Fitness Club Booking Attendance Prediction

Introduction:

Fitness is an integral part of our lives, promoting physical well-being, mental health, and overall quality of life. In an era where health and wellness take center stage, fitness activities, such as group exercise classes and personal training sessions, have become increasingly popular. The ability to predict attendance at these fitness bookings can be a valuable tool for both fitness enthusiasts and fitness center operators.

In this documentation, we delve into a project aimed at predicting attendance at fitness bookings. To achieve this, we utilize a dataset containing a wealth of information about individuals' fitness habits, including their membership duration, weight, the number of days before booking, the day of the week, time of day, fitness category, and whether they attended the session. This dataset provides a valuable resource for building a predictive model that can help answer questions such as, "What factors influence an individual's decision to attend a fitness class?" and "Can we anticipate attendance patterns based on certain attributes?".

Objective of this project:

The main objective of our fitness booking attendance prediction project is outlined as follows:

- **Scope of Project:** The primary objective of this project is to develop a predictive model capable of categorizing individuals into one of two classes: "Attended" or "Not Attended". This binary classification simplifies the task of predicting attendance at fitness bookings.
- **Attribute Set:** The dataset used for this project encompasses a wide range of attributes related to fitness enthusiasts and their booking habits. These include 'months_as_member', 'weight', 'days_before', 'day_of_week', 'time', and 'category'. The selection of these features is motivated by the belief that they contain valuable information for predicting fitness booking attendance.
- **Data Procurement:** The dataset used in this project is originated from **Kaggle**. The dataset utilized in this project has been collected and curated for fitness-related analysis. It serves as the foundational source of information for our predictive modeling efforts, providing real-world insights into factors influencing fitness class attendance.
- **Modeling Techniques:** Throughout this project, we will explore and implement a variety of deep learning techniques and algorithms to construct a robust predictive model. Our selection of methods will be driven by our pursuit of identifying the most

effective approach for predicting fitness booking attendance. Deep learning, known for its ability to capture intricate patterns and relationships in data, offers a promising avenue to gain valuable insights into attendance patterns.

- **Data Preprocessing and Analysis:** Data preprocessing tasks will involve addressing missing values, encoding categorical variables (e.g. 'day_of_week' and 'time'), and scaling features like 'weight' and 'days_before'. Additionally, we will conduct data analysis to gain insights into the dataset's characteristics, distributions, and correlations among features.
- **Evaluation Metrics**: Model performance will be assessed using standard binary classification metrics, including Accuracy, F1 score, Precision, Recall, and Confusion Matrix. These metrics will help us gauge the model's effectiveness in predicting fitness booking attendance, providing a clear understanding of its strengths and limitations.

Data Procurement:

The dataset was procured from Kaggle, a popular platform for data science, machine learning and deep learning datasets.

Dataset Name: Fitness Club Booking Dataset Classification

Dataset Link: https://www.kaggle.com/datasets/ddosad/datacamps-data-science-associate-certification

Dataset Features:

- The Fitness club dataset contains Totally 7 features along with Attendance as class label.

1. Booking_id:

- This is a unique identifier for each fitness booking. It serves as a primary key for distinguishing individual bookings within the dataset.
- It is used for tracking and referencing specific fitness bookings.

2. Months_as_member:

- This feature represents the number of months an individual has been a member of the fitness center. It indicates the duration of the membership for each attendee.
- The length of membership can be an important factor in predicting attendance, as long-term members may exhibit different booking behaviors compared to new members.

3. Weight:

- The 'weight' attribute signifies the weight of the attendee, measured in a specific unit (e.g., kilograms or pounds). Weight is often considered a critical health and fitness metric.

- It can be used to explore whether an attendee's weight affects their likelihood of attending fitness sessions.

4. Days_before:

- This attribute denotes the number of days before the booking that an attendee registered for the fitness class. It reflects the planning horizon for attendees.
- It can be used to understand the booking patterns of attendees, as individuals who book far in advance may have different attendance behavior than those who book on short notice.

5. Day_of_week:

- This feature specifies the day of the week when the fitness class takes place. It is a categorical variable representing the days of the week (e.g., Monday, Tuesday).
- Day of the week can influence attendance, as people's schedules and preferences vary based on the day.

