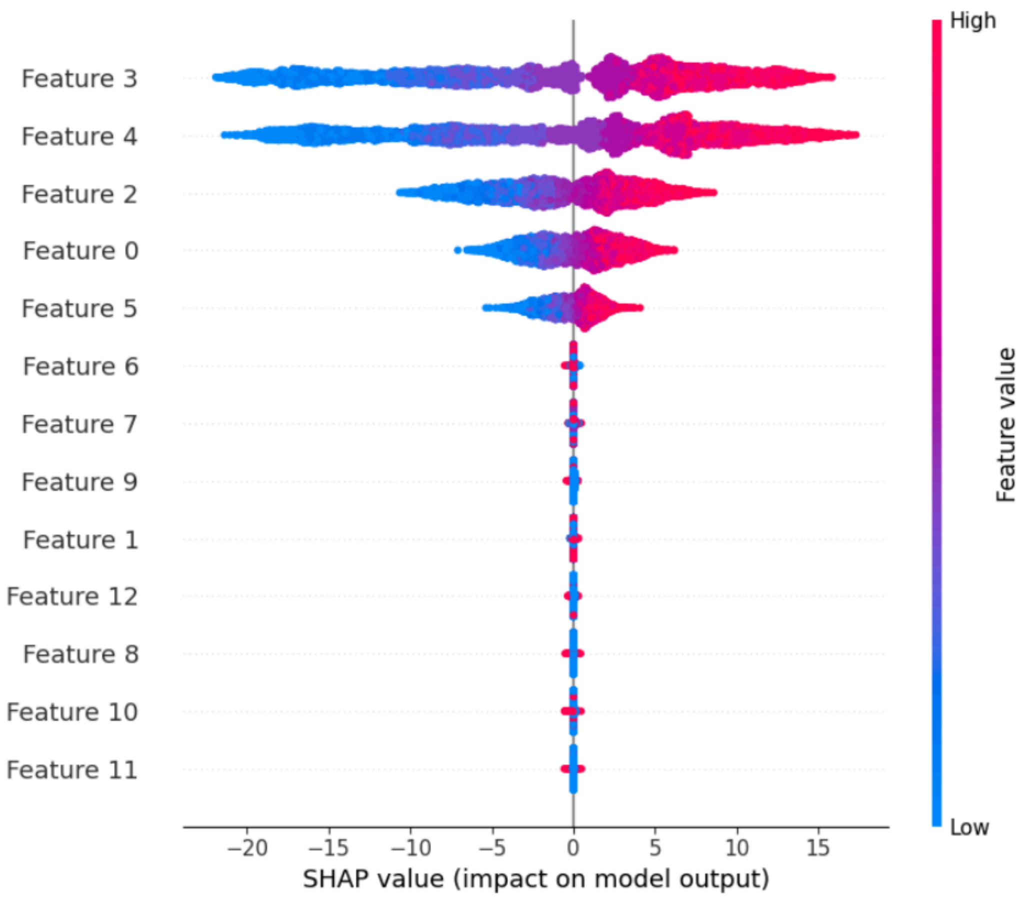
However, when comparing models directly using Python code, the Gradient Boosting model stood out as the top performer, achieving the lowest RMSE (7.76) and the highest R^2 score (0.785). This demonstrated that it predicted satisfaction scores most accurately in this particular setup. Although Deep Learning came close, it performed marginally worse, probably due to limited training time or data size. Consequently, Gradient Boosting was selected as the most effective model for predicting personalised satisfaction.

Once the final model was trained and optimised, it was used to forecast what each customer's satisfaction score should be, based on their profile and behaviour, such as age, loyalty level, and shopping frequency. These predicted scores were then compared to the actual satisfaction scores provided by the customers. A significant discrepancy between predicted and actual scores can reveal hidden issues or unmet expectations.

This personalised comparison helps to identify customers who are less satisfied than expected, something that average group-based analysis might overlook. For example, if the model predicts a customer to be highly satisfied but they give a low rating, this mismatch signals a need for investigation and possible service improvement. It enables businesses to take individual, targeted actions that enhance customer experience more effectively.

After completing the regression analysis, the subsequent step was to identify these mismatches through classification. I created a new label, such as “Mismatch” or “No Mismatch,” based on the difference between predicted and actual satisfaction scores. Then, I employed classification algorithms like Random Forest Classifier and Gradient Boosting Classifier to train models capable of automatically detecting whether a customer’s feedback aligns with expectations. This classification step transforms numerical prediction results into clear, actionable insights, helping businesses swiftly identify dissatisfied customers and respond more efficiently. It builds on the regression outcomes and moves the solution closer to real-world application.

