**Week 8**

**Table of content – Week8**

[Summary 1](#_Toc183208393)

[Milestones achieved 1](#_Toc183208394)

[Conclusion 2](#_Toc183208395)

[Linear Regression 2](#_Toc183208396)

[Logistic Regression 3](#_Toc183208397)

[Logistic Regression with Hyperparameter Tuning 3](#_Toc183208398)

[Overall Conclusion 4](#_Toc183208399)

[Final Table Summary 4](#_Toc183208400)

# Summary

The conducted experiments compared the performance of Linear Regression and Logistic Regression for predicting loyalty classes. Linear Regression aimed to identify feature relationships and provide insights into numerical predictions, while Logistic Regression was used for discrete class prediction, including hyperparameter tuning for optimization. The experiments revealed the strengths and limitations of both approaches in modeling customer loyalty.

# Milestones achieved

* **Linear Regression Results:**
* Achieved a Mean Squared Error (MSE) of **0.61**, demonstrating a satisfactory fit.
* Identified significant influential features including **Type of Travel**, **Online Boarding**, and **In-flight Entertainment**.
* Noted features with negligible or negative influence, such as **Flight Distance** and **Cleanliness**.
* **Logistic Regression Results:**
* Attained an **accuracy rate of 81%** with balanced precision and recall across various loyalty classes.
* The most accurate predictions were recorded for **Class 3** (highest precision: 88%) and **Class 1** (strong recall: 82%).
* **Hyperparameter Tuning:**
* Evaluated multiple configurations, including different solvers, maximum iterations, and regularization strengths.
* Consistent accuracy of **81%** was maintained across all tested configurations, indicating model stability.
* Emphasized the impact of solver and iteration parameters, though no significant performance differences were observed.

