WEEK 3: CHURN ANALYSIS REPORT

TEAM NAME: RIT 1410AI Team 5A

DATE: 04-11-2024

Team members

Team Member Name	**Email ID**
Abdullah Imran	abdullahimranarshad@gmail.com
Matthew Ojo	ojoaisosamatthew@gmail.com
Krishin Tharani	Krishintharani1+internships@gmail.com
Emani Likhita	likhi.m9363@gmail.com
Sangeeta Sahoo	sahoo1107.sangeeta02@gmail.com
John syllah	johnsyllah2003@gmail.com
Tracy Reson	tracyreson@gmail.com
Afra Falakh	afrafalakh16@gmail.com
Eluit Cruz	ej.cruz.sant@gmail.com

CONTENTS

1. INTRODUCTION

The purpose of this week's report is to emphasize the essential steps that go into comprehending, gathering, and evaluating data to pinpoint the primary drivers of student dropout. In today's competitive digital landscape, maintaining high levels of user engagement is essential for the success of online learning platforms. This report focuses on a predictive modeling project designed to understand, quantify, and predict customer churn—the likelihood that a user will stop using or engaging with the platform.

Churn analysis is critical because retaining current users helps maintain a stable user base and is often more cost-effective than acquiring new ones. This report aims to build a reliable model to predict user churn, enabling proactive interventions to enhance user retention. The central goal of this report is to analyze these patterns, engineer relevant features, and develop a predictive model to identify which users are at risk of churning.

The insights from this model can help the platform's management team make data-driven decisions to retain users, tailor engagement strategies, and ultimately reduce churn rates. In this case, we're looking at data provided by the Excelerate. Key information includes how much time users spend on the platform (engagement duration) and how often they are participating on a monthly or weekly basis. We also have records of when users interacted with the platform, which helps us understand how recently they've been active. All this information helps us see patterns in user behavior and identify reasons they might leave the platform. To build effective predictive models, we need to start with good data preprocessing and feature engineering.

In our Google Colab notebook, we followed a clear process to prepare the data. One important step is converting the Opportunity End Date column, which shows the last time each user engaged with the platform, into a standard date format. This allows us to easily calculate how many days have passed since each user was last active.

This information is really useful because it can help us spot users who might be at risk of leaving the platform. To better understand user engagement, we create some new metrics. First, we develop an Engagement Score, which combines different factors like how long users spend on the platform and how many learning opportunities they participate in. This gives us a single score that shows overall engagement. We also track how many different opportunities each user has participated in, which reflects their activity level and interests.

Additionally, we calculate the Days Since Last Engagement to see how long it's been since a user last interacted with the platform; longer gaps often mean they're more likely to leave. Based on these new metrics, we define a "churn" label to identify users who might disengage. A user is marked as churned if they have low engagement scores, long periods of inactivity, or have participated in only a few opportunities. These thresholds are based on an initial look at the data and help us identify patterns in user engagement.

We also check for missing values, as they can mess up our analysis. Finally, we standardize important columns like Engagement Score and Days Since Last Engagement to make sure that all features contribute equally to our model. This way, we prevent any one feature from having too much influence on the results.

The analysis has two main goals. First, we want to gain insights from the data about user engagement. By looking at the engagement patterns we've created, we can identify which factors are most important for keeping users active on the platform. This information can help us figure out what improvements are needed to boost user engagement. Second, we plan to develop a machine learning model that predicts which users are likely to churn or stop using the platform. We'll use techniques like neural networks to analyze the engagement patterns and find connections to churn risk. Once the model is built, we'll evaluate its performance to make sure it can effectively identify users who are at high risk of leaving.

In short, this report outlines the complete process of preparing the data, creating features, developing a churn prediction model, and evaluating its performance. The insights gained from this analysis will provide valuable guidance for strategic user retention efforts, potentially leading to targeted outreach and improvements to keep users engaged. By utilizing the predictive model and the insights generated, the platform's management team can take proactive steps to enhance the user experience, reduce churn rates, and maintain growth in user engagement over time. Overall, this report marks an important advancement in establishing a data-driven approach to user retention and engagement within the learning platform.

2. DATA PREPARATION

The data cleaning process was very essential, it began with changing the date format of the Opportunity End Date column, which is crucial for accurate calculations, such as determining the number of days since a user's last engagement. Having dates in a standard format ensures that the data can be used correctly in analyses. Before conversion, it's important to check for any blank or incorrect entries, as these could lead to errors during the process.

Next, new columns were created to capture patterns of user activity. The Engagement Score combines various user activities into a single metric by incorporating Engagement Duration and participation in different opportunities, which helps represent overall engagement. The Opportunity Participation Count tracks how many different opportunities a user has engaged with, while Days Since Last Engagement calculates the time since a user’s last activity by comparing today’s date to the Opportunity End Date. By creating these new metrics, raw data is transformed into meaningful measures of user behavior, enhancing the model's ability to identify patterns related to churn.

Users are then marked as “churned” based on three criteria: if their Engagement Score falls below a specific threshold if they haven't been active for a certain number of days, or if they've participated in fewer opportunities than a designated amount. Defining these churn labels helps the model understand which users are likely to disengage, thereby improving its predictive accuracy.

Addressing any missing data is also crucial, as blank values can disrupt calculations and lead to errors in analysis. Running a check on the dataset reveals how many blanks exist in each column, enabling informed decisions about whether to fill in missing values or remove incomplete rows or columns.

Finally, standardizing the new columns (Engagement Score, Opportunity Participation Count, and Days Since Last Engagement) using a tool like StandardScaler ensures that all features are on a similar scale, making it easier for models to process the data effectively. Features with very different ranges can confuse the model, so standardization is an important step. Debugging and verifying the data by printing summaries of key columns allows for a quick assessment of whether the data appears as expected. This step helps identify any unexpected or extreme values that could indicate mistakes in the data preparation process.