6. Time:

- The 'time' feature indicates the time of day when the fitness class is scheduled. It can be categorized into segments like 'AM' (Morning) and 'PM' (afternoon/evening).
- Time of day can impact attendance, as some individuals may prefer morning workouts while others may prefer evening sessions.

7. Category:

- This attribute represents the category or type of fitness class being offered (e.g., Strength, HIIT, Cycling). It is a categorical variable.
- The fitness category can be a significant factor in predicting attendance, as people may have different preferences for fitness activities.

8. Attended:

- 'Attended' is the target variable and indicates whether an attendee actually attended the fitness booking. It is a binary variable with values 1(attended) and 0(did not attend).
- This is the variable we aim to predict using the other features. It represents the outcome of interest.

Data Cleaning

Data Analysis is a critical step in the data analysis process. It ensures that the dataset is accurate, consistent, and free from errors or inconsistencies that could affect the quality of analysis and modeling. In the context of our fitness booking attendance prediction project, I performed the following data cleaning steps:

1. Handling 'days_before' column:

- I transformed the 'days_before' column to remove the "days" label and converted the values to integers. This step is essential to make the data suitable for numerical analysis.

2. Standardizing 'day_of_week' values:

- I standardized the 'day_of_week' values by abbreviating the full day names to shorter forms. This ensures consistency in the representation of weekdays.

3. Cleaning 'weight' values:

- I cleaned the 'weight' column by rounding the weight values to two decimal places. This step enhances the precision of the weight data.

4. Handling Missing Values:

- Data integrity is of paramount importance in any data analysis project. To ensure the reliability of our analysis, we conducted a thorough review of the dataset for missing values. In our dataset, there are 20 rows containing missing values, which represented a very small proportion of the total dataset consisting of 1500 rows.
- In the interest of data integrity and to minimize the impact on the analysis, we made the decision to remove these 20 rows with missing values. This action allows us to work with complete and reliable data, maintaining the vast majority of the dataset while sacrificing only a small fraction.
- This careful approach to handling missing the vast helps us ensure the robustness of our analysis and modeling while keeping data loss to a minimum.

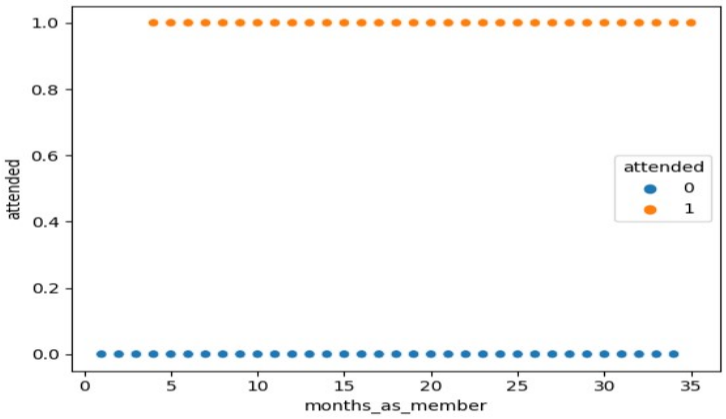
Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a critical step in understanding and preparing the dataset for modeling. The main objectives of EDA are as follows:

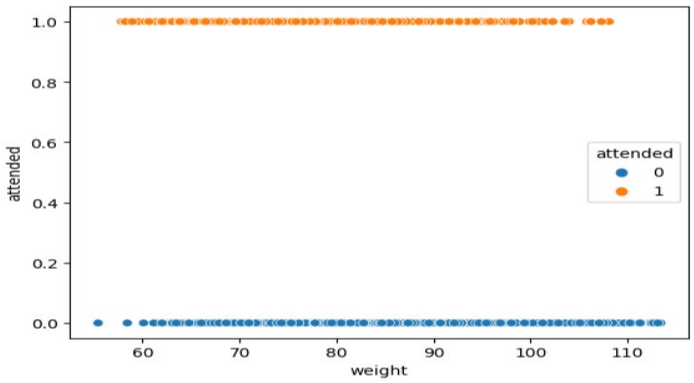
- **Understand the Data:** EDA helps us gain insight into the dataset's structure, its features and the relationships between them.
- **Identify Data Quality Issues:** EDA allows us to identify missing values, outliers, and any data inconsistencies that may need to be addressed.
- **Analyze the Impact on the Target Variable:** we examine how each feature in the dataset influences the target variable, which in this case, is the "attended" column. Understanding these relationships is crucial for building predictive models.
- **Data preprocessing:** EDA often highlights data preprocessing steps that may be required, such as handling outliers or missing values.