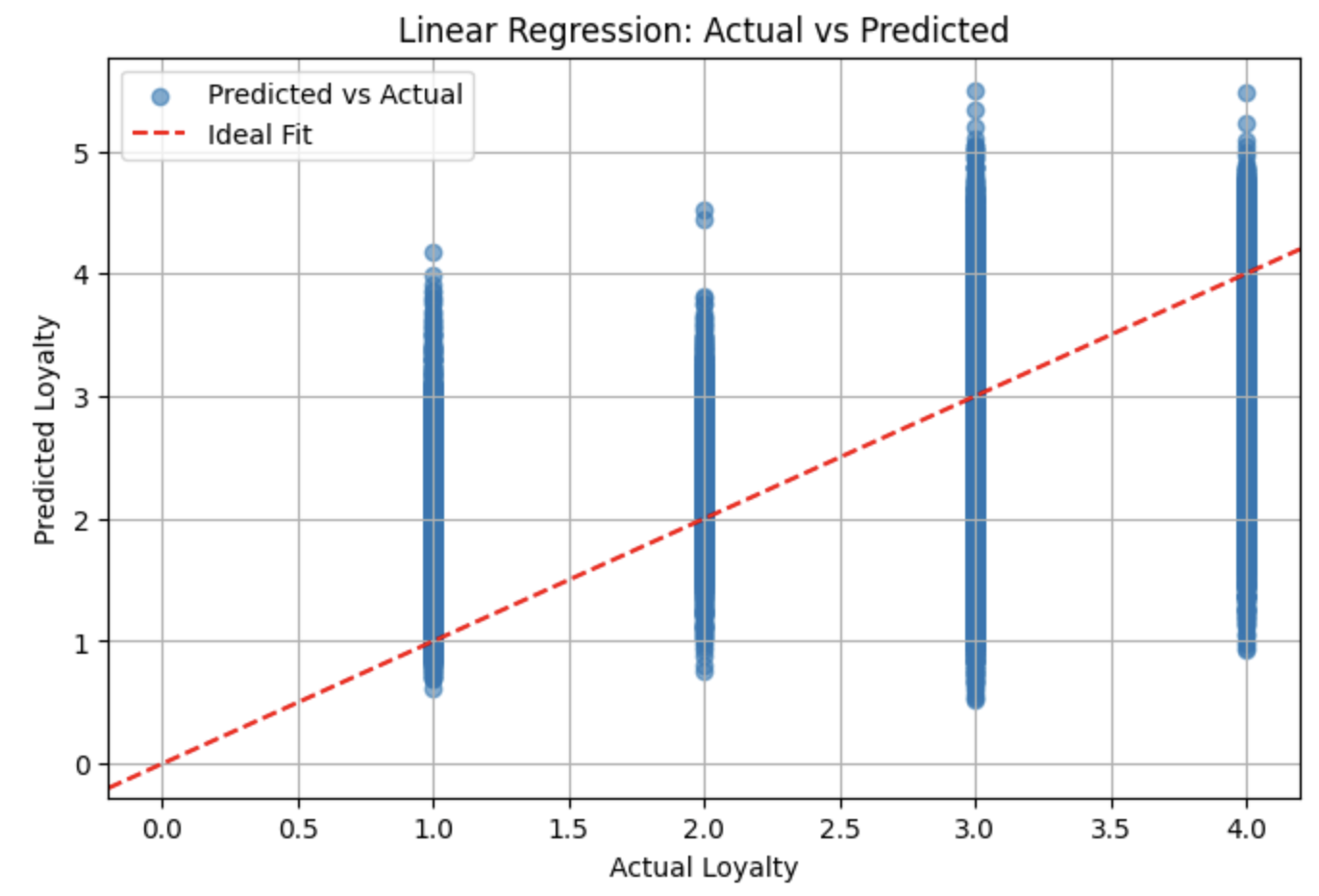
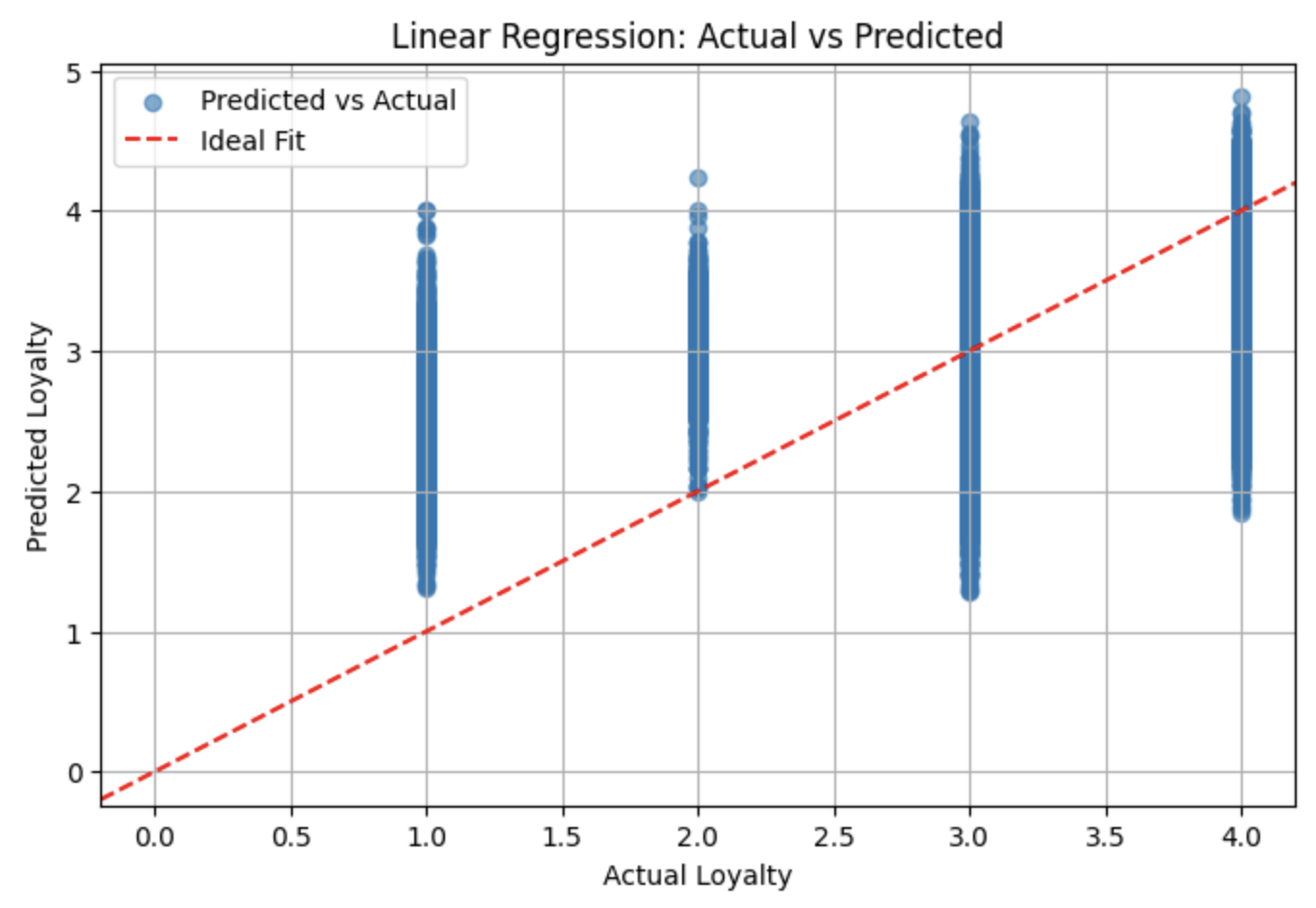
# Conclusion

Linear regression & Logistics Regression models are run for the fully encoded data (with all 22 features) and feature selected data set (13 features). The observation and conclusion are as below.