In summary, our data cleaning process has the key steps of transforming data to be usable, creating helpful new information, setting up rules to define churn, and standardizing data for modeling.

3. EXPLORATORY DATA ANALYSIS (EDA)

3.1 Descriptive Statistics

The dataset includes several key features that provide important statistical insights into user engagement on the platform. **Engagement Duration** measures how long each user spends interacting with the platform. Statistical descriptions for this feature reveal the total number of users with recorded engagement, the average engagement duration, and the standard deviation, which indicates the variation in engagement levels across users. Additionally, the minimum and maximum engagement times highlight the range of activity, helping to identify both highly active and less active users.

Another critical feature is the **Opportunity End Date**, which tracks each user's last recorded date of engagement. This description includes the earliest and latest dates, offering insights into the dataset's period and allowing us to determine whether any recent users are close to the current date. Understanding this time frame is essential for contextualizing user behavior and engagement patterns.

The **Engagement Score** serves as a composite metric that combines various user activities, such as engagement duration and participation in different opportunities. Its statistical description includes the mean and standard deviation, which indicate the typical engagement level and the variability among users. The minimum and maximum scores further illuminate the range, allowing for the identification of users with extremely low or high engagement levels, which is important for targeting retention efforts.

In addition, **Days Since Last Engagement** measures how long it has been since a user last interacted with the platform. This feature provides the mean and standard deviation to show average inactivity and its variation across users, along with minimum and maximum values that indicate the shortest and longest inactivity periods. This information is particularly useful for recognizing which users may be highly disengaged and at risk of churn.

Finally, the **Opportunity Participation Count** reflects the number of distinct opportunities each user has engaged with, offering insights into user activity. Key statistics, such as the average participation count and the range of participation, help identify both the most and least active users in terms of module engagement.

Additionally, the **Churn Count** summarizes the total number of churned and non-churned users based on defined thresholds, providing a high-level understanding of the distribution between at-risk and engaged users.

Together, these statistical descriptions form the foundation for understanding user engagement patterns and for building an effective predictive model to identify high-risk users based on their behavior trends.

3.2 Visualization Done

Charts and graphs showing relationships between features and student drop-offs.

1. Engagement Score by Status Code

Visualization: Bar Chart

This bar chart displays the average engagement score across various status codes, highlighting how status code assignments relate to engagement levels and potential student drop-offs.

Findings: Certain status codes correlate with lower average engagement scores, indicating possible points of disengagement.

Recommendation: Analyze status codes linked to low engagement and evaluate if additional support or intervention strategies are required for students in these categories.

2. Engagement Score by Apply Date

Visualization: Scatter Plot with Trendline

The scatter plot with a trendline explores the relationship between application dates and engagement scores, providing insight into how timing influences engagement and potential drop-off risks.

Findings: Periodic dips in engagement scores suggest seasonal trends or timing issues that could impact retention.

Recommendation: Adjust recruitment or onboarding efforts to align with periods of higher engagement or address challenges associated with lower engagement times.

3. Engagement Score by Days to Opportunity End Date

Visualization: Line Plot

This line plot shows how engagement fluctuates as the opportunity end date approaches. Understanding these fluctuations provides insight into whether engagement consistently wanes near the end of a program.

Findings: A trend of declining engagement as the end date nears may point to disengagement and drop-offs.

Recommendation: Implement targeted support for students approaching the end date to maintain engagement.

4. Engagement Score by Demographics

a) Age vs. Engagement Score (Box Plot)

The box plot illustrates engagement scores across various age groups, revealing demographic patterns related to drop-offs.

Findings: Specific age groups may exhibit lower engagement, suggesting varying levels of interest or challenges.

Recommendation: Develop age-targeted interventions to support at-risk age groups and reduce drop-off rates.

b) Gender vs. Engagement Score (Bar Chart)

The bar chart compares average engagement scores by gender, highlighting differences in engagement that may relate to gender-specific needs.

Findings: Engagement disparities by gender could reflect factors impacting student retention.

Recommendation: Consider gender-sensitive strategies to foster engagement and address potential disparities in program design.

5. Engagement Score by Current/Intended Major

Visualization: Box Plot

This box plot examines engagement scores across different majors, providing insight into subject-specific factors influencing drop-off rates.

Findings: Lower engagement in certain majors suggests subject-specific challenges or lack of interest.

Recommendation: Adjust content or provide additional resources in lower-engagement fields to retain students within those majors.

6. Engagement Score by Institution Name

Visualization: Bar Chart

This analysis compares the average engagement scores of students from the top 10 institutions. Institutions with consistently low engagement scores could indicate environmental factors affecting retention.

Findings: Certain institutions may have inherent challenges that influence student engagement and retention.

Recommendation: Investigate practices in high-engagement institutions and apply these insights to support students from lower-engagement institutions.

7. Engagement Score Over Time by Entry Date

Visualization: Line Plot

The line plot visualizes the average engagement score over time, grouped by month. This helps identify any academic period-based changes in engagement that could affect student drop-offs.

Findings: Seasonal or time-based dips in engagement may correspond to periods when students are more likely to disengage.

Recommendation: Focus on engagement initiatives during periods with lower engagement to reduce the likelihood of drop-offs.

8. Engagement Score by Opportunity Category

Visualization: Bar Chart

The bar chart displays engagement scores by opportunity category, identifying categories that may be more or less engaging and potentially linked to drop-off Patterns.

Findings: High engagement in specific categories could indicate strong interest, while lower scores may flag areas where students lose interest.

Recommendation: Expand high-engagement categories and enhance lower-engagement opportunities to retain students.