Personalised Satisfaction Predictions using SHAP (Shapley Additive Explanations

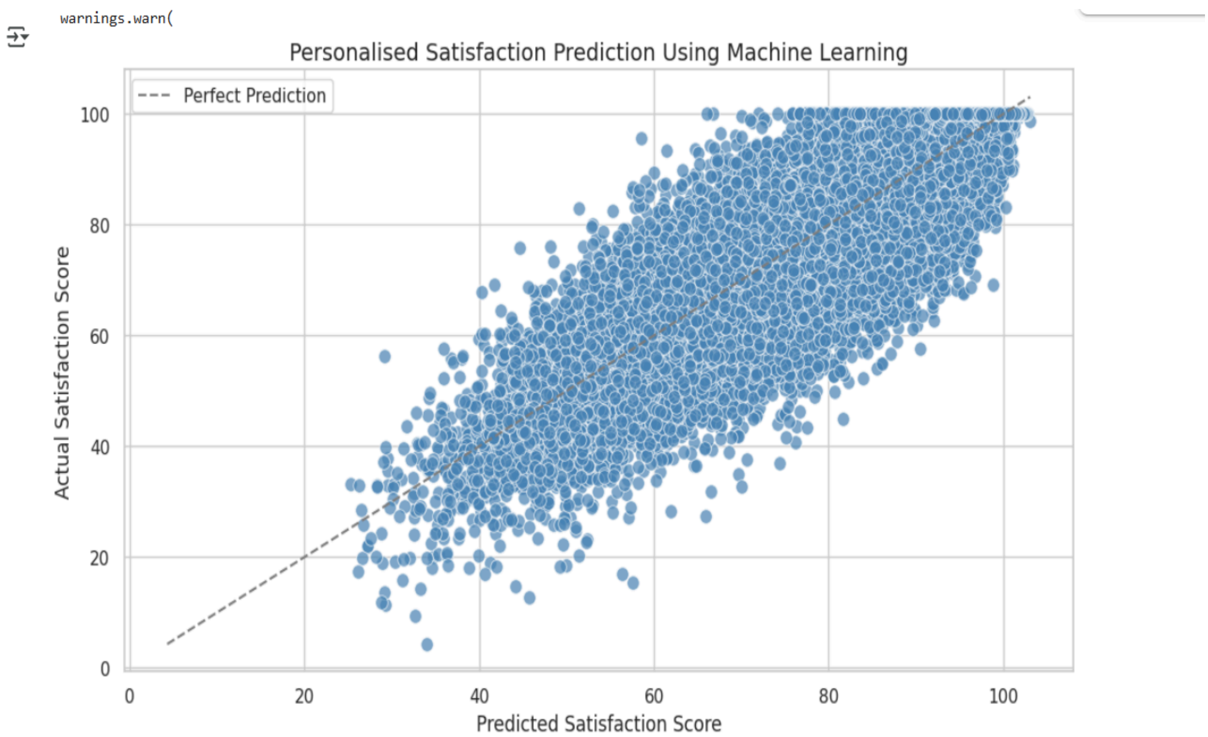


The SHAP summary plot below explains which features most influenced the model's decision to label a customer as satisfied or unsatisfied. Each point in the plot represents one customer, and the position shows how much a particular feature pushed the model's prediction higher or lower.

For example, if "Service Quality" appears at the top of the SHAP plot with many red dots on the right, it means high service quality **strongly increases** the chance of a customer being predicted as **satisfied**. On the other hand, blue dots on the left for "Response Time" may show that long response times **reduce** satisfaction.

This level of insight helps identify mismatches, such as a customer being predicted as satisfied despite low ratings on key features, which might flag hidden issues. SHAP provides a form of **personalised model interpretation**, allowing us to dig into individual cases and understand **what drives satisfaction at a deeper level.**

Compare Predicted vs Actual Customer Satisfaction: Detecting Mismatches



This chart shows how accurately our machine learning model predicts each customer's satisfaction score.

Each point represents a customer , the closer a point is to the diagonal line, the better the prediction.
When points fall far from the line, it highlights a **mismatch** between what the model expected and what the customer actually felt.
These mismatches help us **identify hidden issues** or **unexpected experiences**, making it easier to take targeted action and improve customer satisfaction

Classification

After successfully predicting customer satisfaction scores through regression analysis, the next logical step in the project involved identifying mismatches between predicted and actual satisfaction levels, a critical task for understanding unmet expectations. This required a shift from predicting continuous scores to classifying customers as “satisfied” or “unsatisfied,” which is best handled through classification models. In this stage, we explored a range of machine learning classifiers to detect satisfaction mismatches, aiming to flag customers whose experience fell short of their predicted satisfaction.