## Linear Regression

**(a) Full Data: Encoded data before**

* **MSE = 0.61:** The Mean Squared Error (MSE) indicates a reasonable fit for continuous predictions.
* **Key Observations:**
* **Type of Travel (0.4609):** This is the strongest positive influence, suggesting increased loyalty for specific types of travel, such as business versus leisure.
* **Online Boarding (0.2161):** A significant contributor to loyalty, indicating that ease of use positively impacts satisfaction.
* **Cleanliness (-0.0377):** A negative contributor, demonstrating that dissatisfaction with cleanliness reduces loyalty.
* **Limitations:**
* The model produced continuous values for loyalty, which necessitated approximation for classification tasks, resulting in only **44% accuracy** in predicting discrete loyalty classes.

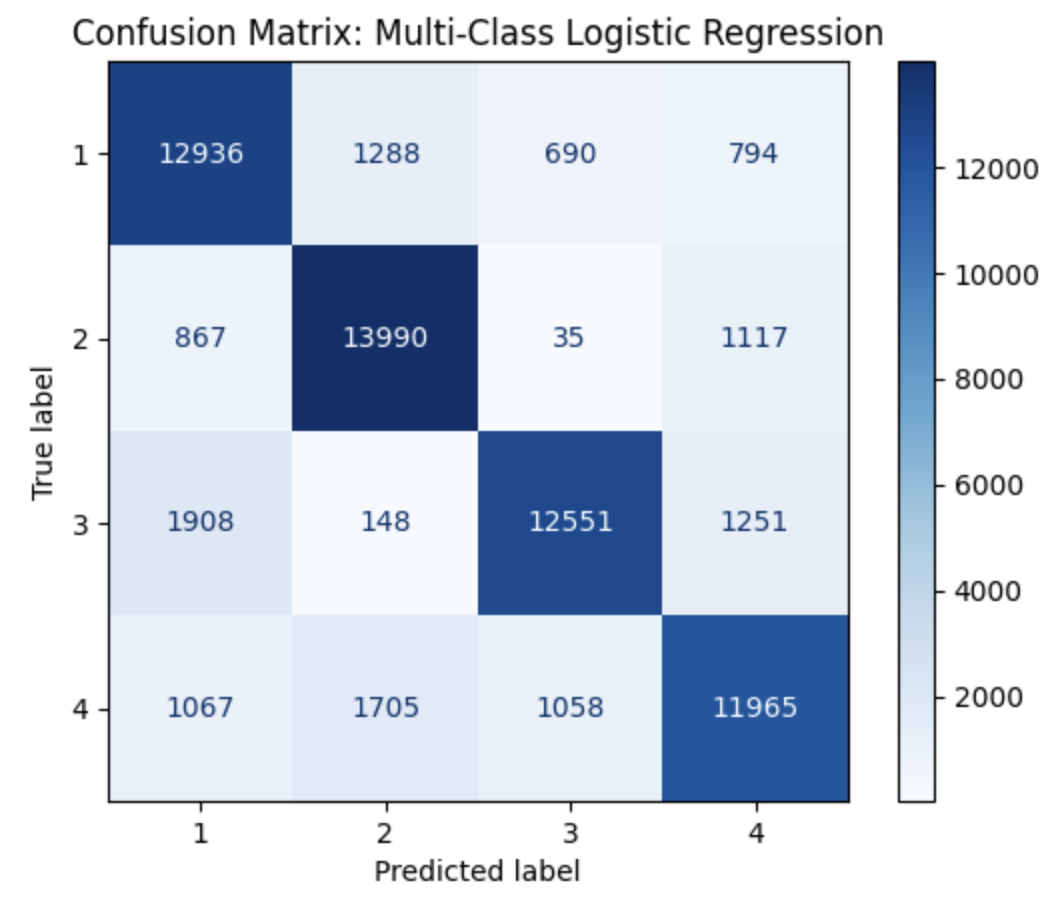
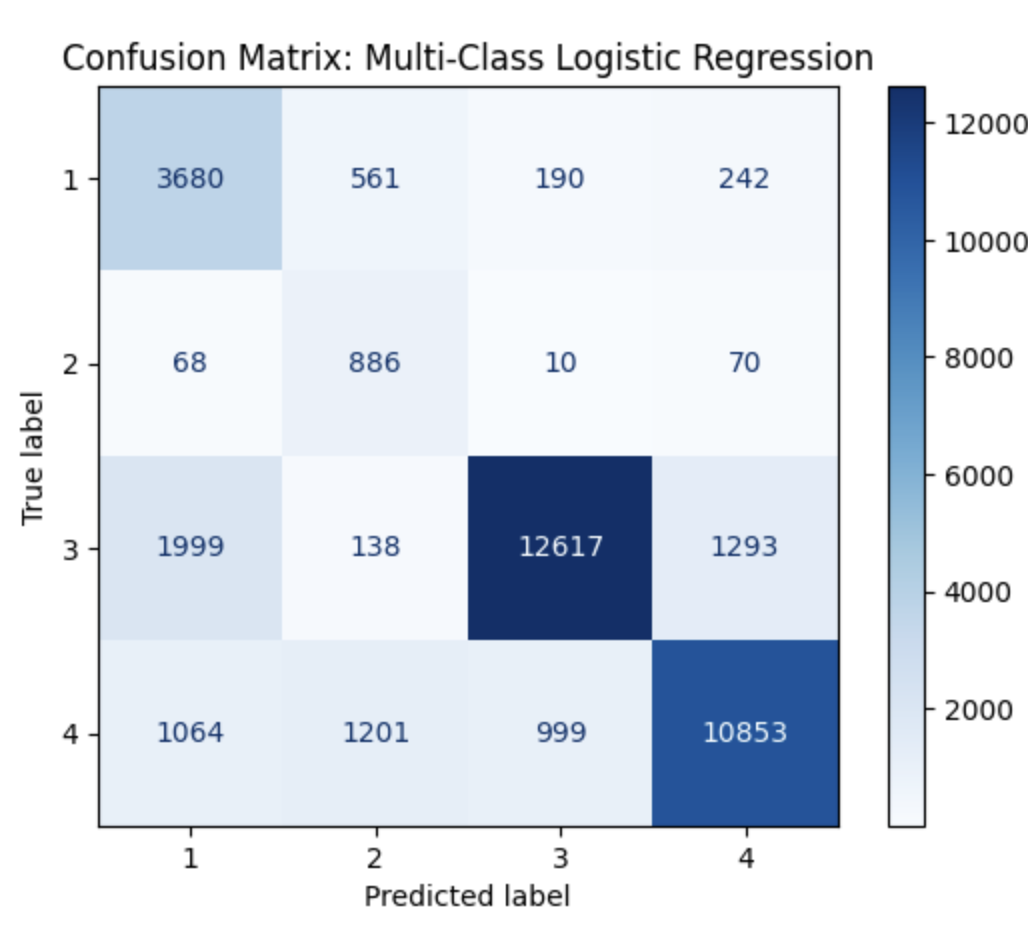