9. Engagement Score by SignUp DateTime

Visualization: Scatter Plot

The scatter plot reveals engagement levels about sign-up times, which may indicate optimal times to engage or predict drop-off points.

Findings: Patterns in engagement based on sign-up time may help optimize timing for outreach and reduce early drop-offs.

Recommendation: Schedule onboarding and engagement touchpoints to align with high-engagement sign-up periods.

3.3 Patterns and Trends:

Several trends and patterns can be observed in user engagement data that provide insights into behavior and potential strategies for improvement. One significant pattern is that users with higher Engagement Scores tend to interact more frequently and for longer durations. If the distribution of Engagement Scores shows a notable skew, it may indicate the presence of a small group of highly engaged users, while the majority exhibit lower engagement levels. This "power-user" effect suggests that a small percentage of users are driving overall activity on the platform. Identifying strategies to encourage moderate users to increase their engagement could be an effective way to boost retention.

Another important trend relates to the Days Since Last Engagement. Users who have higher values in this metric are more likely to churn, particularly if the distribution is right-skewed, indicating that while most users remain active, some have not engaged recently. This pattern suggests the existence of a threshold period of inactivity—such as 60 days—after which users are significantly more likely to leave the platform. Recognizing this threshold can inform proactive re-engagement strategies aimed at reaching out to users before they hit critical inactivity levels.

Additionally, the Opportunity Participation Count offers valuable insights into user behavior. Users who participate in more opportunities generally have higher Engagement Scores, indicating that exploring a variety of content is linked to overall engagement. This trend highlights the potential benefits of encouraging users to try multiple opportunities, which could improve their engagement and reduce churn rates. If certain opportunities are consistently showing low participation, it may signal the need for content optimization or enhanced promotional efforts to attract more users.

Finally, analyzing the relationship between user features and churn reveals that those with lower Engagement Scores, fewer opportunities engaged, and higher Days Since Last Engagement are more likely to be classified as churned. If distinct thresholds for these features are identified, it could suggest that specific engagement activities are critical for user retention. Monitoring these metrics regularly would allow for timely interventions to prevent users from reaching levels associated with a higher risk of churn.

3.3 Data Standardization:

The following things were done in this phase:

- Scaling Age of Learner, Engagement Duration
- Applying One Hot encoding to: 'Opportunity Name', 'Opportunity Category', 'Gender', 'Current Student Status', 'Status Description'
- Applying the hashing technique of hot encoding: first name, last name, institution name, location, and the major

4. CHURN ANALYSIS

4.1 Key Factors

Based on the analysis, several primary factors contribute to student drop-offs:

1. **Low Engagement Score**: The engagement score, a composite metric derived from participation in specific opportunities, is a key predictor of student retention. A low score indicates minimal interaction with available resources and activities, suggesting disengagement from the program. This disengagement often precedes drop-off, as students who do not find value in the resources or events offered are more likely to leave.
2. **Low Opportunity Participation**: Students with limited involvement in opportunities, such as workshops or courses, exhibit a higher likelihood of churn. The notebook defines low participation as engagement in fewer than two opportunities. When students do not actively participate in these engagements, it often reflects a lack of connection with the program, reducing their commitment and increasing the chance of departure.

3. **Extended Inactivity (Days Since Last Engagement)**: Long periods of inactivity, defined as over 126 days in this analysis, significantly correlate with student drop-offs. When students remain inactive for extended periods, they lose the momentum to stay connected with the learning environment, making re-engagement increasingly challenging. This extended disengagement period strongly indicates a reduced likelihood of their return.

These factors contribute to student drop-offs by capturing varying dimensions of disengagement: a low engagement score reflects overall disinterest, limited participation indicates a lack of integration into the program, and extended inactivity marks a physical and mental disconnection from learning activities. Together, they form a comprehensive picture of students at risk of leaving, enabling targeted interventions to improve retention.

4.2 Impact Analysis

These factors—low engagement score, limited opportunity participation, and extended inactivity—each contribute to a heightened likelihood of students leaving by highlighting distinct forms of disengagement.

1. **Low Engagement Score**: This metric reflects how involved and committed a student is with the program. A low engagement score signals a lack of meaningful interaction, which often correlates with diminished interest and motivation. When students don’t feel actively engaged or see value in the activities, they are more inclined to detach, eventually leading to drop-off.
2. **Limited Opportunity Participation**: Participation in workshops, courses, or other opportunities fosters a sense of belonging and progress. Students who engage in fewer than two opportunities may not feel integrated into the program’s community or lack exposure to enriching experiences that build their commitment. Without these touchpoints, students may feel disconnected and are more likely to leave.

3. **Extended Inactivity**: A prolonged period of inactivity, such as exceeding 126 days without engagement, creates both a physical and psychological distance from the program. The longer students remain inactive, the harder it becomes for them to reconnect, especially as they may lose familiarity with the program's structure and momentum. This disconnection reduces their likelihood of re-engagement, making drop-off a more probable outcome.

Together, these factors paint a picture of disengagement that, if left unaddressed, greatly increases the chances of students leaving the program. Identifying and addressing these elements early on can help in implementing targeted interventions to re-engage at-risk students and improve retention.

4.3 Churn Analysis Code Explanation:

4.3.1. Data Preparation and Conversion

The analysis begins by converting the 'Opportunity End Date' column in the student's DataFrame to a date-time format. This conversion allows for accurate time-based calculations, particularly for determining the time elapsed since the student's last engagement.

4.3.2. Creating Engagement Metrics

To assess student engagement, two key metrics were created:

- **Engagement Score:** This composite score is calculated by weighting the Engagement Duration by a factor of 0.5 and adding participation counts in three significant learning opportunities: Career Essentials, Data Visualization, and Digital Marketing. The Engagement Score provides a holistic measure of a student's interaction with the program.

