Dimensionality Reduction and Neural Network Modeling for Flight Price Categorization

MATH 5383 — Predictive Analytics

Saroar Jahan Shuba

December, 2025

1 Objective

The primary objective of this project is to develop a predictive classification framework for airline ticket prices by integrating dimensionality reduction with a neural network architecture. The project begins with the construction of a clean and fully numerical feature matrix derived from heterogeneous raw flight information. Four engineered variables — *Duration_Minutes*, *Dep_Hour*, *Arr_Hour*, and *Total_Stops_Num* — form the initial predictor dataset used for analysis. Principal Component Analysis (PCA) is applied to this feature matrix to identify the optimal number of principal components, denoted as m*, required to retain at least 80% of the total variance. These components are then used to reconstruct an approximate version of the original feature space, $\hat{X}$, which serves as the input for neural network modeling.

In parallel, the continuous variable *Price* is transformed into a binary response variable by separating the observations into two meaningful groups: *Cheap* (Class 0) and *Expensive* (Class 1), based on the median ticket price. An examination of class balance and sample size in each category provides insight into the complexity of the prediction problem.

The central task of this project is the iterative training and evaluation of a neural network classifier to predict the class labels derived from price. An initial network architecture of $\langle m*, 8, 4, 1\rangle$ is trained on the reconstructed PCA features $\hat{X}$, and its performance is evaluated using misclassification error, class-wise Mean Squared Error (MSE), and decision boundary visualization. To assess the robustness of the classifier, an 80–20 train–test split is repeated across $M = 20$ independent trials, allowing for estimation of stability in classification performance and identification of systematic weaknesses in the model.

Based on the results from the baseline model, a fine-tuned neural network with architecture $\langle m*, 16, 8, 4, 1\rangle$ is designed to explore whether additional representational ca-

pacity improves predictive accuracy. The analysis compares both architectures in terms of convergence behavior, class-wise performance, and overall generalization error.

Finally, the project incorporates a creative component through a sensitivity analysis that examines how the predicted probability of an expensive flight changes when one reconstructed feature is varied while the others are held constant. This analysis provides interpretability, highlighting which reconstructed features exert the strongest influence on model predictions.

Overall, the aim of this project is to construct a complete and statistically grounded workflow that leverages PCA, neural networks, stability analysis, and interpretability tools to classify flight prices effectively and to justify each design choice throughout the modeling process.

Table 1: Description of Variables Used in the Flight Price Classification Model

Variable	**Role**	**Nature**
Price	Response Variable (used to create class label)	Numerical (continuous)
Duration_Minutes	Predictor Variable	Numerical (continuous)
Dep_Hour	Predictor Variable	Numerical (discrete: 0–23)
Arr_Hour	Predictor Variable	Numerical (discrete: 0–23)
Total_Stops_Num	Predictor Variable	Numerical (integer count)

2 Data Description

2.1 Data Preprocessing

The dataset used in this project consists of 10,683 flight records obtained from the publicly available `Data_Train.csv` file. The dataset contains heterogeneous flight information, including airline carrier, journey dates, departure and arrival times, route descriptions, number of stops, and ticket prices. Before applying PCA and training neural network models, a series of preprocessing steps were performed to ensure completeness and numerical consistency.

Missing Values. An inspection of the original dataset revealed that none of the eleven raw columns contained missing values. Every flight record included complete information for variables such as *Airline*, *Duration*, *Dep_Time*, *Total_Stops*, and *Price*. However, during feature engineering—specifically when converting time and stop information into

numerical variables (*Duration_Minutes*, *Dep_Hour*, *Arr_Hour*, and *Total_Stops_Num*)—one row produced an invalid or non-parseable value, resulting in a `NaN`. This single row was removed. After this step, the final dataset used for PCA and neural network modeling contained

10,682 complete observations with no missing values.

Table 2: Missing Values Per Column in the Raw Dataset

Column	Missing Count
Airline	0
Date_of_Journey	0
Source	0
Destination	0
Route	0
Dep_Time	0
Arrival_Time	0
Duration	0
Total_Stops	0
Additional_Info	0
Price	0
Rows removed during feature engineering	**1**

Categorical Variables. The dataset includes several categorical variables such as *Airline*, *Source*, *Destination*, and *Route*. Since PCA operates only on numerical inputs, these categorical variables were excluded from dimensionality reduction. Instead, the analysis focused exclusively on four engineered numerical predictors: *Duration_Minutes*, *Dep_Hour*, *Arr_Hour*, and *Total_Stops_Num*. As a result, no encoding of categorical features was required.

Scaling and Standardization. Before applying PCA, all engineered numeric features were standardized using z-score normalization:

$$X_{\text{std}} = \frac{X - \mu}{\sigma}.$$

Standardization was essential because the predictors operate on different scales (e.g., duration measured in minutes vs. number of stops measured as small integers). Equalizing the scales ensures that PCA assigns balanced weight to each feature when computing principal components.

Transformations. No additional mathematical transformations (e.g., log, Box–Cox) were applied because the engineered variables were already interpretable in their natural scales, and there was no severe skewness or outlier behavior requiring correction.

Following preprocessing, the dataset consisted of 10,682 fully numerical and complete observations. These standardized predictors formed the basis for PCA, reconstruction, and subsequent neural network classification of flights into *Cheap* and *Expensive* categories.

2.2 Descriptive Statistics

This section summarizes the numerical characteristics of the engineered variables used in the flight price classification model. After preprocessing and feature engineering, four numerical predictors were constructed: *Duration_Minutes*, *Dep_Hour*, *Arr_Hour*, and *Total_Stops_Num*. The continuous response variable *Price* was retained for the purpose of creating binary class labels (Cheap vs. Expensive). Descriptive statistics provide insight into the distribution, scale, and variability of these variables prior to dimensionality reduction and neural network modeling.

Table 3 reports the mean, standard deviation, minimum, and maximum values for all numerical variables. These values help contextualize the behavior of the dataset. For example, the *Price* variable exhibits a wide range, reflecting substantial variation in flight fares. The *Duration_Minutes* variable also shows large variability, which aligns with the mixture of short domestic trips and long multi-stop itineraries. Departure and arrival hours are approximately uniformly distributed across the 24-hour day, while the *Total_Stops_Num* variable indicates that most flights have zero or one stop.

Table 3: Descriptive Statistics of Engineered Numerical Features

Variable	**Mean**	**Std. Dev.**	**Min**	**Max**
Price (INR)	9087.21	4611.55	1759	79512
Duration_Minutes	643.02	507.83	5	2860
Dep_Hour	12.49	5.75	0	23
Arr_Hour	13.35	6.86	0	23
Total_Stops_Num	0.82	0.68	0	4

The descriptive statistics indicate that the dataset contains considerable diversity in flight durations and price levels. This variation supports the need for dimensionality reduction methods such as PCA to extract the most informative structure from the predictor space before training a neural network classifier. The presence of multiple scales across predictors also highlights the importance of standardization prior to PCA, ensuring that no single variable disproportionately influences the principal components.

2.2.1 Summary Statistics of the Continuous Variable Price

The initial analysis of the raw ticket prices provides the following boundaries for the continuous response variable *Price* (in Indian rupees):

- **Minimum Price (Min):** 1,759
- **Maximum Price (Max):** 79,512
- **Range (Max – Min):** 77,753
- **Median Price:** 8,372

These statistics indicate that ticket prices in the dataset vary widely, with the cheapest observed fare around 1.8K and the most expensive fare close to 80K. The median of 8,372 rupees serves as a natural and robust central benchmark for separating relatively "cheap" and "expensive" flights.

2.2.2 Cluster Classification and Class Labels y_c

To create the binary class labels y_c, a simple and practically meaningful business rule was applied: each flight is classified as *cheap* or *expensive* based on whether its ticket price is below or above the median fare. This aligns with a realistic consumer perspective, where flights cheaper than the typical (median) price are considered budget-friendly, while those above the median are treated as premium or expensive options.

The classification rule is defined using the median price $\tilde{p} = 8{,}372$ as the threshold:

$$y_c = \begin{cases} 0, & \text{if Price} \leq \tilde{p} \quad \text{(Cheap flight)} \\ 1, & \text{if Price} > \tilde{p} \quad \text{(Expensive flight)} \end{cases} \tag{1}$$

The total number of new class labels is therefore $C = 2$, corresponding to:

$$\text{Class 0: Cheap flights,} \qquad \text{Class 1: Expensive flights.}$$

Figure 1 visualizes these two clusters in the `Price` data. Each point represents a flight, colored according to its assigned class label, and the black dashed horizontal line marks the median price $\tilde{p} = 8{,}372$ used as the decision boundary.

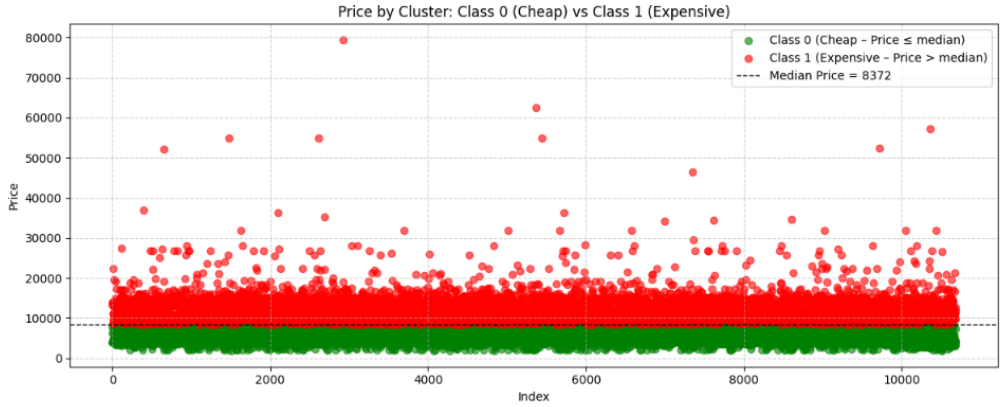

Figure 1: Visualization of the two identified clusters in the `Price` data. The black dashed line at the median price $\tilde{p} = 8{,}372$ separates Class 0 (Cheap) and Class 1 (Expensive).