	NB	SD+-	GLM	SD+-	Logit	SD+-	FLM	SD+-	DL	SD+-	DT	SD+-	RF	SD+-
accuracy	0.54	0.02	0.64	0.01	0.872	0.004	0.571	0.010	0.871	0.003	0.836	0.003	0.8440	0.0033
classification_error	0.46	0.02	0.36	0.01	0.128	0.004	0.429	0.010	0.129	0.003	0.164	0.003	0.1560	0.0033
AUC	0.62	0.04	0.71	0.02	0.951	0.003	0.603	0.012	0.949	0.002	0.919	0.002	0.9255	0.0023
precision	0.52	0.02	0.60	0.04	0.870	0.011	0.551	0.006	0.868	0.009	0.824	0.008	0.8301	0.0084
recall	0.96	0.02	0.82	0.03	0.876	0.008	0.762	0.017	0.874	0.008	0.857	0.005	0.8649	0.0063
f_measure	0.68	0.02	0.69	0.02	0.873	0.004	0.639	0.009	0.871	0.003	0.840	0.003	0.8471	0.0040
sensitivity	0.96	0.02	0.82	0.03	0.876	0.008	0.762	0.017	0.874	0.008	0.857	0.005	0.8649	0.0063
specificity	0.11	0.04	0.46	0.04	0.869	0.009	0.382	0.009	0.867	0.006	0.816	0.006	0.8231	0.0054

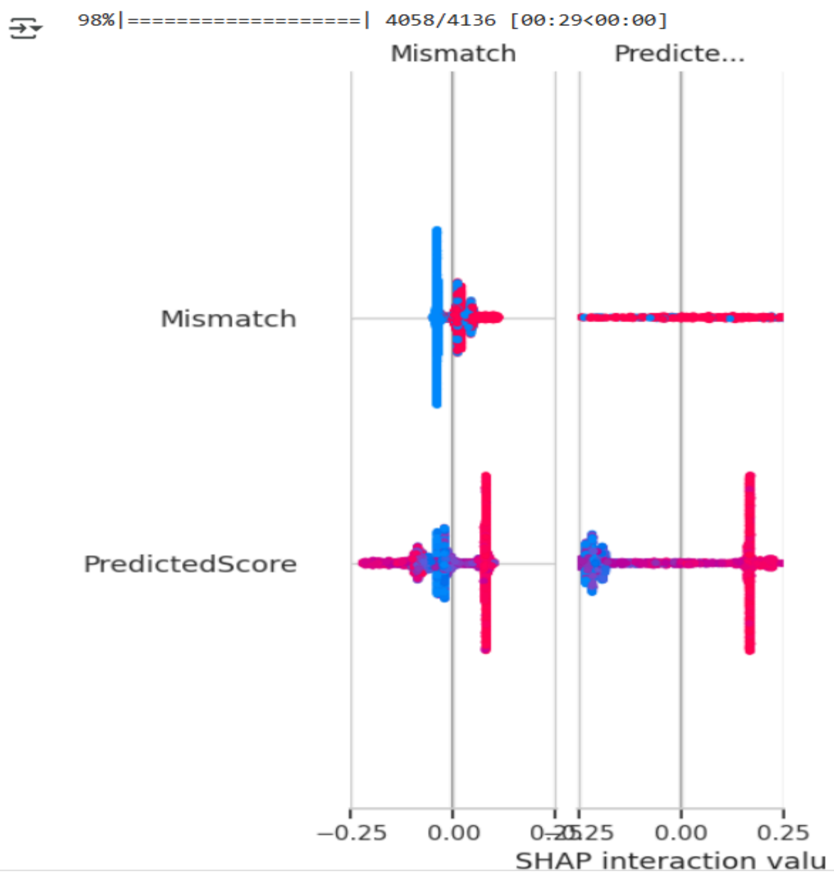
In the context of my project, Personalised Satisfaction Prediction and Mismatch Detection Using Machine Learning, the AI Studio classification results clearly show that Logistic Regression is the most effective model for the mismatch detection step. It achieved the highest accuracy (0.872), lowest classification error (0.128), and the best AUC (0.951), along with strong precision (0.870), recall (0.876), and specificity (0.869). Deep Learning (accuracy 0.871, AUC 0.949) and Gradient Boosting (accuracy 0.867, AUC 0.949) were close competitors, making them strong alternatives. Since mismatch detection is critical for identifying customers, whose actual satisfaction is lower than predicted, these top-performing models, especially Logistic Regression, offer the most reliable foundation for turning prediction results into actionable insights for targeted customer experience improvements.