- **Opportunity Participation Count:** This metric sums the student's participation in the same three key opportunities, providing a snapshot of their involvement across multiple learning modules.

4.3.3. Defining Churn Indicators

Churn was identified using three thresholds based on observed engagement patterns:

- **Engagement Score Threshold:** Students with an Engagement Score below 2.112069 are flagged as at risk for churn. This threshold was set based on score distribution, aiming to capture students with low overall engagement.
- **Recent Inactivity:** The time since the last engagement (measured in days from the most recent 'Opportunity End Date') was flagged if it exceeded 126 days. This threshold was established to detect students who have been inactive for a significant period, suggesting potential disengagement.
- **Low Opportunity Participation Count:** Students with fewer than two participations across key opportunities are flagged as churned, as limited participation indicates disengagement from the program's core learning modules.

How these were identified: The 25% values were used as a threshold for each column to indicate a churn.

- Number of Churned Students: 878
- Number of Non-Churned Students: 1317

4.3.4. Churn Label Creation

A binary Churn column was created, where students are labeled as churned (1) if they meet any of the churn criteria above, and not churned (0) otherwise. This label facilitates a clear analysis of the churned versus non-churned population for targeted engagement strategies.

4.3.5. Feature Scaling

To prepare the data for further analysis, selected features—Engagement Score, Opportunity Participation Count, and Days Since Last Engagement—were scaled using StandardScaler. This normalization ensures that each feature contributes proportionately to any subsequent analysis or modeling.

4.3.6. Churn Analysis Summary

The number of churned and non-churned students was calculated and printed, providing a quick overview of the population's engagement and churn risk distribution.

4.3.7. Descriptive Statistics for Debugging

To validate data accuracy and confirm appropriate threshold settings, descriptive statistics were printed for key variables: Engagement Duration, Opportunity End Date, Engagement Score, Days Since Last Engagement, and Opportunity Participation Count. This debugging step helped ensure that thresholds and transformations aligned with the data's distribution and intended analysis.

5. PREDICTIVE MODELING

5.1 Trial Predictive Analysis

In predictive modeling, we first analyzed the code differently and tried to predict the data using the entirety of the dataset, and then we would have predicted churn through that. This was the first approach that we would like to label as **"trial predictive analysis".** Let us walk through exactly what we did in this phase.

5.1.1 Dimensionality Reduction

In this phase, firstly we did the EDA process, cleaned and transformed the data, and then tried to see the features and the labels, which were way too much, and simply not needed. We then tried to test at what particular number of features the data showed a high amount of variance.

As seen below, we can visualize that the data still shows a very high variance of 90% when it had 200 features. So, we used this particular approach and used dimensionality reduction to reduce the number of features using Principal Component Analysis.

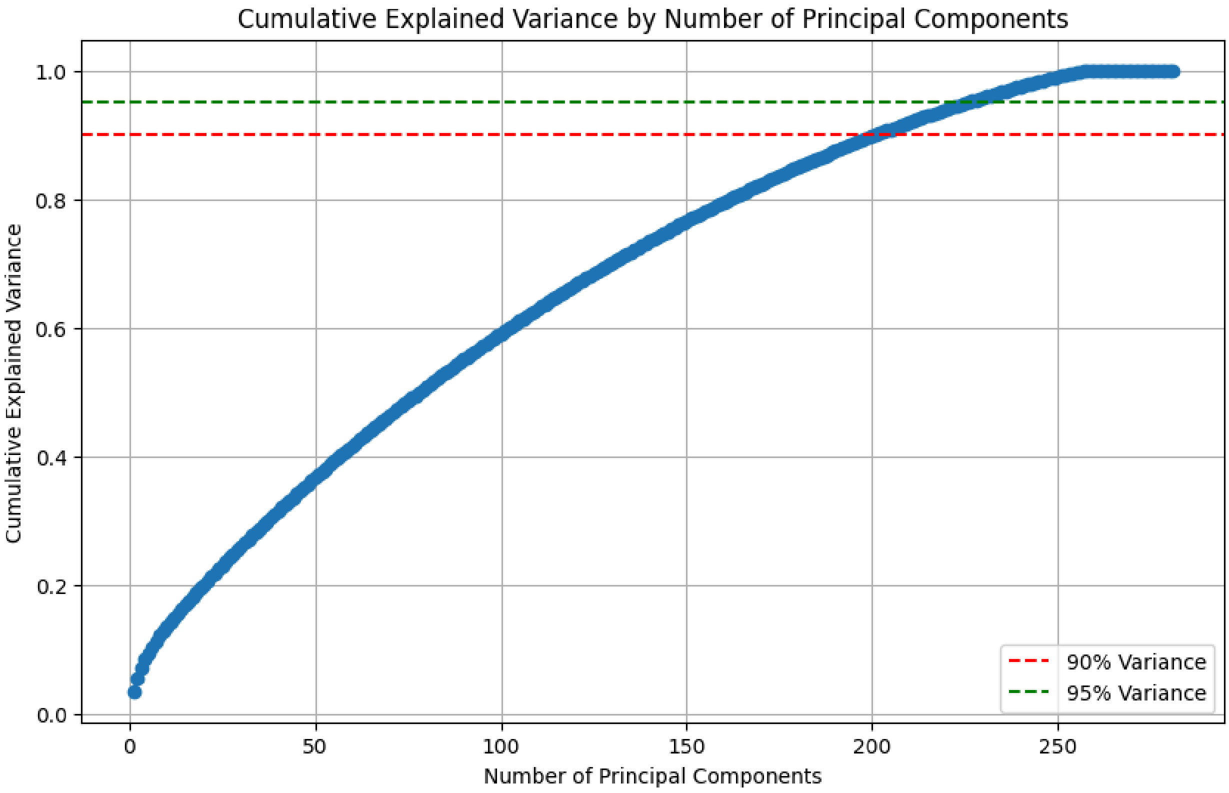
Data Standardization: $X=X-\mu/\sigma$

Covariance Matrix: C = (1/n-1)X^T.X

Eigen Values and Eigen Matrices: Cv=λv

Transform the data: Y=X⋅Vk

The other features were dropped and then we used only the 200 features as standard.