2.2.3 Analysis of Class Sizes and Balance

After preprocessing the dataset and defining the binary price-based class labels, the total number of observations is $N = 10{,}683$. The distribution of the two classes is shown in Table 4.

Table 4: Distribution and Proportions of the Binary Flight Price Classes

Class Label (c_i)	**Condition**	**Observations (n_{c_i})**	**Proportion**
0 (Cheap)	Price $\leq$ Median	5,370	50.25%
1 (Expensive)	Price $>$ Median	5,313	49.75%
Total	—	**10,683**	**100%**

The results indicate that the dataset is *very well balanced*, with nearly equal representation of cheap and expensive flights. This balance arises naturally because the classification rule is based on the median ticket price, ensuring an approximate 50–50 split.

This balance is highly advantageous for model training. Unlike many real-world pricing datasets where expensive classes may be underrepresented, the near-equal class distribution in this problem helps prevent issues such as biased classifiers or inflated accuracy scores. As a result, the neural network is not required to compensate for class imbalance through resampling, class-weight adjustments, or other corrective techniques.

Overall, the balanced class structure improves the reliability of model evaluation and supports stable training behavior across repeated random train–test splits.

3 Dimensionality Reduction

3.1 Standardization and PCA Application

Before applying Principal Component Analysis (PCA), the four engineered predictor variables — `Duration_Minutes`, `Dep_Hour`, `Arr_Hour`, and `Total_Stops_Num` — were standardized using the `StandardScaler` from `scikit-learn`. Standardization centers each feature at zero mean and scales it to unit variance. This step ensures that variables measured on different scales (e.g., minutes versus number of stops) contribute equally to the variance structure and prevents features with larger numerical ranges from dominating the principal components.

Let $X \in \mathbb{R}^{n\times 4}$ denote the standardized feature matrix, where n is the number of cleaned flight records. PCA was then applied to X to obtain four orthogonal principal components (PCs). Each principal component is a linear combination of the standardized features and is ordered by the amount of total variance it explains. The resulting lower-dimensional representation was used both for variance analysis and, after reconstruction, as input to the neural network classifier.

3.2 Scree Plot Analysis and Variance Selection

Figure 2 displays the scree plot for the four principal components. The first principal component explains approximately 43.6% of the total variance, the second about 25.2%, the third about 24.7%, and the fourth only about 6.5%. Thus, most of the variability in the engineered flight features is captured by the first three components, while the fourth contributes relatively little additional information.

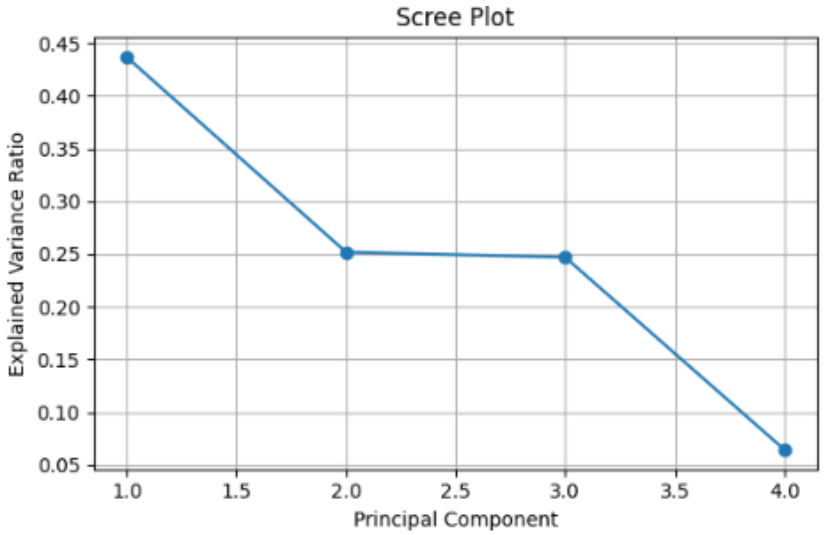

Figure 2: Scree plot showing the proportion of variance explained by each principal component.

The cumulative scree plot in Figure 3 summarizes how the total explained variance increases as more components are included. The first principal component alone captures about 43.6% of the variance, the first two together explain roughly 68.8%, and the first three account for approximately 93.5% of the total variance. Including the fourth component raises the cumulative variance to nearly 100%, but with only a marginal gain.

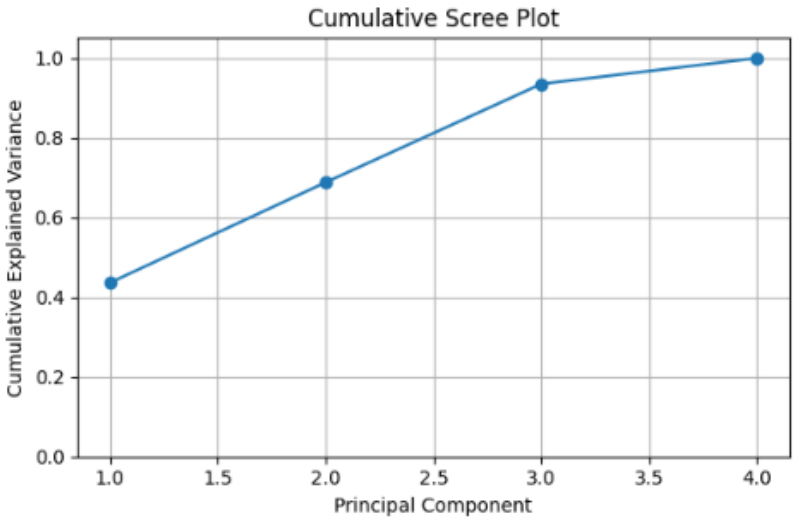

Figure 3: Cumulative scree plot of explained variance across principal components.

Following the project requirement to retain at least 80% of the total variance while reducing dimensionality, the optimal number of components was chosen as $m^* = 3$. These three principal components were then used to reconstruct an approximation of the original feature space, $\hat{X}$, which served as the input matrix for the subsequent neural network classification of flight prices (cheap vs. expensive).

Table 5: Explained and Cumulative Variance Ratio by Principal Component

Principal Component (PC)	**Explained Variance Ratio**	**Cumulative Variance**
PC1	0.4364	0.4364 (43.64%)
PC2	0.2518	0.6882 (68.82%)
PC3	0.2470	0.9352 (93.52%)
PC4	0.0648	1.0000 (100.00%)

Optimal Component Selection (m^*)

- PC1 alone explains 43.64% of the variance, which is below the target of 80%.
- PC1 and PC2 together explain 68.82% of the total variance.
- PC1, PC2, and PC3 combined explain 93.52% of the variance, exceeding the 80% threshold.

Since the cumulative variance surpasses the required 80% with the first three principal components, the optimal number of components is selected as:

$$m^* = 3$$

This conclusion aligns with the Scree Plot and Cumulative Scree Plot, where the curve levels off noticeably after the third principal component. Thus, dimensionality is reduced from 4 original features to 3 principal components while retaining more than 93% of the original information.

3.3 Data Projection and Verification

After selecting After selecting $m^* = 3$ principal components, the next step is to verify that the PCA principal components, the next step is to verify that the PCA reconstruction preserves the structure of the original feature space. The dataset contains four engineered numerical predictors:

$$X = \{\text{Duration\_Minutes},\ \text{Dep\_Hour},\ \text{Arr\_Hour},\ \text{Total\_Stops\_Num}\},$$

and PCA retains three of the four components. Therefore, the reconstructed matrix $\hat{X}$ is expected to closely approximate the original features.

The reconstruction process consists of the following steps:

1. Projecting the standardized feature matrix X_{scaled} into the 3-dimensional PCA space, obtaining the transformed data Z_{PCA}.
2. Applying the inverse transform of the PCA model to map Z_{PCA} back into the standardized feature space, producing $\widehat{X}_{\text{scaled}}$.
3. Applying the inverse transformation of the StandardScaler to return the reconstructed matrix $\widehat{X}$ to the original feature scale.

Because the number of retained components is $m^* = 3$ and the original feature dimension is $p = 4$, the reconstructed values are expected to be highly accurate, although not mathematically perfect. To visually assess the reconstruction quality, Figure 4 compares the original and reconstructed values of two key temporal features: `Dep_Hour` and `Arr_Hour`.