Analysis of Columns:

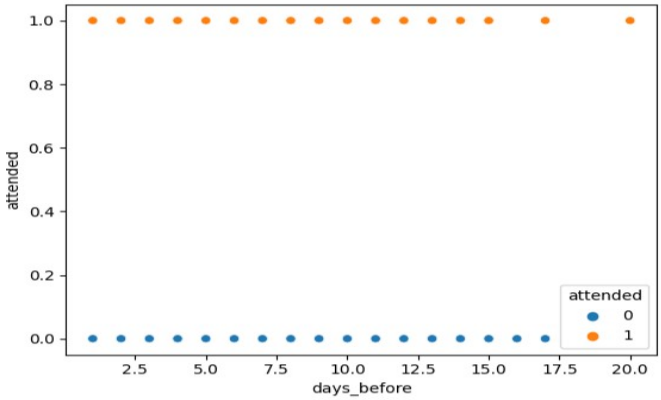
1. **Months_as_members:**
   - This column represents the number of months an individual has been a member.
   - A higher value in this column may indicate that longer-term members are more likely to attend bookings.
   - We should explore the distribution of this feature and its relationship with attendance.



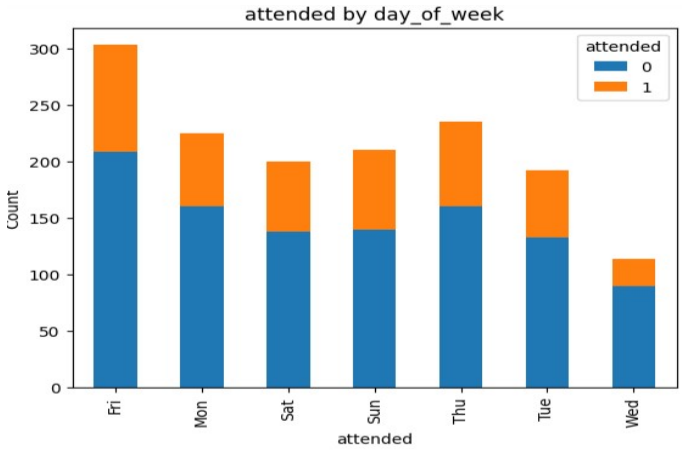
2. **Weight:**
   - The weight of the attendee could potentially impact attendance.
   - EDA can help us understand if there is a correlation between weight and attendance.



3. **Days_before:**
   - This column specifies the number of days before the booking that the attendee registered.
   - We should examine whether bookings made further in advance have a different attendance pattern compared to last-minute registrations.

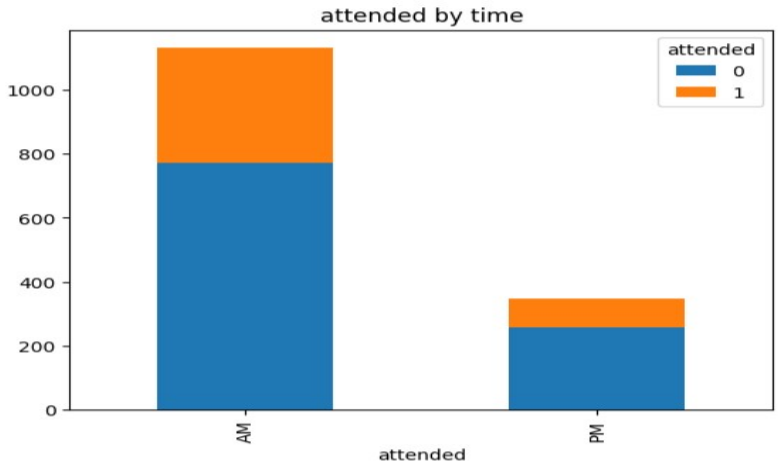


4. **Days_of_week:**
   - This categorical feature specifies the day of the week on which the booking occurs.
   - EDA can reveal whether certain days of the week have a higher or lower attendance rate.