	DL	DT	RF	GBT	SVM
accuracy	0.60	0.86	0.90	0.91	0.59
classification_error	0.39	0.14	0.10	0.09	0.40
AUC	0.61	0.86	0.97	0.97	0.62
precision 0	0.90	0.86	0.90	0.91	0.62
precision 1	0.50	0.85	0.91	0.90	0.59
recall 0	0.26	0.85	0.91	0.90	0.55
recall 1	0.97	0.86	0.90	0.91	0.65
f_measure 0	0.40	0.86	0.91	0.91	0.58
f_measure 1	0.71	0.86	0.90	0.91	0.62
sensitivity	0.97	0.85	0.90	0.90	0.65
specificity	0.25	0.86	0.91	0.90	0.54

In the **Python-based mismatch detection step** of my project, **Gradient Boosting** delivered the best results with the highest accuracy (**0.91**), lowest classification error (**0.09**), excellent AUC (**0.97**), and balanced precision/recall for both classes (around **0.90–0.91**). **Random Forest** was a close second (accuracy **0.90**, AUC **0.97**), while **Decision Tree** also performed well (accuracy **0.86**, AUC **0.86**) but was slightly less consistent. **Deep Learning** (accuracy **0.60**, specificity **0.25**) and **SVM** (accuracy **0.59**, specificity **0.54**) scored lower, especially in identifying non-mismatch cases. These results confirm that **Gradient Boosting**, trained and tuned in Python, is the most reliable model for accurately detecting satisfaction mismatches in the personalised satisfaction prediction workflow.

In the classification stage, both AI Studio and Python delivered strong results with different leading models. **AI Studio's Logistic Regression** achieved the best performance there, with an **accuracy of 0.872**, **AUC of 0.951**, and balanced precision and recall. In contrast, **Python's Gradient Boosting** outperformed all models with even higher **accuracy (0.91)** and **AUC (0.97)**, offering excellent detection of both satisfied and unsatisfied cases. While Logistic Regression is more interpretable, Gradient Boosting offers stronger predictive power, making it the top choice for complex mismatch detection

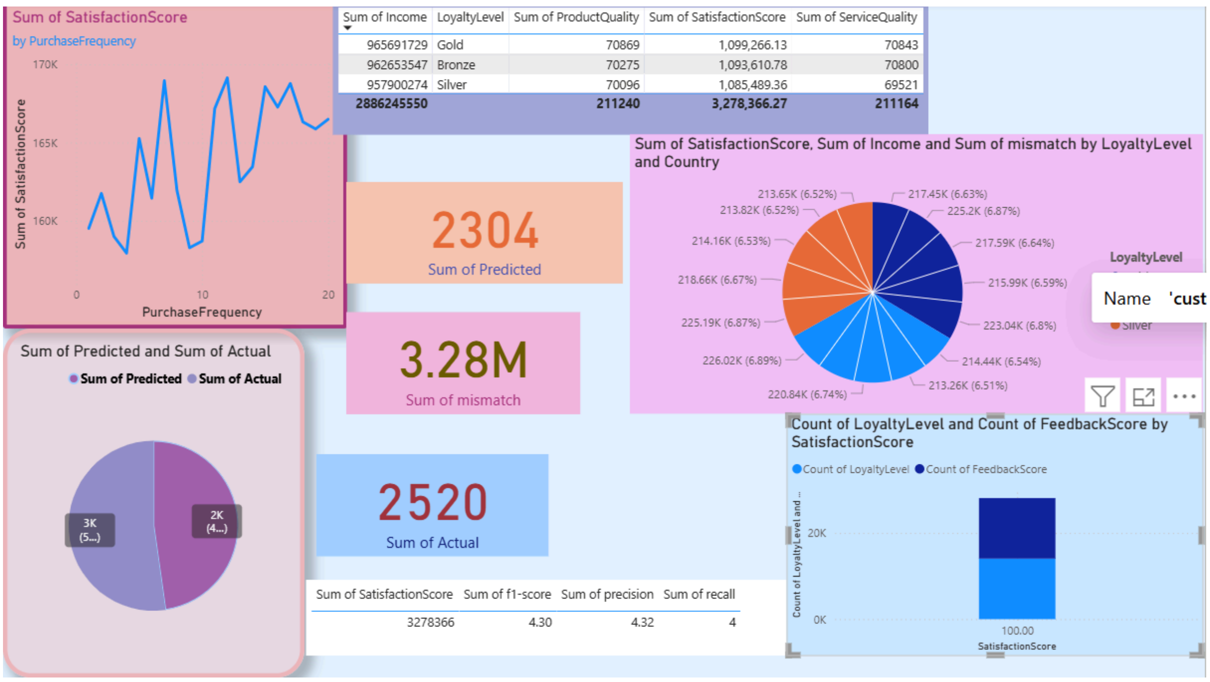
Explaining Model Predictions with SHAP for Personalised Satisfaction