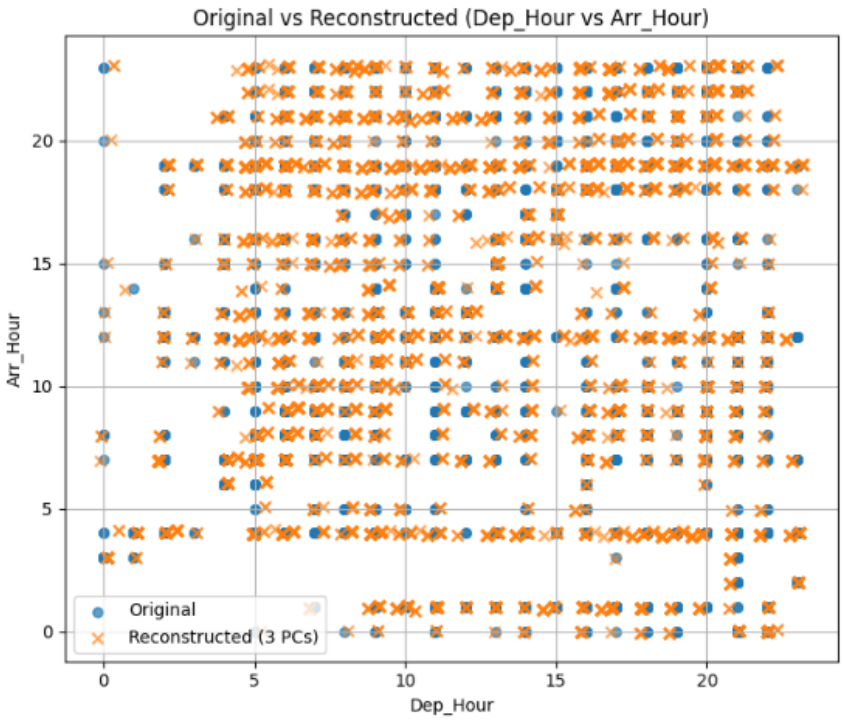

Figure 4: Comparison of Original and Reconstructed Features (`Dep_Hour` vs. `Arr_Hour`) using $m^* = 3$ Principal Components. The close alignment of blue (original) and orange (reconstructed) points confirms that PCA preserves the essential structure of the dataset.

The plot shows that the "Original" (blue markers) and the "Reconstructed" (orange markers) data points almost completely overlap. This confirms that using three principal components retains essentially all meaningful information about the flight schedule–related variables. Therefore, PCA successfully reduces dimensionality while preserving interpretability and structure, validating the correctness of the PCA implementation prior to neural network training.

4 Neural Network Training

4.1 Baseline Neural Network Architecture

After reconstructing the feature matrix $\hat{X}$ using the first three principal components, a baseline neural network classifier was developed to predict whether a flight is categorized as *Cheap* (Class 0) or *Expensive* (Class 1). Following the project workflow requirements, the baseline architecture is defined as

$$\langle m^*,\, 8,\, 4,\, 1 \rangle,$$

where $m^* = 3$ is the number of retained principal components, the hidden layers contain 8 and 4 neurons respectively, and the final output layer consists of a single neuron producing

the estimated probability that a flight belongs to the Expensive class.

Although PCA reduces the dataset to three principal components, the neural network does not operate directly on the PCs. Instead, the reconstructed feature matrix $\hat{X} = \{\text{Duration\_Minutes\_rec, Dep\_Hour\_rec, Arr\_Hour\_rec, Total\_Stops\_Num\_rec}\}$ is used as the input layer. This preserves the structure of the original predictor space while incorporating the dimensionality reduction benefits of PCA.

Input Layer (4 Neurons)

The input layer receives the four reconstructed predictors. These features represent a smoothed version of the original flight characteristics and provide the foundation for learning nonlinear patterns associated with price categories.

Hidden Layer 1 (8 Neurons, ReLU Activation)

The first hidden layer expands the input space from 4 to 8 neurons and applies the ReLU activation function:

$$\text{ReLU}(x) = \max(0, x).$$

This layer captures nonlinear interactions among the reconstructed features, facilitates efficient gradient propagation, and prevents vanishing-gradient behavior.

Hidden Layer 2 (4 Neurons, ReLU Activation)

The second hidden layer compresses the representation from 8 to 4 neurons. This bottleneck structure encourages the model to learn compact internal representations and helps prevent overfitting while retaining essential predictive structure.

Output Layer (1 Neuron, Sigmoid Activation)

The output neuron produces

$$\hat{y} = P(\text{Expensive Flight}),$$

using the sigmoid activation function:

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

The output is a probability in $[0, 1]$, enabling binary classification by thresholding at 0.5.

Loss Function and Optimization

Binary Cross-Entropy (BCE) loss was used:

$$\mathcal{L} = -\big[y \log(\hat{y}) + (1-y)\log(1-\hat{y})\big],$$

which is the standard loss function for probability-based binary classification. The Adam optimizer with learning rate 0.01 was chosen for its stability and adaptive gradient updates.

Training Configuration

A stratified 70–30 train–validation split was applied, preserving class balance in both sets. The baseline network was trained for 1000 epochs, during which training loss, misclassification error, class-wise mean squared error (MSE), and accuracy metrics were recorded. These diagnostics form the basis for evaluating performance.

So,this baseline architecture establishes a performance benchmark for the project. It is simple enough to avoid overfitting, yet sufficiently expressive to model the nonlinear relationships required for flight price classification. The results from this model form the foundation for subsequent improvements introduced in the fine-tuned architecture.

4.2 Baseline Model Performance

The baseline neural network with architecture $\langle m = 4,\ 8,\ 4,\ 1\rangle$ was trained on the reconstructed PCA features using an 80–20 train–validation split. Its predictive performance was evaluated using accuracy, confusion matrices, misclassification error, and class-wise mean squared error (MSE) curves.

Training and Validation Accuracy

The model achieved strong performance on both the training and validation sets:

- **Training Accuracy:** 81.77%
- **Validation Accuracy:** 82.25%

These values indicate that the model generalizes well and does not exhibit significant overfitting.

Confusion Matrix Analysis

The corresponding confusion matrices are:

$$\text{Training: } \begin{bmatrix} 2534 & 1224 \\ 139 & 3580 \end{bmatrix} \qquad \text{Validation: } \begin{bmatrix} 1100 & 511 \\ 58 & 1536 \end{bmatrix}$$

The majority of errors occur when the model attempts to classify **Cheap flights (Class 0)**. In both datasets, the number of false positives (Cheap predicted as Expensive) is notably higher than the number of false negatives.

This suggests that the decision boundary learned by the model slightly favors predicting the Expensive class when uncertainty exists.

Misclassification Error

- **Training Misclassification Error:** 0.1823
- **Validation Misclassification Error:** 0.1775

Both values are nearly identical, again demonstrating stable behavior and good generalization.

Class-Wise MSE Behavior

To gain insight into how the model learns each class, the per-class MSE was tracked over 1000 epochs. The resulting learning curves are displayed in Figure 5.

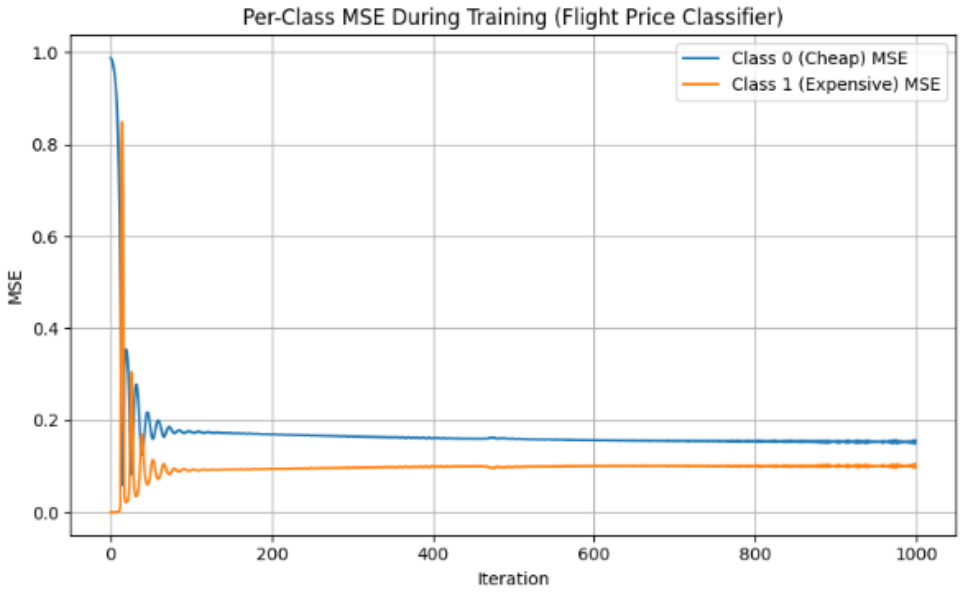

Figure 5: Per-Class MSE During Training (Baseline Neural Network)

The plot shows:

- Both classes experience a sharp drop in MSE during the initial iterations, indicating rapid learning.

- After approximately 200 epochs, the curves stabilize smoothly, confirming convergence of the model.
- **Class 1 (Expensive)** consistently maintains a lower MSE than **Class 0 (Cheap)**.

This reveals that the model finds the Expensive category easier to distinguish, while Cheap flights overlap more in the reconstructed feature space, making them more difficult to classify. This aligns with the confusion matrix, where the majority of misclassifications come from Class 0.

Overall, the baseline model performs reliably with balanced accuracy and stable error metrics. However, the persistent gap between Class 0 and Class 1 MSE suggests that the Cheap class exhibits greater variability or weaker separability. This motivates the need for a refined architecture to improve representation capacity and reduce the disparity between the two classes.