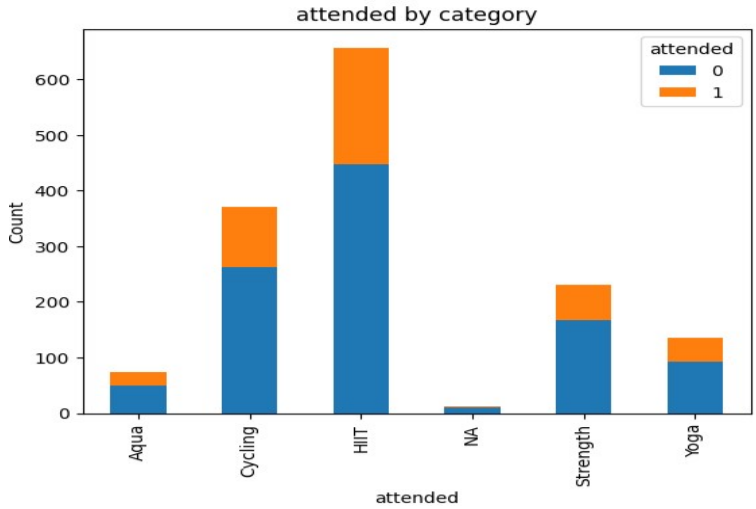


5. **Time:**
   - This categorical feature specifies the time of day for the booking.

- We can investigate whether attendance varies depending on the time of the day, such as morning or evening.



6. **Category:**
   - The category of the fitness class may play a significant role in attendance.
   - EDA can help us analyze which categories have higher or lower attendance rates.



Handling outliers:

- It's worth noting that during the EDA process, it was observed that some columns, such as **'months_as_member'**, **'weight'**, and **'days_before'**, contained outliers. These outliers were addressed using the Interquartile Range (IQR) method to ensure that they do not unduly influence the model. The removal of outliers is a common data preprocessing step to improve model performance and reliability.

Model Building

Model building is the process of creating a mathematical representation or algorithm that can make predictions or decisions based on input data, allowing machines to perform specific tasks, such as classification, regression, or pattern recognition. It involves selecting an appropriate architecture and training the model using data to optimize its performance.

Deep learning model building is utilized to leverage the power of artificial neural networks, which can capture complex patterns and relationships in data. This is particularly valuable for tasks where traditional machine learning models may struggle, such as image and speech recognition, natural language processing, and other intricate, high-dimensional data problems. Deep learning models can automatically learn and extract features from data, making them highly effective for a wide range of tasks.

The following steps involved in the Model building:

- **Data Splitting:** Data was split into training and testing sets using a 80-20% ratio (train-test split) to evaluate the model's performance. This separation ensures that the model is trained on one subset of the data and tested on an independent subset, allowing us to assess its generalization capabilities and avoid overfitting.

- **Data Preprocessing:**
  - **Separating Numerical and Categorical Columns:** In the initial data, we had a dataset with 1480 rows and 8 columns. To prepare the data for deep learning, we first separated the dataset into two types of columns: numerical and categorical. This separation is crucial because different preprocessing techniques are applied to each type.
  - **Preprocessing the data:** Before training and evaluating any model, it's essential to preprocess both the training and test data. This ensures that the model performs optimally and consistently across all datasets. The preprocessing steps for both the training and test data include the following:
    - ✓ **Scaling the Numerical Columns:** We standardized the numerical columns in both the training and test data using the StandardScaler from the scikit-learn library. Standardization transforms the data to have a mean of 0 and a standard deviation of 1. This step is crucial because it ensures that all numerical features have the same scale, preventing some features from dominating the learning process.
    - ✓ **One-Hot Encoding for Categorical Features:** For the categorical columns in both the training and test data, we used one-hot encoding. One-hot

encoding coverts categorical variables into binary (0/1) format, where each category becomes a separate binary column. This technique ensures that the deep learning algorithm can work with categorical data effectively.

- ✓ **Concatenating Numerical and Categorical Features:** After scaling and one-hot encoding, we concatenated the processed numerical and categorical features back together for both the training and test datasets. This step ensures that both datasets are prepared in the same way and are ready for model training and evaluation.