5.1.2 Model Training & Evaluation

We used Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting as an ensemble, K Nearest Neighbor, Gaussian Naive Bayes, and Support Vector Machines as our model to train and test our data. We achieved the following accuracies:

- Logistic Regression Accuracy: 1.00
- Decision Tree Accuracy: 0.97
- Random Forest Accuracy: 0.97

- Gradient Boosting Accuracy: 0.99
- K-Nearest Neighbors Accuracy: 0.69
- Naive Bayes Accuracy: 0.96
- Support Vector Machine Accuracy: 0.99

Then we ensembled these models using Voting Classifier, particularly the soft voting classifier and got a 100% accuracy. However, the models did struggle against a few class instances, that could have indicated the class imbalance, and using this further, we had to do churn analysis. We got the following results in the form of a classification report:

Classification Report:

```
             Precision recall  f1-score   support

       1050      1.00      1.00      1.00        46
       1070      0.98      1.00      0.99       292
       1080      0.98      0.99      0.98        95
       1110      1.00      0.20      0.33         5
       1120      0.00      0.00      0.00         2

   accuracy                          0.98       440
  macro avg      0.79      0.64      0.66       440
weighted avg      0.98      0.98      0.98       440
```

Confusion Matrix:

```
[[ 46   0   0   0   0]
 [  0 292   0   0   0]
 [  0   1  94   0   0]
 [  0   2   2   1   0]
 [  0   2   0   0   0]]
```

Since we had to do churn analysis further, this would indicate that a few columns would have predicted the churn instead of the entirety of the dataset, and hence, in the actual prediction, we used that approach.

5.2 Model Selection:

5.2.1. Logistic Regression

- **Formula**: P(Y=1∣X)=11+e−(β0+β1X1+⋯+βnXn)P(Y = 1 | X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \cdots + \beta_n X_n)}}P(Y=1∣X)=1+e−(β0+β1X1+⋯+βnXn)1
- **What It Is**: A linear model used for binary classification by estimating the probability of an event.
- **Use for Churn**: Suitable for churn analysis as it provides clear probabilities, helping predict the likelihood of churn based on key features.

5.2.2. Decision Tree

- **Formula**: f(x)=arg maxy Majority Class(X)f(x) = \text{arg max}_y \, \text{Majority Class}(X)f(x)=arg maxyMajority Class(X)
- **What It Is**: A tree-structured model where each node represents a decision based on a feature split.
- **Use for Churn**: Good for churn as it handles non-linear relationships and makes interpretable predictions based on feature importance.

5.2.3. Support Vector Machine (SVM)

- **Formula**: f(x)=sign(w⋅X+b)f(x) = \text{sign}(w \cdot X + b)f(x)=sign(w⋅X+b)
- **What It Is**: A classification model that finds the optimal hyperplane maximizing the margin between classes.
- **Use for Churn**: Effective for churn prediction with high-dimensional data, especially when a clear separation exists between churned and non-churned customers.

5.2.4. K-Nearest Neighbors (KNN)

- **Formula**: f(x)=arg maxy∑i∈Nk1(yi=y)f(x) = \text{arg max}_y \sum_{i \in N_k} \mathbb{1}(y_i = y)f(x)=arg maxy∑i∈Nk1(yi=y)
- **What It Is**: A non-parametric model that classifies based on the majority label among the nearest K neighbors.
- **Use for Churn**: Useful for churn detection as it can capture local patterns by comparing each customer's data to similar profiles.

5.2.5. Random Forest

- **Formula**: f(x)=majority vote of {fm(x)}m=1Mf(x) = \text{majority vote of } \{f_m(x)\}_{m=1}^{M}f(x)=majority vote of {fm(x)}m=1M
- **What It Is**: An ensemble of decision trees, combining their outputs for improved accuracy and robustness.
- **Use for Churn**: Random Forest handles complex data interactions and reduces overfitting, making it effective for churn prediction.

5.2.6. Artificial Neural Network (ANN)

- **Formula**: y=f(WTX+b)y = f(W^T X + b)y=f(WTX+b)
- **What It Is**: A network of neurons with multiple layers that capture complex patterns in data.
- **Use for Churn**: ANNs can capture non-linear relationships in churn data, especially when input features interact in complex ways.

7. Recurrent Neural Network (RNN)

- **Formula**: ht=f(W⋅ht−1+U⋅Xt)h_t = f(W \cdot h_{t-1} + U \cdot X_t)ht=f(W⋅ht−1+U⋅Xt)
- **What It Is**: A neural network with memory, designed to handle sequential data and time-series relationships.

- **Use for Churn**: Useful for churn analysis when customer behavior over time impacts churn likelihood, allowing predictions based on activity history.

5.2.8. Autoencoder

- **Formula**: X≈g(f(X))X \approx g(f(X))X≈g(f(X))
- **What It Is**: A neural network that learns to encode and then reconstruct data, identifying essential features.
- **Use for Churn**: Effective for churn detection as it highlights anomalies, making it useful to detect rare or unusual customer behaviors that signal churn.