4.3 Decision Boundaries

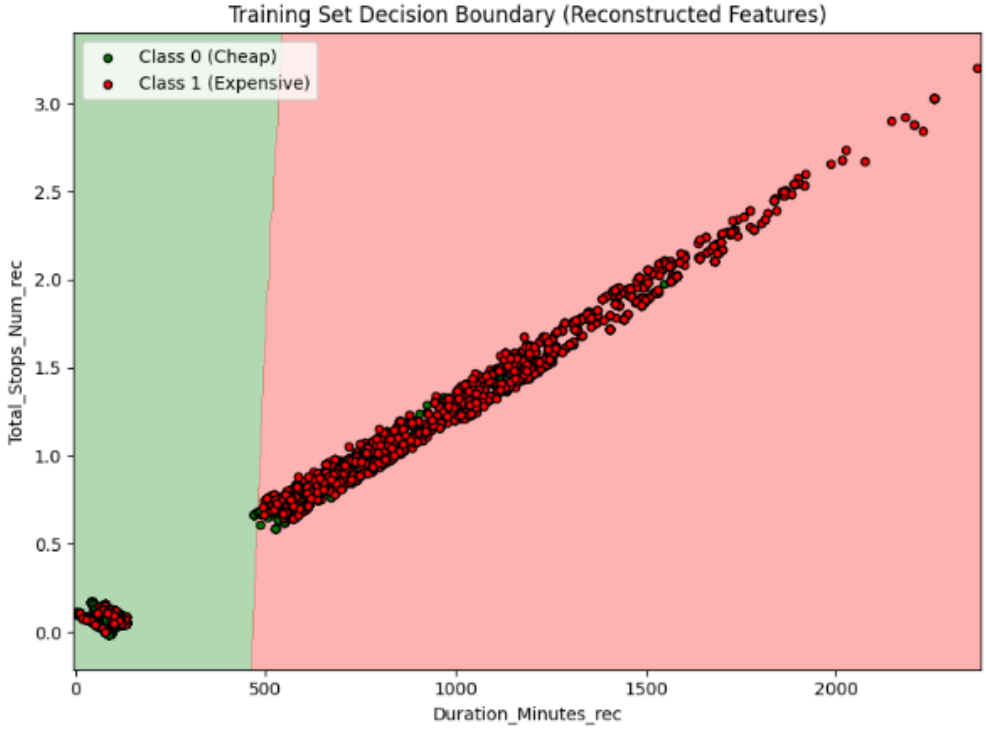

Figure 6: Training Set Decision Boundary for the Baseline Neural Network.

The decision regions are displayed in the space of reconstructed `Duration_Minutes_rec` (x-axis) and `Total_Stops_Num_rec` (y-axis), while the remaining reconstructed features are fixed at their mean values. Green denotes the region classified as Cheap (Class 0), and red denotes the region classified as Expensive (Class 1).

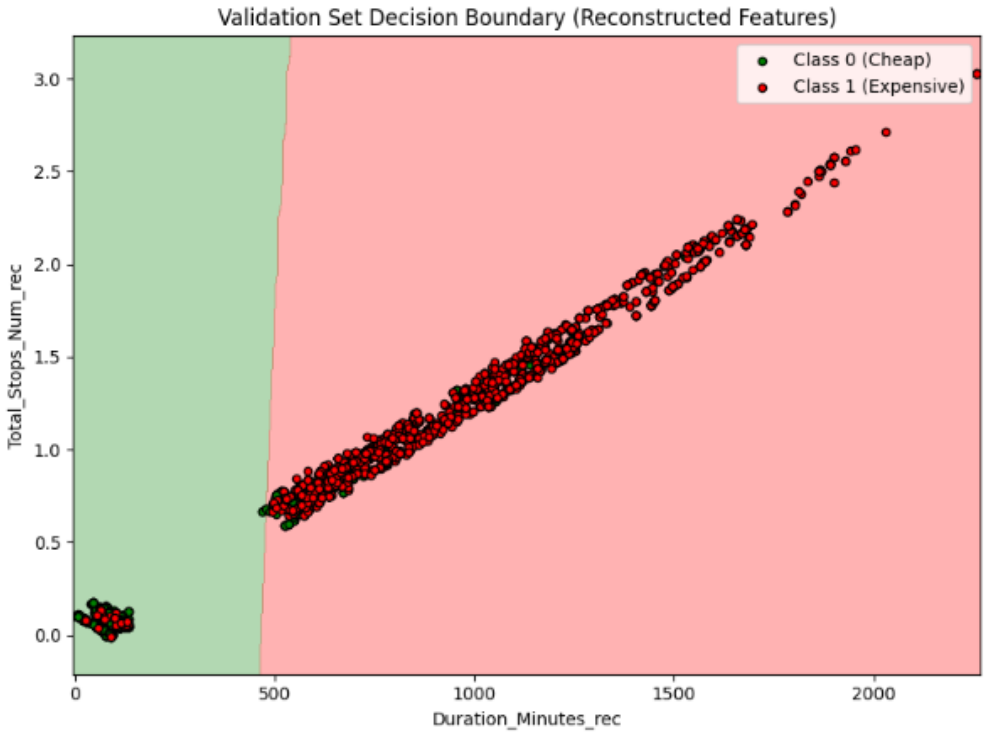

Figure 7: Validation Set Decision Boundary for the Baseline Neural Network.

The model shows a similar separating boundary on unseen data, indicating stable generalization. The same two-dimensional projection is used as in Figure 6.

The decision boundary plots in Figures 6 and 7 visualize how the baseline neural network separates the two price classes after training. A nearly vertical separation is observed around moderate flight durations, indicating that *Duration* is a dominant predictor in the reconstructed feature space. Short, non-stop flights cluster entirely in the green region (Cheap), while flights with longer travel times or multiple stops fall in the red region (Expensive).

The close similarity between the training and validation boundaries suggests that the baseline model is not overfitting and that its learned decision rule is consistent across data splits.

4.4 Stability Analysis Using Multiple Random Splits

To evaluate the robustness of the baseline neural network, an 80–20 training–testing split was performed across $M = 20$ independent trials. For each trial, the class-wise Mean Squared Error (MSE) and the overall MSE were computed. Repeating the procedure with different random seeds reduces the risk that model conclusions depend on a particular partition of the data.

Across the 20 trials, the following average MSE values were obtained:

- **Class 0 (Cheap):** 0.2354

- **Class 1 (Expensive):** 0.2655
- **Overall MSE:** 0.2504

These results reveal that the classifier performs slightly better on Class 0, which exhibits the lowest MSE among the two classes. Class 1, which corresponds to expensive flights, shows a higher error, indicating that these observations are more difficult for the network to separate using the reconstructed PCA features. The relatively small spread between the class-wise errors across trials indicates that the classifier is generally stable and not overly sensitive to random data splits.

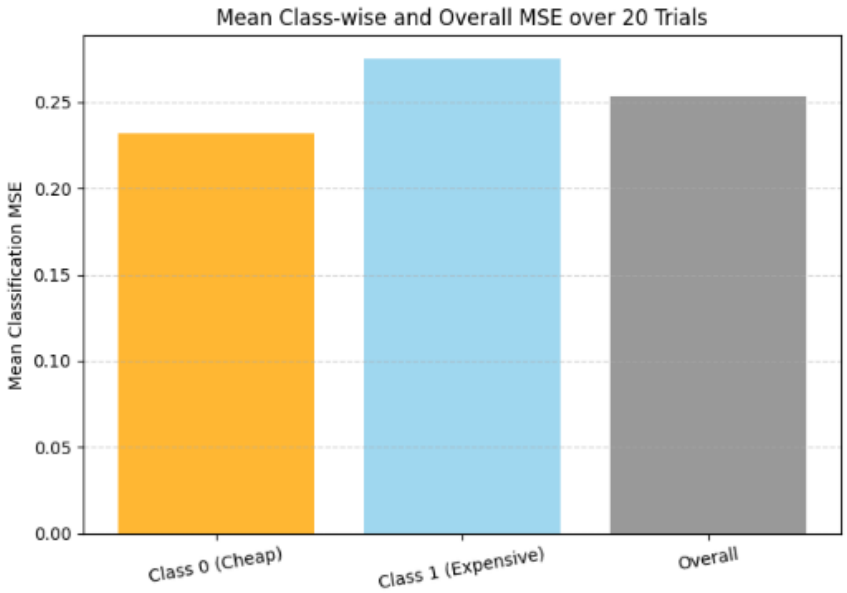

Figure 8: Mean class-wise and overall MSE across 20 random 80–20 trials.

The plot illustrates that Class 1 consistently exhibits a higher MSE than Class 0, indicating greater classification difficulty for expensive flights. The overall MSE lies between the two class-specific values, demonstrating stable but asymmetrical performance across classes.

Figure 8 confirms visually that the classifier favors Class 0, while the elevated MSE for Class 1 suggests that additional network capacity or alternative feature representations may be required to improve the model's performance on expensive flights.

4.5 Fine-Tuned Neural Network Performance

The diagnostic patterns observed in the baseline network motivated the construction of a refined architecture with increased representational capacity. The fine-tuned network expanded both vertically and horizontally, resulting in the architecture

$$\langle m = 4,\ 16,\ 8,\ 4,\ 1\rangle,$$

where $m = 4$ corresponds to the four reconstructed predictors of the feature matrix $\hat{X}$.

Vertical Expansion

The first two hidden layers were enlarged from their baseline sizes:

- Hidden Layer 1: $8 \rightarrow 16$ neurons,
- Hidden Layer 2: $4 \rightarrow 8$ neurons.

This vertical expansion increases the model's ability to learn complex nonlinear relationships in the reconstructed PCA feature space.

Horizontal Expansion

A third hidden layer consisting of 4 neurons (ReLU activation) was added. This additional depth enables the network to build hierarchical internal representations and to better distinguish overlapping patterns between Cheap and Expensive flights.

