This report presents the development and evaluation of a neural network model designed to predict whether a customer will accept a travel-related product offer. The dataset, sourced from `data/class5.csv`, contains 3,208 observations with multiple demographic, behavioral, and interaction-related features. The target variable, **ProdTaken**, is binary, where 1 indicates that the product was taken and 0 indicates it was not. For modeling purposes, all columns except `ProdTaken` were treated as input features (X), while `ProdTaken` was used as the label (y).

To prepare the data for modeling, the dataset was divided into training and testing sets using an 80–20 split. The `train_test_split` function was applied with `test_size = 0.2`, `random_state = 42` to ensure reproducibility, and `stratify = y` to preserve the original class distribution in both sets. This stratification is particularly important because the dataset is moderately imbalanced, with approximately 19% of observations belonging to the positive class. Maintaining the same proportion in training and testing prevents misleading evaluation results.

Categorical variables were automatically identified using `X.select_dtypes(include=["object"])` and converted into numerical format through one-hot encoding using `pd.get_dummies(..., drop_first=True)`. This transformation converts each category into binary indicator variables while removing one reference category to prevent multicollinearity. One-hot encoding avoids imposing artificial ordinal relationships between categories, which is especially important for variables such as product type or contact type. After encoding, the dataset consisted entirely of numerical features suitable for neural network input.

To improve training stability and convergence, feature scaling was performed using a StandardScaler. The scaler was fit on the training data and then applied to both the training and testing datasets. This transformation standardizes each feature to have approximately zero mean and unit variance. Neural networks are sensitive to differences in feature scale, and standardization ensures that all inputs contribute proportionally during optimization.

An examination of the dataset reveals that it contains demographic variables such as age, gender, marital status, number of children visiting, car ownership, and monthly income; interaction-related variables such as type of contact, duration of pitch, number of follow-ups, and pitch satisfaction score; and travel preference variables including city tier, product pitched, preferred property star rating, number of trips, passport status, and designation. The class distribution shows 2,589 instances of customers who did not take the product and 619 who did, confirming a moderate imbalance toward the negative class. The data appears generally clean, with sensible numeric ranges and no major missing values. Minor categorical inconsistencies are automatically handled through one-hot encoding.

All available non-target features were included in the model to allow the neural network to automatically learn feature importance rather than manually selecting variables. Continuous variables were standardized, and categorical variables were one-hot encoded, resulting in a

higher-dimensional but purely numerical feature matrix. This structure is well-suited for a feed-forward neural network, which can learn complex nonlinear relationships among features.

The predictive model was implemented as a feed-forward neural network using `tf.keras.Sequential`. The input layer corresponds to the number of encoded features. The first hidden layer contains 64 neurons with ReLU activation, followed by a dropout layer with a rate of 0.3 to reduce overfitting. A second hidden layer with 32 neurons and ReLU activation was added, followed by another dropout layer with the same rate. The decreasing layer size from 64 to 32 encourages the network to learn compressed internal representations. The output layer consists of a single neuron with sigmoid activation, producing a probability between 0 and 1 for the positive class.

The model was trained using binary cross-entropy loss and the Adam optimizer, which provides adaptive learning rates and robust performance on tabular data. Performance metrics included accuracy and area under the ROC curve (AUC). Training was conducted for up to 100 epochs with a batch size of 32, and 20% of the training data was reserved for validation. Early stopping was applied with patience set to 10 epochs, monitoring validation loss and restoring the best weights to prevent overfitting. Random seeds were set to ensure reproducibility, and the trained model was saved both with a timestamped filename and a fixed filename for convenient reuse.

Evaluation on the test set shows strong overall performance. The model achieved an accuracy of approximately 93.77%. For the positive class, precision was 89.31%, recall was 76.90%, and the F1-score was 0.8264. Specificity for the negative class reached 97.80%. The confusion matrix indicates 476 true positives, 2,532 true negatives, 57 false positives, and 143 false negatives. These results demonstrate that the model is highly effective at identifying customers who will not take the product, while also performing reasonably well in detecting those who will. The relatively high precision suggests that when the model predicts a customer will take the product, it is usually correct. However, recall below 80% indicates that some potential buyers are still missed.

From a business perspective, the balance between precision and recall can be adjusted depending on strategic priorities. If missing potential buyers is costly, the classification threshold can be lowered to increase recall at the expense of more false positives. Conversely, if contacting uninterested customers incurs high cost, maintaining higher precision may be preferable.

In conclusion, the neural network model demonstrates strong predictive capability on a moderately imbalanced marketing dataset. Through appropriate preprocessing, encoding, scaling, regularization, and early stopping, the model achieves high accuracy and solid classification performance. With further threshold tuning or class-balancing techniques, the model could be further optimized for specific business objectives.