5.2.9. Multilayer Perceptron (MLP)

- **Formula**: y=f(Wn⋅f(Wn−1⋯f(W1X+b1)⋯+bn−1)+bn)y = f(W_n \cdot f(W_{n-1} \cdots f(W_1 X + b_1) \cdots + b_{n-1}) + b_n)y=f(Wn⋅f(Wn−1⋯f(W1X+b1)⋯+bn−1)+bn)
- **What It Is**: A fully connected feedforward neural network that captures complex relationships through multiple layers.
- **Use for Churn**: MLPs are versatile for churn prediction, especially in datasets with non-linear relationships and interactions across multiple features.

5.3 Model Training

5.3.1 Logistic Regression

Data Preparation

The code prepares the data by setting up feature and target variables from the students_churn DataFrame. The features selected—Engagement Score, Opportunity Participation Count, and Days Since Last Engagement—represent key aspects of student engagement, while the target variable, Churn, indicates whether a student is likely to churn (1) or not (0).

Data Splitting

The dataset is split into training and test sets, with 30% of the data reserved for testing. Stratified splitting is used to ensure that the churn ratio in the training and test sets reflects the overall dataset, maintaining class balance for reliable model performance evaluation.

Hyperparameter Tuning and Pipeline Setup

A pipeline is created with a StandardScaler to normalize feature values and a Logistic Regression model to predict churn. The pipeline is optimized through a grid search with cross-validation to find the best hyperparameters (C values for regularization strength and regularization type). This helps fine-tune model performance by selecting the optimal parameter combination.

Model Evaluation

Predictions are made on the test set, and key evaluation metrics are calculated, including the confusion matrix, classification report (precision, recall, F1-score), and accuracy score. These metrics assess the model's ability to correctly classify churned and non-churned students, indicating its effectiveness in predicting churn.

Cross-Validation

Cross-validation is performed on the best estimator found during grid search to assess the model’s stability and generalizability. The code calculates cross-validation scores over five folds and outputs the mean score, offering insight into how well the model performs across different subsets of data.

5.3.2 Decision Tree

Splitting Criteria and Structure

The decision tree works by creating binary splits at each node based on a feature value. Each split aims to maximize the separation between churned and non-churned students, to reach a pure classification in the leaf nodes.

The tree's structure allows it to capture non-linear relationships and interactions among the features, making it well-suited for datasets where factors contribute differently to each class.

Hyperparameter Tuning

Several parameters are adjusted to optimize the model's performance:

- Max Depth: Controls the maximum depth of the tree to prevent overfitting by limiting its complexity.
- Min Samples Split: Specifies the minimum number of samples required to split a node, which helps in controlling the granularity of the model.
- Min Samples Leaf: Ensures a minimum number of samples in the leaf nodes, further reducing overfitting by smoothing predictions.

Model Interpretation and Churn Prediction

The decision tree provides interpretable rules for churn prediction, as each path from root to leaf in the tree represents a decision rule based on feature values.
This interpretability is particularly useful in churn analysis, as it allows the identification of specific engagement thresholds or behaviors linked to a higher likelihood of churn.

5.3.3 Support Vector Machines

Feature Selection

The model uses three key features:

- Engagement Score: Measures overall engagement intensity.
- Opportunity Participation Count: Counts the number of key opportunities the student has participated in.
- Days Since Last Engagement: Captures the recency of the student's last activity.
- Kernel Selection and Hyperparameters.

Hyperparameters:

SVM has various kernels (linear, radial basis function (RBF), polynomial) that map data into higher-dimensional spaces to find better separations. The model tests different kernel types, along with:

- C: A regularization parameter that controls the trade-off between achieving a low error on the training data and a large margin.
- Gamma: Controls how much influence a single training example has, affecting the model's flexibility in capturing patterns.

Hyperparameter Tuning

Grid search with cross-validation is used to identify the optimal values for C, gamma, and kernel type, balancing the model's complexity and accuracy. This tuning process refines the SVM to achieve the best separation between churned and non-churned students.

5.3.4 K-Nearest Neighbor

Parameter Selection

KNN has multiple tunable parameters that influence its performance:

- n_neighbors: Determines the number of closest students used to make a churn prediction.
- Weights: Controls the weighting of neighbors (uniform or distance-based).
- Metric: Specifies the distance calculation method (e.g., Euclidean or Manhattan distance).

Hyperparameter Tuning

Grid Search with Cross-Validation is performed to identify the optimal values for n_neighbors, weights, and metrics. This tuning helps balance the bias-variance trade-off, optimizing the KNN's ability to accurately predict churn.

5.3.5 Random Forest

Parameter Selection

The Random Forest Classifier has several hyperparameters that can be tuned for better performance:

- n_estimators: The number of trees in the forest, impacts the model's robustness.
- max_depth: Controls the maximum depth of the trees, which can help prevent overfitting.
- min_samples_split: The minimum number of samples required to split an internal node.
- min_samples_leaf: The minimum number of samples required to be at a leaf node, ensuring that the model is not too complex.
- max_features: The number of features to consider when looking for the best split, influencing the diversity among trees.

Hyperparameter Tuning

Randomized Search with Cross-Validation is utilized to efficiently explore the hyperparameter space and identify the optimal settings for the Random Forest model. This method allows for a more comprehensive exploration of parameters compared to Grid Search, leading to potentially better model performance with reduced computational cost.