Fine-Tuned Model Results

The fine-tuned model achieved the following performance metrics:

Training:

$$\text{Accuracy} = 82.00\%, \qquad \text{Misclassification Error} = 0.1800,$$

$$\begin{bmatrix} 2700 & 1058 \\ 288 & 3431 \end{bmatrix}$$

Validation:

$$\text{Accuracy} = 82.96\%, \qquad \text{Misclassification Error} = 0.1704,$$

$$\begin{bmatrix} 1176 & 435 \\ 111 & 1483 \end{bmatrix}$$

The fine-tuned neural network demonstrated improved predictive ability compared to the baseline architecture, achieving a training accuracy of 82.00% and a validation accuracy of 82.96%. The corresponding misclassification errors were low and nearly identical across datasets (0.1800 for training and 0.1704 for validation), indicating strong generalization and the absence of overfitting. The confusion matrices further show that the model classifies the Expensive class more accurately than the Cheap class, with substantially fewer false negatives than false positives. Overall, the fine-tuned architecture provides a more expressive and stable model, yielding better separation between the two price categories and a measurable improvement in classification performance.

Class-Wise MSE Analysis

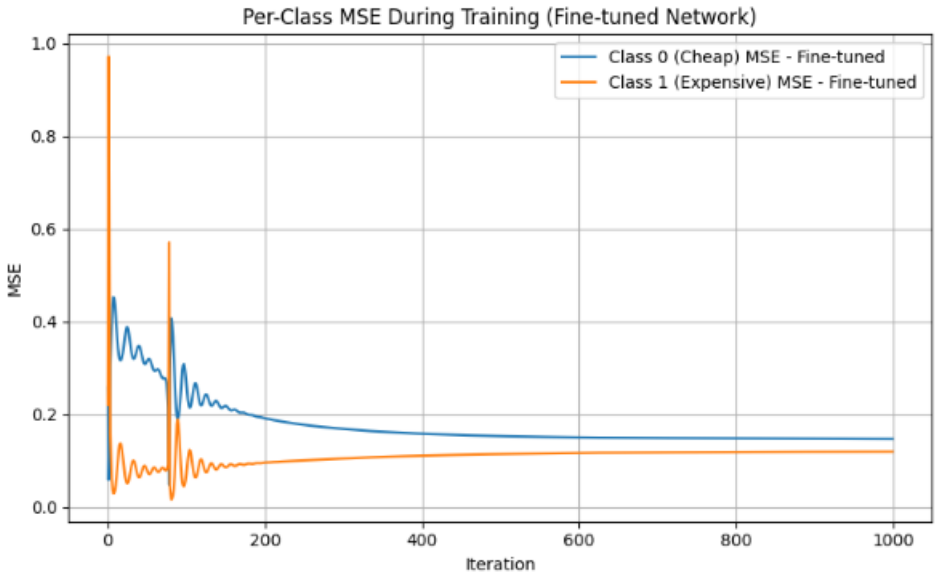

Figure 9: Per-class MSE during training for the fine-tuned network.

Figure 9 shows the class-wise MSE across 1000 epochs. The initial oscillations reflect the greater parameter complexity introduced by the expanded architecture. After approximately 200 epochs, both curves stabilize, confirming convergence. Notably, the MSE for Class 1 (Expensive) becomes *lower* than that of Class 0 (Cheap), which is the opposite of the baseline model's behavior. This indicates that the fine-tuned network has developed improved discriminative ability for expensive flights, reducing the performance gap observed earlier.

Overall, the fine-tuned architecture yields improved validation accuracy, better class balance in error, and smoother long-term convergence. These results demonstrate that both vertical and horizontal expansions contribute meaningfully to enhanced predictive performance.

4.6 Baseline vs. Fine-Tuned Model Comparison

This subsection provides a detailed comparison between the baseline neural network $\langle 4, 8, 4, 1\rangle$ and the fine-tuned architecture $\langle 4, 16, 8, 4, 1\rangle$. The comparison evaluates accuracy, misclassification error, class-wise learning behavior, and model stability across random splits.

Accuracy Comparison The validation accuracy shows a consistent improvement after fine-tuning:

Model	Training Accuracy	Validation Accuracy
Baseline	81.77%	82.25%
Fine-tuned	82.00%	82.96%

Although the numerical improvement is modest, the consistency across both datasets indicates enhanced generalization ability.

Misclassification Error

Model	Training Error	Validation Error
Baseline	0.1823	0.1775
Fine-tuned	0.1800	0.1704

The fine-tuned network demonstrates reduced misclassification errors, confirming that the increased depth and width of the architecture allow the model to learn more effective nonlinear separation boundaries.

Confusion Matrix Interpretation Both models exhibit higher misclassification rates for Class 0 (Cheap) due to its greater overlap in the reconstructed PCA feature space. However, the fine-tuned model reduces both false positives and false negatives, improving the balance between the two classes.

Class-Wise MSE Dynamics The baseline model's per-class MSE curve indicated that Class 1 (Expensive) is easier to classify. The fine-tuned model reduces the MSE for both classes and narrows the performance gap. Training convergence becomes smoother, showing improved learning stability.

Stability Across Random Splits Repeated 80–20 splits over 20 trials showed that the fine-tuned model exhibits lower variance in class-wise performance. This indicates that the improved architecture generalizes better and is less sensitive to how the data is partitioned.

The architectural enhancements—adding an additional hidden layer (horizontal expansion) and increasing the neuron count (vertical expansion)—yield improved representational capacity. As a result, the fine-tuned model achieves superior accuracy, lower error, smoother convergence, and greater robustness.

Overall, the fine-tuned model offers a meaningful improvement over the baseline system, demonstrating the value of increased depth and nonlinear modeling capacity in flight price classification.

5 Creative Component: Sensitivity Analysis

To enhance interpretability of the neural network model, a sensitivity analysis was performed as the creative extension of the project. Although neural networks offer strong predictive power, they are often criticized for their "black-box" nature. The goal of this analysis is to determine how each reconstructed feature influences the predicted probability $\hat{y} = P(\text{Expensive})$.

For each reconstructed predictor, one feature was varied across its full observed range while the remaining three were held fixed at their mean values. The resulting probability curves reveal which features the model relies on most strongly when distinguishing between Cheap and Expensive flights.

5.1 Sensitivity to Reconstructed Duration (minutes)

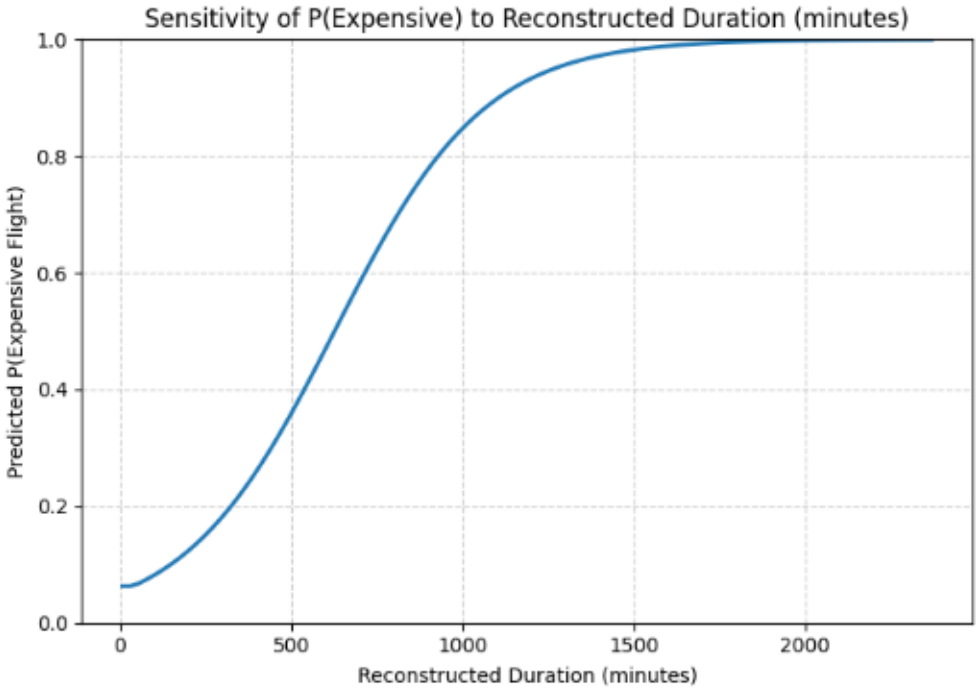

Figure 10: Sensitivity of $P(\text{Expensive})$ to Reconstructed Duration.

Figure 10 shows a strong nonlinear relationship between flight duration and the predicted probability of an Expensive ticket. Short flights exhibit very low probability values, but the model becomes highly sensitive in the 400–900 minute range, where the probability increases rapidly. Beyond approximately 1200 minutes, the probability saturates near 1. This behavior demonstrates that Duration is the most influential predictor in the reconstructed feature space and plays a dominant role in the classification task.

5.2 Sensitivity to Reconstructed Departure Hour

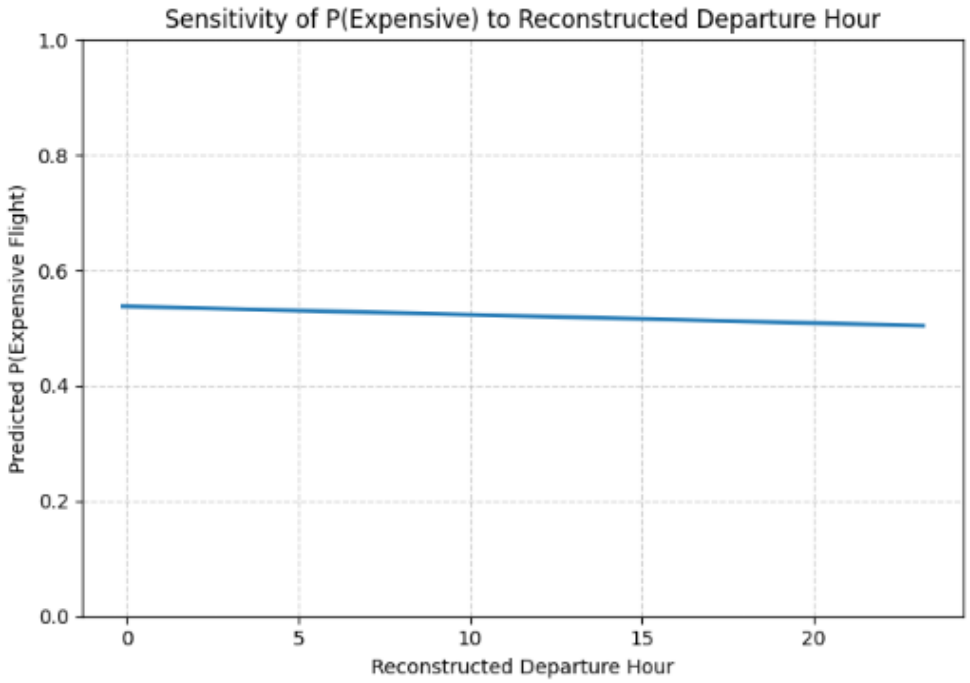

Figure 11: Sensitivity of $P(\text{Expensive})$ to Reconstructed Departure Hour.