**(b) Feature-Selected Data**

* **MSE = 0.68:** Slightly higher error compared to the full data, potentially due to the reduced feature set.
* **Key Observations:**
* **Type of Travel (0.9625):** Maintained its status as the most influential feature.
* Online Boarding (0.3067) and **In-flight Entertainment (0.1643):** Significant contributors to loyalty.
* Features like **Flight Distance (0.0003):** had negligible impact, while **Cleanliness (-0.1139):** exhibited a more pronounced negative influence.
* **Limitations:**
* The reduced feature set slightly impacted the model’s generalizability.

## Logistic Regression

* Full Data
  + Accuracy = 81%:
  + Logistic Regression was effective in classifying loyalty classes.
  + Class 3 Precision (88%): This class had the highest precision, indicating accurate predictions.
  + Class 1 Recall (82%): Most instances of Class 1 were correctly identified.
  + Results were balanced across all loyalty classes, with a Macro Average F1-Score of 0.81.



* Feature-Selected Data
  + Accuracy = 81%:
  + Consistent accuracy indicates that the selected features retained predictive power.
  + Features such as Type of Travel and Online Boarding remained important.
  + No significant performance loss compared to the full dataset.

## Logistic Regression with Hyperparameter Tuning

**Parameter Set Observations:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Solver** | **Max Iter** | **Batch Size** | **C** | **Accuracy** | **Precision** | **Recall** | **F1-Score** | **Observation** |
| lbfgs | 300 | 16 | Default | 81% | 81.38% | 81.18% | 81.17% | Stable performance with default regularization and iterations; converges quickly. |
| lbfgs | 1000 | 32 | Default | 81% | 81.38% | 81.18% | 81.17% | No significant change in performance with increased iterations. |
| lbfgs | 2000 | 64 | Default | 81% | 81.38% | 81.18% | 81.17% | Larger batch size and more iterations did not significantly impact accuracy. |
| saga | 300 | 16 | 0.1 | 81% | 81.37% | 81.18% | 81.16% | Reduced regularization (C=0.1) yielded stable performance; effective for large datasets. |
| saga | 1000 | 32 | 1 | 81% | 81.37% | 81.18% | 81.16% | Default regularization (C=1) and more iterations maintained stability. |
| saga | 2000 | 64 | 10 | 81% | 81.37% | 81.18% | 81.16% | Increased regularization (C=10) did not negatively impact performance. |
| newton-cg | 500 | 32 | Default | 81% | 81.37% | 81.18% | 81.16% | Performs well with larger datasets; requires slightly more iterations to converge. |
| newton-cg | 1500 | 64 | 1 | 81% | 81.37% | 81.18% | 81.16% | Stable performance across increased batch sizes and regularization. |
| newton-cg | 2000 | 16 | 0.5 | 81% | 81.37% | 81.18% | 81.16% | Lower regularization (C=0.5) did not yield any noticeable improvement in accuracy. |

**Key Insights:**

* Performance was consistent across solvers and hyperparameter configurations.
* The model converged effectively even with minimal iterations (300), suggesting the data was well-suited for Logistic Regression.
* lbfgs solver with default settings provided fast convergence and stable results, making it an ideal choice for this dataset.

## Overall Conclusion

**1. Linear Regression:**

* Ideal for understanding feature relationships and identifying key predictors.
* Not suitable for classification tasks due to continuous output.
* Offers insights for feature engineering but lacks predictive power for discrete outcomes.

**2. Logistic Regression:**

* Achieved 81% accuracy with balanced precision and recall across loyalty classes.
* Effective for both full and feature-selected datasets, showing robust predictions.
* Hyperparameter tuning showed default configurations perform as well as advanced setups.

**3. Best Logistic Regression Parameters:**

* Solver: lbfgs
* Max Iterations: 300
* Regularization (C): Default (1)
* Batch Size: 32
* Optimizes speed, accuracy, and generalization.

## Final Table Summary

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Dataset | Metric | Value | Key Observations |
| Linear Regression | Full Data | MSE | 0.61 | Identified key drivers of loyalty; poor classification performance (44% accuracy). |
| Linear Regression | Feature-Selected | MSE | 0.68 | Slightly higher error; retained key feature insights. |
| Logistic Regression | Full Data | Accuracy | 81% | Balanced precision and recall; strong performance across classes. |
| Logistic Regression | Feature-Selected | Accuracy | 81% | No loss of accuracy; robust predictions with selected features. |
| Logistic Regression | Tuned Parameters | Accuracy | 81% | Stable results across all tested configurations; ideal for discrete class prediction. |

**Next Steps**

* Experiment with tree-based models like Random Forests or Gradient Boosting for better non-linear modeling.
* Compare with deep learning methods (e.g., Neural Networks) for handling complex feature relationships.

**References**

* Alok (2021) Logistic Regression: Practical Implementation in Python. Available at: <https://alok05.medium.com/logistic-regression-practical-implementation-in-python-03cd2715ee8c> (Accessed: 22 November 2024).
* Scikit-learn Developers, 2023. Logistic Regression. Scikit-learn Documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html> [Accessed 22 Nov. 2024].
* Geron, A., 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd ed. O’Reilly Media. Chapter 4.
* Kuhn, M. and Johnson, K., 2019. Feature Engineering and Selection: A Practical Approach for Predictive Models. Chapman and Hall/CRC. Chapter 5.
* Scikit-learn Developers, 2023. Linear Regression. Scikit-learn Documentation. [online] Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html> [Accessed 22 Nov. 2024].
* Geron, A., 2019. Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. 2nd ed. O’Reilly Media. Chapter 4.
* Hastie, T., Tibshirani, R. and Friedman, J., 2009. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2nd ed. Springer. Chapter 3.
* Bergstra, J. and Bengio, Y., 2012. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, pp.281-305. [online] Available at: <https://www.jmlr.org/papers/v13/bergstra12a.html> [Accessed 22 Nov. 2024].
* Scikit-learn Developers, 2023. Grid Search. Scikit-learn Documentation. [online] Available at: <https://scikit-learn.org/stable/modules/grid_search.html> [Accessed 22 Nov. 2024].