❖ **Balancing the data:** In deep learning, an imbalanced dataset can lead to model bias, where the model may perform poorly on the minority class. To address this issue, we used the Synthetic Minority Over-Sampling Technique (SMOTE) to balance the dataset. SMOTE generates synthetic samples for the minority class by interpolating between existing samples. This balancing step ensures that the model has an equal representation of both classes (attended and not attended) and helps improve its performance.

- After preprocessing, our training data now contains a balanced set of 1652 samples, with 826 samples for both attended and not attended categories.

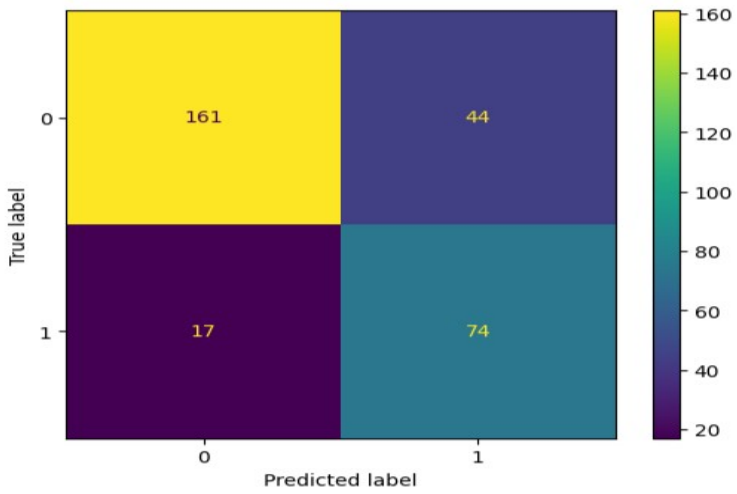
Building a Model Using Keras-Tuner:

- **Introduction to Keras Tuner:** Keras Tuner is a powerful library that automates the process of hyper parameter tuning, helping to find the optimal configuration for your deep learning models. In this documentation, we'll walk through the process of building a deep learning model using Keras Tuner for the given dataset.
- **Hyper parameter Optimization Strategy:** We will use Keras Tuner's Random Search strategy to explore different hyper parameter combinations. This strategy randomly samples from the defined hyper parameter space to find the best- performing model.
- **Hyper parameter Explored and Tuned:**
  1. Number of Hidden Layers: We vary the number of hidden layers between 5 and 10 to explore the network's depth.
  2. Neuron's per Hidden Layers: For each hidden layer, the number of neurons is tuned between 1 and 15.
  3. Activation Function: We explore three activation functions – **sigmoid**, **tanh**, and **ReLU** – for the hidden layers.

4. Weight Initialization: We experiment with weight initialization methods, including **glorot_uniform**, **glorot_normal**, **he_uniform**, **he_normal**.
5. Optimizer: We choose between four optimization algorithms – **Stochastic Gradient Descent (SGD)**, **Adam, RMSprop, and Adadelta**.

- **Objective and Metrics:** Our objective is to maximize validation accuracy while minimizing the binary cross-entropy loss. We aim to find the model configuration that best balances these two metrics.
- **Hyper parameter Tuning Process:**
  1. We define a function called '**best_model'** that constructs the neural network based on the hyper parameters selected by Keras Tuner.
  2. We use Keras Tuner to perform a random search over the hyper parameter space to find the best model configuration.
  3. The search is conducted with 15 training epochs, and we validate the models using a separate validation dataset.
- **Best Model Selection:** We select the best model based on the highest validation accuracy.
- **Optimal Hyper parameters:** The optimal hyper parameters are determined by Keras Tuner and may vary from one run to another. The selected hyper parameters will be used to build the final model.
- **Model Training with Optimal Hyper parameters:** We train the selected model using the optimal hyper parameters for 10 epochs, with a batch size of 10, and a 15% validation split.
- **Model Configurations and Results:** The best model configuration may include a varying number of hidden layers, neurons per layer, and different activation functions. It's essential to understand that the model's architecture is determined by the tuning process.

  The model achieved an accuracy of 79% on the validation dataset. The training and validation loss trends are plotted to visualize the model's learning process.
- **Conclusion:** In this document, we have outlined the process of building a deep learning model using Keras Tuner. The model's hyper parameters have been efficiently optimized to achieve a balance between accuracy and loss, resulting in a high-performing model for the given dataset.


161
44
17
74
0
1
True label
0
1
Predicted label
160
140
120
100
80
60
40
20