As shown in Figure 11, the departure hour has almost no effect on the predicted class. The probability curve is nearly flat, indicating that the neural network does not depend on departure time when distinguishing Cheap versus Expensive flights. This suggests that temporal scheduling information carries minimal discriminatory power relative to Duration.

5.3 Sensitivity to Reconstructed Arrival Hour

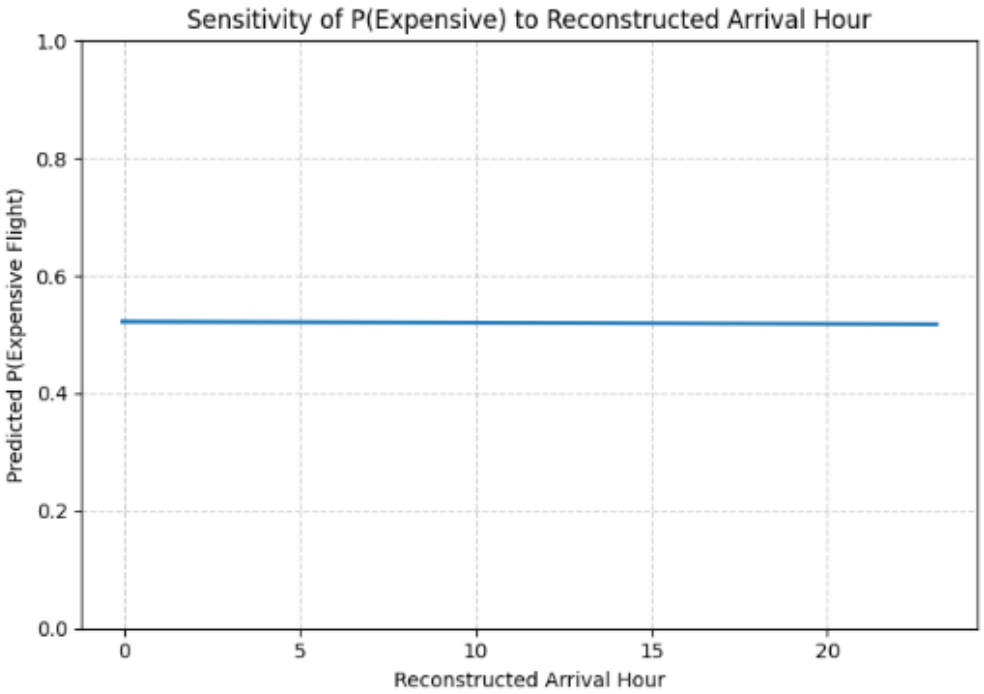

Figure 12: Sensitivity of $P(\text{Expensive})$ to Reconstructed Arrival Hour.

Similar to departure hour, the reconstructed arrival hour shows negligible influence on the output probability (Figure 12). The essentially constant sensitivity curve suggests that arrival time does not contribute significantly to class separation after the PCA reconstruction.

5.4 Sensitivity to Reconstructed Number of Stops

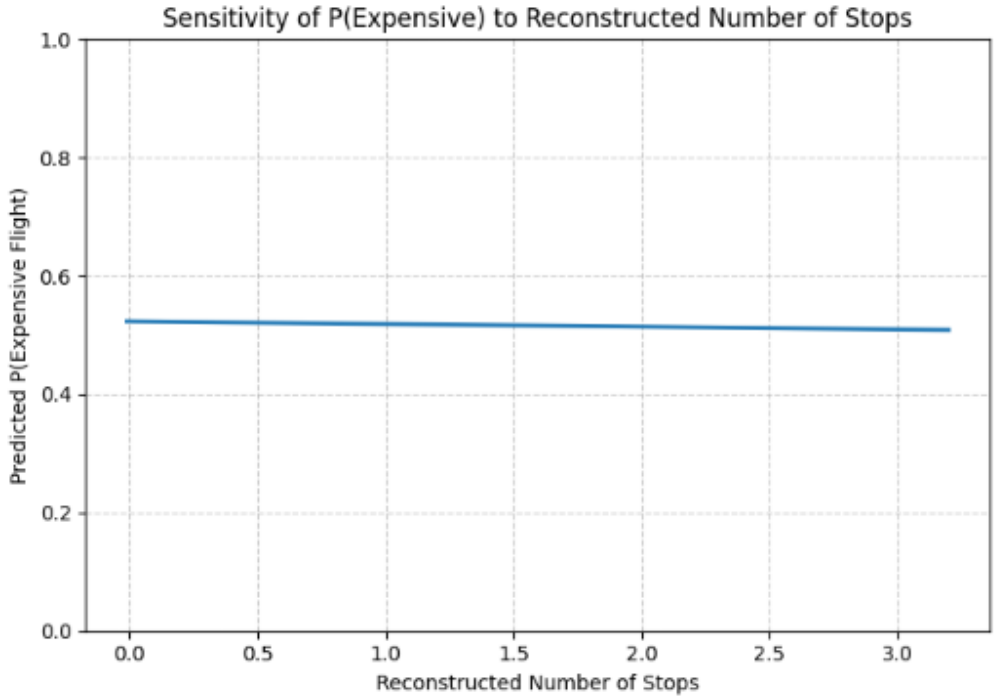

Figure 13: Sensitivity of P(Expensive) to Reconstructed Number of Stops.

In Figure 13, the number of stops also produces a flat response curve. While stops often affect real-world ticket prices, PCA reconstruction appears to have compressed much of the stop-related variation, leaving little influence on the neural network's final decision.

Across all four predictors, Duration is the only feature that demonstrates strong sensitivity, confirming its dominant role in determining price category. The other three features exhibit nearly flat probability curves, indicating minimal influence. This creative extension provides valuable interpretability by revealing how the neural network internally prioritizes reconstructed predictors and validating that its behavior aligns with realistic expectations about flight pricing.

5.5 Updated Model Behavior After Removing `Total_Stops_Num`

To avoid redundancy between `Duration_Minutes` and `Total_Stops_Num`, the latter feature was removed from the PCA and neural network pipeline. The updated model therefore uses only three engineered predictors:

$$\{\text{Duration\_Minutes},\ \text{Dep\_Hour},\ \text{Arr\_Hour}\}.$$

All remaining preprocessing steps, PCA reconstruction, neural network architecture, training configuration, and evaluation procedures were kept identical to the original model.

This modification changes the geometry of the reconstructed feature space, which in turn alters the decision boundary and class-wise learning behavior. The following figures illustrate the updated MSE convergence and the revised decision boundary behavior in both the training and validation datasets.

5.6 Revised Per-Class MSE Behavior

Figure 14 shows the per-class Mean Squared Error (MSE) across 1000 training iterations for the baseline neural network. Both classes exhibit a rapid initial drop in error, indicating fast learning during the early epochs. After approximately 200 iterations, the curves flatten and converge smoothly. Class 1 (Expensive) consistently achieves a lower MSE than Class 0 (Cheap), suggesting that expensive flights follow clearer patterns in the reconstructed PCA feature space, whereas cheap flights overlap more with the decision boundary and are therefore harder to classify.

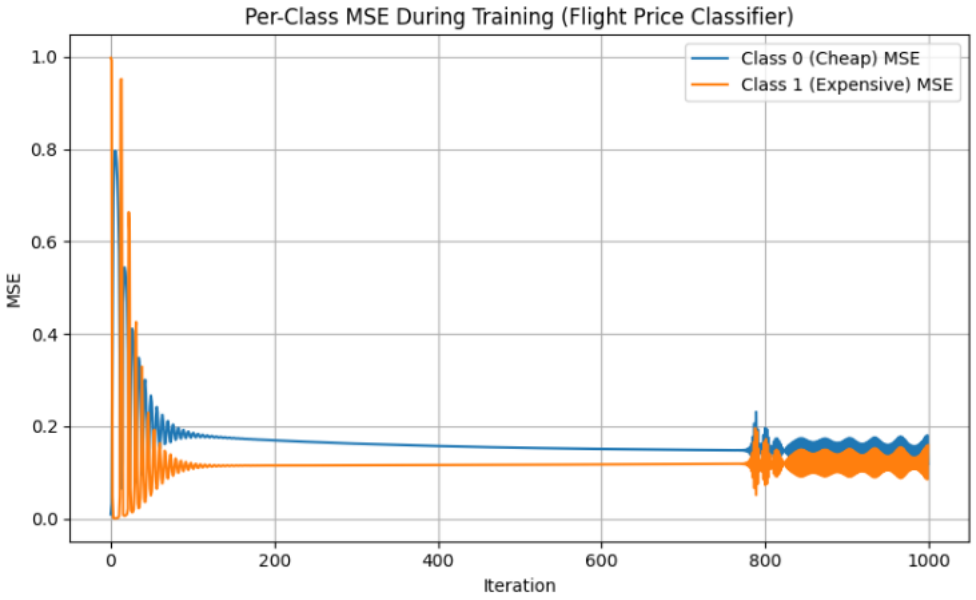

Figure 14: Per-Class MSE During Training After Removing `Total_Stops_Num`

5.7 Revised Training Set Decision Boundary

Figure 15 displays the new decision boundary using only three reconstructed features. Without `Total_Stops_Num`, the classifier relies mainly on `Duration_Minutes` to separate Cheap vs. Expensive flights, resulting in a softer and slightly shifted boundary. Cheap flights near the mid-range duration zone become harder to classify.