5.3.6 Artificial Neural Network

Model Architecture

- The model is defined using Keras as a Sequential model, indicating a linear stack of layers where each layer has a specific function.
- The architecture consists of:
    - **Input Layer**: Accepts input features with a shape matching the number of features in the dataset. In this case, it takes the transformed features from the **`students_churn`** DataFrame, which include:
        - Engagement Score
        - Opportunity Participation Count
        - Days Since Last Engagement
    - **Hidden Layers**:
        - **First Hidden Layer (Dense Layer)**:

- Contains **64 neurons** with a ReLU (Rectified Linear Unit) activation function.
- Each neuron processes the input it receives by applying a weighted sum followed by the activation function, introducing non-linearity into the model.
- This layer learns to extract complex patterns from the input data, capturing relationships between the features and churn outcomes.
- **Second Hidden Layer (Dense Layer)**:
  - Contains **32 neurons**, also using the ReLU activation function.
  - This layer further refines the learned representations from the previous layer, enabling the model to learn hierarchical feature interactions.

○ **Output Layer**:

- Contains a **single neuron** with a sigmoid activation function.
- This neuron outputs a probability score between 0 and 1, representing the likelihood of a student churning. If the output is greater than 0.5, the student is classified as churned (1); otherwise, they are not churned (0).

Neurons and Their Functions

- Each neuron in the hidden layers functions by receiving inputs, applying weights to these inputs, summing them, and passing them through the ReLU activation function.
- The ReLU function outputs zero for any negative input and retains positive values, which helps mitigate the vanishing gradient problem during training.
- By combining the outputs from multiple neurons, the hidden layers can capture complex relationships in the data, enabling the model to learn patterns that might indicate a student's likelihood of churn.

- The output neuron consolidates the information from the hidden layers to produce a final decision, effectively summarizing the insights gained through the neural network's processing.

Model Compilation and Training

- The model is compiled using the Adam optimizer, which is efficient for training deep learning models due to its adaptive learning rate capabilities.
- The loss function is set to binary cross-entropy, suitable for binary classification tasks, measuring the difference between predicted probabilities and actual class labels.
- The model is trained over **100 epochs** with a batch size of **32**, using a validation split to monitor performance on unseen data during training. This helps in detecting overfitting.

Evaluation and Predictions

- After training, the model is evaluated on the test dataset, providing an accuracy score that reflects its performance in predicting churn.
- Predictions are made on the test set, where the output probabilities are thresholded to classify students as churned or not.
- The confusion matrix and classification report are generated to assess the model's classification performance, providing insight into metrics such as precision, recall, and F1-score.

5.3.7 Recurrent Neural Network

Model Structure

- **Input Features**: The model uses three features:
    - Engagement Score
    - Opportunity Participation Count
    - Days Since Last Engagement

- **Reshaping Data**: The input data is reshaped to fit the RNN format of `[samples, timesteps, features]`, with a single timestep for simplicity.

Layers

- **First Layer (SimpleRNN)**: Contains 64 neurons with ReLU activation and returns sequences for the next layer.
- **Dropout Layer**: Reduces overfitting by randomly dropping 20% of the neurons during training.
- **Second Layer (SimpleRNN)**: Contains 32 neurons with ReLU activation, consolidating information from the previous layer.
- **Output Layer (Dense)**: A single neuron with a sigmoid activation function outputs the probability of churn.

Training Process

- The model is compiled with the Adam optimizer and binary cross-entropy loss function.
- It is trained for 50 epochs with a batch size of 32, using a validation split to monitor performance.

5.3.8 Autoencoder

Model Purpose

- The autoencoder is designed to learn efficient representations of the input data, specifically reducing the dimensionality of features related to student engagement to identify patterns relevant to churn prediction.

Model Structure

- **Input Features**: The model utilizes three features:
  - Engagement Score

    - Opportunity Participation Count
    - Days Since Last Engagement
- **Encoder**:
    - The encoder consists of a dense layer with 2 neurons and ReLU activation, compressing the input data into a lower-dimensional representation (2 dimensions).
- **Decoder**:
    - The decoder reconstructs the input data from the encoded representation with a dense layer matching the original feature dimensions and using a sigmoid activation function.

Training Process

- The model is compiled with the Adam optimizer and mean squared error (MSE) as the loss function.
- It is trained over 50 epochs with a batch size of 32, using the same data for both input and output (self-supervised learning).

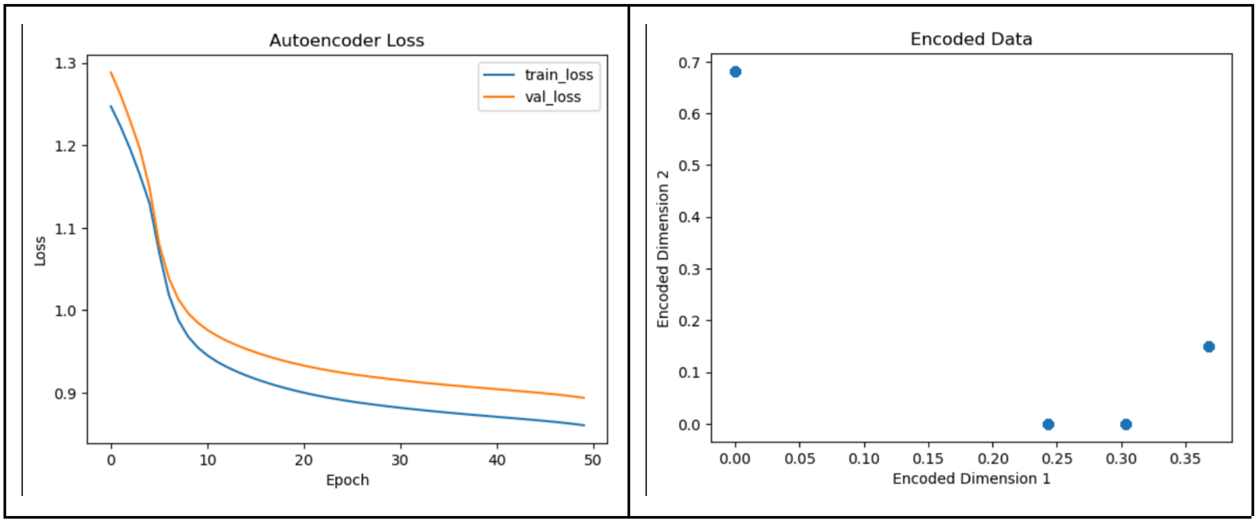
Evaluation

- The model's performance is assessed on the test set, reporting the reconstruction loss. A lower loss indicates a better representation of learning.