To make the model's predictions more transparent and understandable, SHAP (Shapley Additive Explanations) was used to interpret the outputs of the trained Random Forest classifier. After reconstructing the test data with the correct feature names following polynomial transformation, a SHAP explainer was created using the training data. SHAP values were then calculated for each customer in the test set, showing how much each feature contributed to the prediction of satisfaction or dissatisfaction. A summary plot was generated to provide a global overview of feature importance, where red dots represent high feature values and blue dots represent low values. The further a point is from zero, the more influence that feature has on the prediction. Within the context of this project, SHAP helped reveal which factors most strongly affected satisfaction predictions, making it easier to detect hidden mismatches, cases where a customer is predicted to be satisfied but has underlying dissatisfaction signals. This level of interpretability supports personalised customer experience improvements and promotes confidence in data-driven decisions.

Deployment

This project, titled **"Personalised Satisfaction Prediction and Mismatch Detection Using Machine Learning,"** aims to improve how businesses understand and respond to customer satisfaction. It uses machine learning models to predict how satisfied each customer is likely to be, based on various service and interaction features. It then compares these predictions with actual customer feedback to detect mismatches—cases where a customer appears satisfied in the data but is not, or vice versa. These mismatches help identify hidden issues in the customer experience. The project also uses SHAP (Shapley Additive Explanations) to explain which factors most influence satisfaction predictions, offering transparency and actionable insights. The results are visualised in a Power BI dashboard, allowing business teams to track satisfaction trends, pinpoint dissatisfaction drivers, and take targeted actions to improve customer engagement.



In the Deployment phase, we move from testing models in a development environment to applying the best-performing model in a real-world setting. After evaluating several machine learning algorithms, we selected the **Gradient Boosting model** as the final model for deployment, based on its strong predictive performance, achieving the **lowest RMSE (7.76)** and the **highest R^2 score (0.785)**. This indicates it can reliably predict individual customer satisfaction scores. Although we also tested a deep learning model using the code provided in evaluate_deep_learning_model, it performed slightly less accurately in this case.

After the model is trained and ready, it can be saved using simple Python tools like joblib or pickle. This means you don't have to retrain it every time; you can just load it and use it whenever you need. Once the model is deployed, it can be added to a business system, like a customer feedback platform, to automatically predict what each customer's satisfaction score should be. By comparing these predicted scores with the actual feedback from customers, the business can quickly spot when someone's experience isn't meeting expectations. This helps the company respond faster, fix problems for individual customers, and improve overall satisfaction and loyalty. In this way, the model becomes a practical tool for making better decisions and giving customers a better experience, not just a technical project

conclusion

This project aimed to predict each customer's satisfaction in a personalised way and identify mismatches between what was expected and what occurred, utilising machine learning. By following the CRISP-DM steps, I explored the data, prepared it, tested different models, and chose the one that worked best. Gradient Boosting provided the most accurate predictions, enabling us to estimate the level of satisfaction each customer should expect based on their information. By comparing these predictions to real feedback, we can find customers who might be less happy than expected and help them in a more targeted way. This shows that machine learning can do more than just analyse feedback; it can help businesses act early to improve customer experience. Next, I'll focus on fine-tuning the model to make these predictions even more reliable.

Acknowledgments

I would like to thank my supervisor, Terri Hoare, for her clear explanations, guidance, and support throughout this project. Her advice was invaluable at every stage.

References

Altair AI Studio. [Online platform]. Available at: https://www.altair.com/ai-studio/ (Accessed: 19 Jun 2025).

Google Collaboratory. (2025). *Colab: Interactive Python Notebooks*. Available at: https://colab.research.google.com/ (Accessed: 20 JUN 2025).

https://docs.h2o.ai/h2o/latest-stable/h2o-docs/data-science/deep-learning.html (accessed: 27 Jun 2025).

https://www.datasciencecentral.com/crisp-dm-a-standard-methodology-to-ensure-a-good-outcome/ (accessed: 04 July 2025).

https://www.epa.gov/caddis/exploratory-data-analysis#:~:text=References-,Overview,step%20in%20any%20data%20analysis (accessed:05 July 2025)

Python code link:
https://colab.research.google.com/drive/1XvHhdaidj_Fhvra0jiDCAtTpJY6s-JyQ?usp=sharing