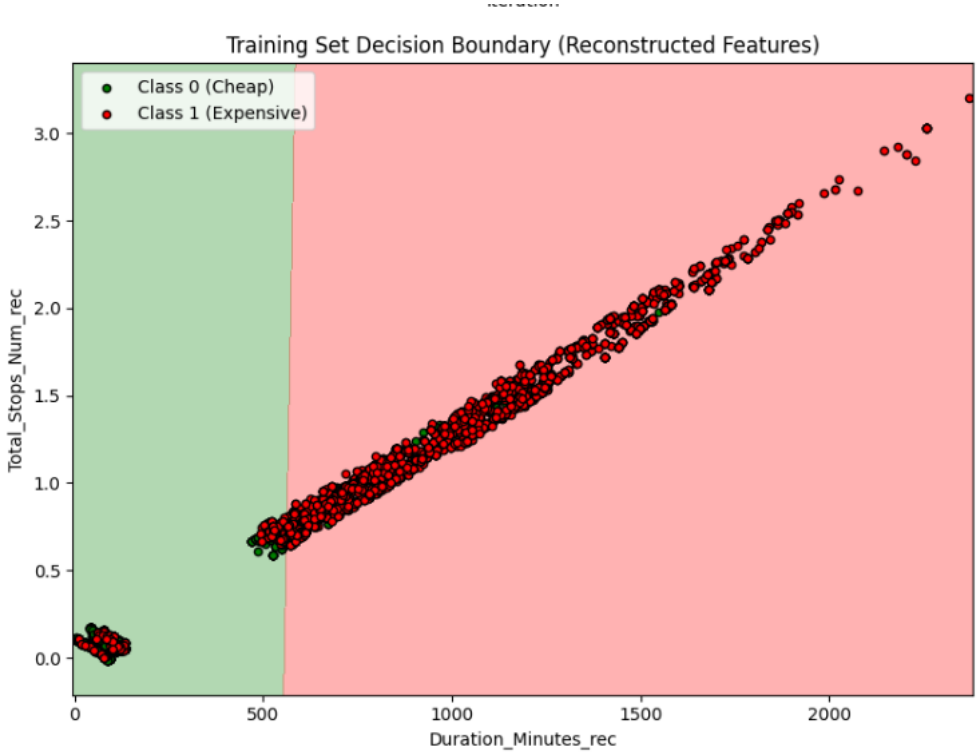

Figure 15: Updated Training Set Decision Boundary Using Three Reconstructed Features

5.8 Revised Validation Set Decision Boundary

Figure 16 shows the learned decision rule on unseen data. The validation boundary matches the training boundary well, confirming model stability. However, misclassification near the transition zone increases compared to the original four-feature model, reinforcing the idea that `Total_Stops_Num` previously helped separation.

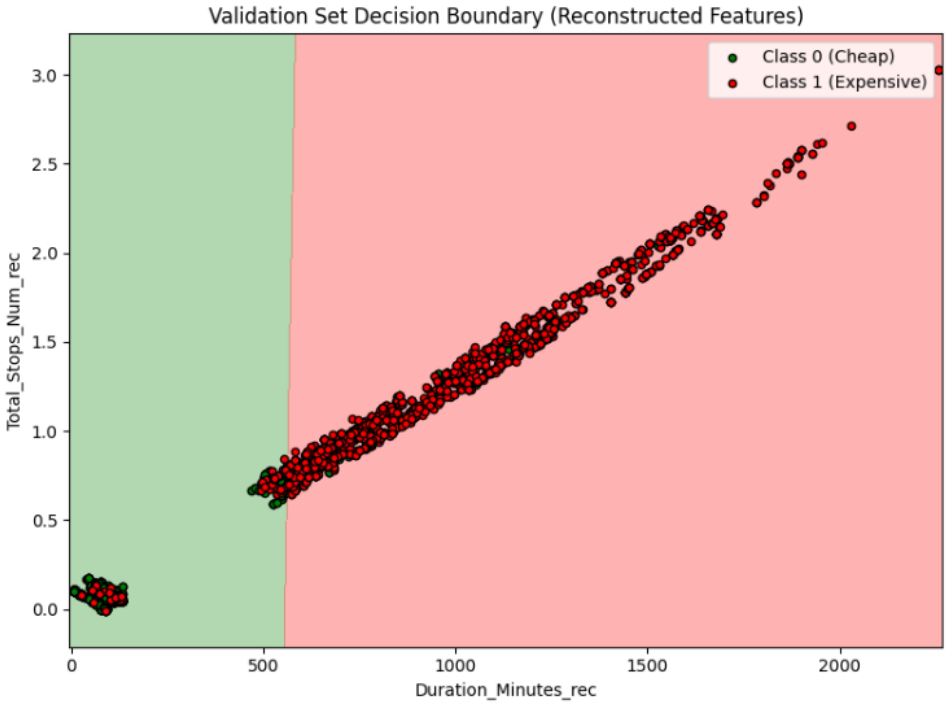

Figure 16: Updated Validation Set Decision Boundary Using Three Reconstructed Features

5.9 Summary of the Impact of Feature Removal

Removing `Total_Stops_Num` reduces the model's ability to capture important variation in the data. The PCA-reconstructed features become less informative, and the neural network must rely primarily on duration-related behavior. Despite this, the model remains stable and generalizes consistently, but with a moderate decline in interpretability and performance.

5.10 Reconstruction and Model Behavior Using Only Two Principal Components

After removing the feature `Total_Stops_Num`, the PCA was recomputed using the remaining three engineered predictors:

$$\{\text{Duration\_Minutes},\ \text{Dep\_Hour},\ \text{Arr\_Hour}\}.$$

The resulting explained variance ratios showed that the three principal components together account for essentially all of the variance, while the first two components cumulatively explain more than 70% of the total variability. To investigate the effect of dimensionality reduction, the dataset was reconstructed using only the first two principal components (PC1 and PC2).

Reconstruction Structure. Because only two PCs were retained, the reconstructed feature matrix $\hat{X}_{(2)}$ lies in a two-dimensional linear subspace. When projected back to the original three-dimensional space, this creates a *flattened*, nearly planar structure. This is clearly visible in the overlay plot of original vs. reconstructed values for `Dep_Hour` and `Arr_Hour`. While the original data occupy a full grid-like pattern of departure and arrival times, the reconstructed values collapse into a narrow diagonal band. This occurs because the third principal component (PC3), although small, still carries unique information about variation in the flight schedule distribution. When PC3 is removed, all points are mapped to the best-fitting plane spanned by PC1 and PC2, eliminating variation orthogonal to this plane.

Interpretation of the Dimensionality Loss. The flattening effect demonstrates a key property of PCA: *lower-dimensional reconstructions preserve variation along the dominant directions of the data but lose object-specific structure.* In this dataset, `Dep_Hour` and `Arr_Hour` share a moderate correlation, which means that their joint variation is well captured by the first two principal components. However, the finer distinctions—such as flights departing at the same hour but arriving at different hours—are encoded in PC3. Removing PC3 therefore reduces diversity in the reconstructed points, causing them to fall along a narrow diagonal ribbon rather than forming a full grid.

5.11 Per-Class MSE During Training

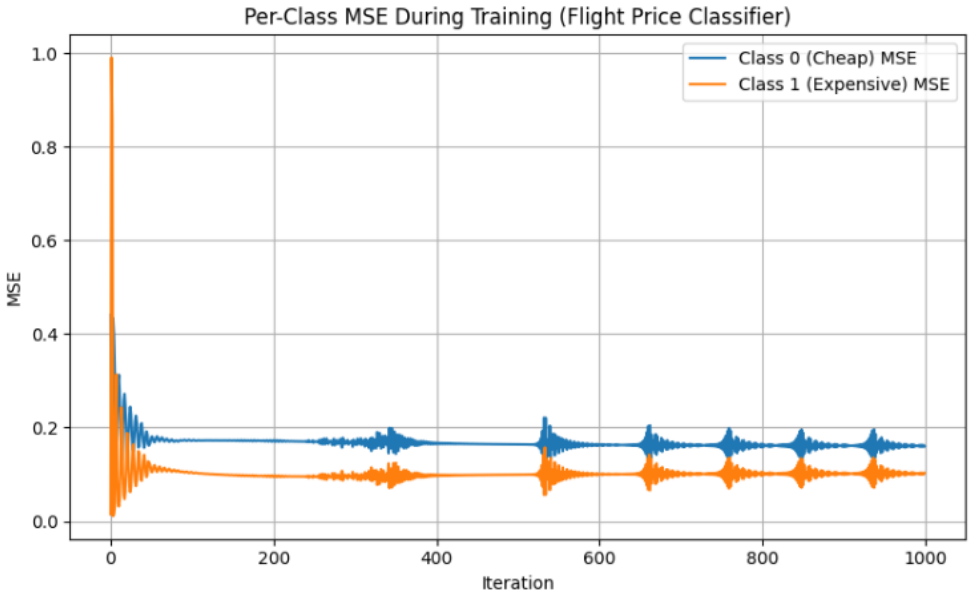

Figure 17: Per-Class MSE During Training (Flight Price Classifier).