Visualization

- The training and validation loss can be plotted over epochs to monitor convergence.
- The encoded data can be visualized in a scatter plot to explore the distribution and separability of data points in the reduced feature space.

Test Loss: 0.8767512440681458



5.3.9 Multi-layer Perceptron

Model Purpose

- The MLP classifier is employed to predict student churn based on engagement metrics. It aims to classify students as likely to churn or not, based on their engagement levels and participation in opportunities.

Model Structure

- **Input Features**: The model takes three key features:
  - Engagement Score
  - Opportunity Participation Count
  - Days Since Last Engagement
- **Hidden Layer**:
  - The MLP consists of a single hidden layer with 100 neurons. Each neuron applies a nonlinear activation function (default is ReLU) to learn complex patterns in the data.
- **Output Layer**:
  - The output layer uses a sigmoid activation function to produce a probability score for binary classification (churn vs. no churn).

Training Process

- The model is set to a maximum of 1000 iterations for training, allowing sufficient time to converge on a solution. It utilizes backpropagation to adjust weights based on the error of predictions.

Evaluation

- The model's performance is evaluated on a test set, using metrics such as:
    - Confusion Matrix: Displays true positives, true negatives, false positives, and false negatives.
    - Classification Report: Provides precision, recall, F1 score, and support for each class.
    - Accuracy Score: Represents the overall correctness of the model's predictions.

5.4 Performance Metrics

5.4.1. Confusion Matrix

A confusion matrix is a summary of prediction results on a classification problem. It presents the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) in a tabular format.

For the provided confusion matrix:

```
Confusion Matrix:

[[262   0]  # Predicted: 0

 [  0 177]]  # Predicted: 1
```

- **True Positives (TP)**: 177 (correctly predicted churn)

- **True Negatives (TN)**: 262 (correctly predicted no churn)
- **False Positives (FP)**: 0 (incorrectly predicted churn)
- **False Negatives (FN)**: 0 (incorrectly predicted no churn)

5.4.2. Precision

Precision indicates the accuracy of positive predictions. It measures how many of the predicted positive cases were positive.

Formula: Precision=TPTP+FP\text{Precision} = \frac{TP}{TP + FP}Precision=TP+FPTP

In this case: Precision=177177+0=1.00\text{Precision} = \frac{177}{177 + 0} = 1.00

Precision=177+0177=1.00

5.4.3. Recall

Recall (also known as sensitivity) measures the ability of a model to find all the relevant cases (i.e., actual positives). It assesses how many actual positive cases were correctly identified.

Formula: Recall=TPTP+FN\text{Recall} = \frac{TP}{TP + FN}Recall=TP+FNTP

In this case: Recall=177177+0=1.00\text{Recall} = \frac{177}{177 + 0} = 1.00

Recall=177+0177=1.00

5.4.4. F1 Score

The F1 score is the harmonic mean of precision and recall. It provides a balance between the two metrics, especially when the class distribution is imbalanced.

Formula: F1 Score=2×Precision×RecallPrecision+Recall\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}

F1 Score=2×Precision+RecallPrecision×Recall

In this case: F1 Score=2×1.00×1.001.00+1.00=1.00\text{F1 Score} = 2 \times \frac{1.00 \times 1.00}{1.00 + 1.00} = 1.00

F1 Score=2×1.00+1.001.00×1.00=1.00

5.4.5. Overall Interpretation

From the classification report:

- **Precision**: A precision of 1.00 for both classes indicates that every predicted positive (both churn and no-churn) was correct. There were no false positives.
- **Recall**: A recall of 1.00 for both classes shows that the model identified all actual positives. There were no false negatives.
- **F1 Score**: An F1 score of 1.00 confirms the model's perfect balance between precision and recall.
- **Support**: The support indicates the number of actual occurrences for each class in the test dataset. There were 262 instances of no churn and 177 instances of churn.
- **Accuracy**: An accuracy score of 1.0 signifies that the model made correct predictions for all instances in the test set.

6. CONCLUSION

The analysis was focused on the predictions of student churn through key indicators like engagement scores, opportunity participation, and days since the last engagement. Churn is defined based on thresholds where a low engagement score (below 2.112), low participation in opportunities (fewer than two), and prolonged inactivity (126 days or more) indicate a higher risk of churn. This approach reveals that low engagement and participation are crucial predictors of student disengagement. Distribution visualizations of engagement scores and inactivity days support these findings, showing that a significant portion of students have minimal engagement. Additionally, churn statistics offer a balanced perspective on engaged versus churned students, providing insights into potential retention improvements.

For further analysis, refining churn thresholds could yield a more accurate model by adjusting values based on feedback and emerging data patterns. Investigating specific opportunities (such as "Data Visualization" versus "Digital Marketing") may uncover trends in participation that correlate with churn, guiding targeted curriculum enhancements. Regularly monitoring churn over time could reveal seasonal patterns, aiding in the development of engagement strategies.

In conclusion, the distributions of engagement scores and days since the last engagement suggest that many students show minimal engagement, corroborating the churn prediction findings. Churn Statistics counts show a notable number of churned versus non-churned students, allowing for a balanced view of churn factors and offering insights for improving retention.

Finally, experimenting with additional predictors like demographic details or academic background could improve the model’s robustness and predictive power which can be inculcated as future work scope.