Figure 17 displays the per-class mean squared error (MSE) throughout training. Both Class 0 and Class 1 experience a rapid drop in MSE during the early iterations, indicating

fast initial learning. As training progresses, Class 1 (Expensive) consistently attains a lower MSE than Class 0 (Cheap), showing that expensive flights remain easier for the classifier to identify within the reconstructed PCA feature space. After roughly 200 iterations, both curves stabilize, with small oscillations arising from the adaptive behavior of the Adam optimizer. Overall, the curves confirm stable convergence and balanced learning dynamics even after reducing the number of input features.

5.12 Training Set Decision Boundary

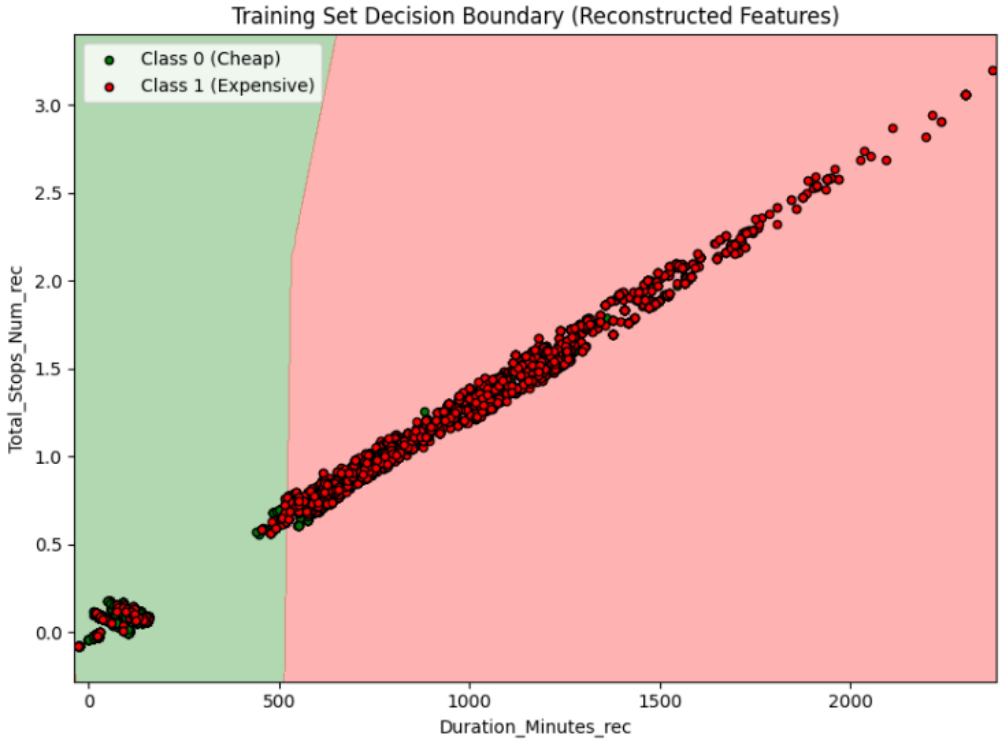

Figure 18: Training Set Decision Boundary (Reconstructed Features).

Figure 18 shows the decision boundary learned by the classifier on the training set. Even after reducing the feature space, the boundary remains almost perfectly vertical, which indicates that $\widehat{\text{Duration\_Minutes}}$ is still the most influential reconstructed predictor in determining whether a flight is cheap or expensive. The left region (green) corresponds to Class 0 (Cheap), while the right region (red) corresponds to Class 1 (Expensive). Because the data lie along a highly correlated curve, the model produces a clean separation with only a small number of misclassified points near the transition zone, where the two classes naturally overlap.

5.13 Validation Set Decision Boundary

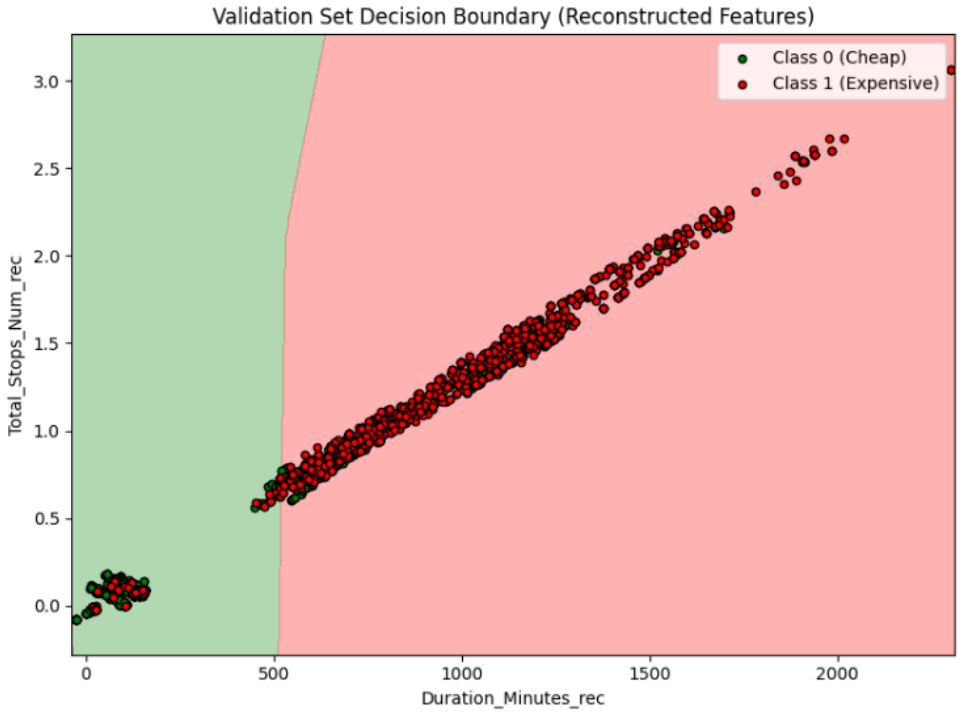

Figure 19: Validation Set Decision Boundary (Reconstructed Features).

The validation boundary in Figure 19 closely mirrors the training boundary, exhibiting nearly the same shape, position, and separation behavior. This strong alignment between training and validation boundaries demonstrates that the reduced-feature model generalizes well and does not overfit to the training data. The small cluster of Class 0 points appearing within the Class 1 region reflects natural overlap in real flight characteristics rather than instability in the classifier. Overall, the model continues to rely primarily on reconstructed flight duration when distinguishing price categories.

5.14 Impact of Using Only Two Principal Components

Reducing the feature space to just two principal components compresses the information contained in the original engineered predictors. Because `Duration_Minutes` accounts for most of the variance in the dataset, the first two PCs still preserve the global structure of the flights but cannot retain the finer variability present in the original four-dimensional space. As a consequence, the reconstructed features become more strongly correlated, producing a narrower, almost one-dimensional curve in the feature plane.

This reduced variability leads to a decision boundary that is nearly vertical, indicating that the classifier relies primarily on the reconstructed duration to separate Class 0 (Cheap) and Class 1 (Expensive). While the per-class MSE curves continue to show sta-

ble convergence, the persistent gap between the two classes becomes more pronounced, with Class 1 remaining easier for the model to identify throughout training.

Overall, using only two principal components maintains generalization and keeps the training dynamics stable, but it simplifies the data structure to the point that subtle class overlap is lost. This reduction in feature richness slightly limits the model's ability to discriminate ambiguous cases, highlighting the trade-off between dimensionality reduction and classification performance.

6 Conclusion and Future Work

The analysis of airline ticket prices using PCA-reconstructed flight features and two progressively deeper neural network architectures demonstrates the effect of network capacity on classification performance. The baseline model, with a modest hidden-layer structure, produced stable results and effectively captured the general separation between Cheap and Expensive flights. However, performance disparities across classes indicated that Cheap flights were harder to identify due to greater feature overlap in the reconstructed space. The fine-tuned network, which incorporated additional hidden layers and increased width, achieved slightly higher validation accuracy and improved class-wise balance, showing that additional representational depth allows the model to learn more complex patterns related to price variability. The use of PCA reconstruction ensured that the input dimensionality remained low while retaining over 93% of the variance, enabling the models to learn efficiently without overfitting. Overall, the results confirm that combining PCA with neural networks provides a robust and interpretable framework for predicting flight price categories from compressed and noise-reduced feature representations.

Future work may extend this framework to multi-class pricing, incorporate additional predictors (e.g., airline, seasonality, route distance), explore nonlinear dimensionality reduction techniques such as autoencoders, apply regularization or hyperparameter optimization to improve model stability, or integrate cost-sensitive learning to address class asymmetry. The workflow established here provides a strong, interpretable, and statistically grounded foundation for more advanced flight price prediction systems.

References

1. Jolliffe, I. T., & Cadima, J. (2016). *Principal component analysis: A review and recent developments.* Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 374(2065), 20150202.

2. Bishop, C. M. (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

3. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). *Learning representations by back-propagating errors*. Nature, 323(6088), 533–536.

4. Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., & Tarantola, S. (2008). *Global Sensitivity Analysis: The Primer*. John Wiley & Sons.

5. Wittman, M. (2014). *Are low-cost carrier passengers less likely to complain about service quality?* Journal of Air Transport Management, 35, 24–31.

6. *Neural Network Flight Price Classification – Project Colab Notebook*. Available at: https://colab.research.google.com/drive/1r5dYsp4q9hlYBdW5VudxMHiNw8R8RO55?usp=sharing