Topic: Data Science Product Development for Enhancing Student Retention in Higher Education Institutions

Student Name: FADELE, Michael Ayomide

Student ID: 239032281

Word Count: 2177

Date: 17/05/2023

1. Introduction

This project aims to create a data science product to improve student retention at higher education institutions. This product will detect at-risk students and intervene quickly using modern data analytics. The data will provide actionable insights that educational administrators and staff can utilize to make decisions. The scope of this project is the design, development, and deployment of a non-expert system that merges predictive analytics with user-friendly dashboards to convey data insights.

1.1 The study's importance

This study has the potential to change how higher education institutions deal with student retention. Student retention affects institution reputation, financial stability, and student success (Tinto, 2022; Bean, 2020). A technology that predicts and mitigates dropout risks promotes proactive educational tactics, individualized student support services, and an improved educational experience, all of which improve student outcomes. The individualized approach to interventions follows current educational trends toward adaptive learning environments that meet student needs (Kahu and Nelson, 2022).

1.2 Report Outline

The report will cover a wide range of project aspects, including:

Product Design: This part covers data source selection, application domain specification, and functional and non-functional requirements. It will also describe end-user-specific software architecture and use cases. This part also features the visualisation of the data used for the study.

Product Development: This section discusses software tools and platforms, as well as their

logic. The development process, testing methods, and user evaluation goals will be explained.

Project Management: The outline includes time management using a Gantt chart, data security risk assessment, and quality control procedures. Marketing and customer relationship management for the data science product will also be discussed.

Conclusion: The study's overview, major findings, and future research and product enhancement directions will be presented.

To ensure the project's real-world relevance and efficacy, each segment will incorporate current academic literature and industry best practices.

2. Product Design

2.1 Data Source and Theme Selection

This project used Kaggle's extensive data-centric project datasets. This dataset of student data contains demographics, academic performance, engagement metrics, and retention outcomes. This dataset is crucial because it represents the diverse and varied data that educational institutions handle. It was chosen for its expertise in constructing prediction models meant to identify students in danger of dropping out, improving retention measures. According to Delen (2020), extensive datasets like this improve the accuracy of student retention model predictions.

2.2 End-User and Application Domain Analysis

The administrative staff and educators at higher education institutions are the primary users of this product, with a focus on improving student retention. These stakeholders require data analytics solutions that simplify tasks without technical expertise. Early identification of at-risk students, insights into student attrition, and successful intervention strategy deployment

are required. The product's user-friendly interface provides actionable insights for quick and targeted interventions (Vemula and Moraes, 2024).

2.3 Functional and non-functional requirements

- **Functional Requirements**

1. Create a complete student profile using multiple data sources.
2. Use predictive analytics to predict dropouts.
3. Create customized reports and alerts for educators and administrators.

- **Non-functional Requirement**

With the minimum amount of training, the system will be user-friendly. It must efficiently process big datasets and provide real-time analytics. Given the sensitivity of educational data, the system must comply with data protection rules and maintain data integrity and confidentiality (Bertino and Ferrari, 2019).

2.4 Designing the Software Architecture

The proposed method uses microservices for flexible, scalable, and independent service deployment. Integrating multiple data sources and analytics tools without affecting system performance is possible with this architecture. The system includes a data ingestion module for data collection and preprocessing, a data storage system, an analytics engine for analysis and predictions, and a dashboard presentation layer. This architecture improves data-centric system scalability and maintainability, according to Vemulapalli (2023).

2.5 The Use-Case Specifications

Case 1: Predictive Alert Generation

Based on current academic performance and engagement metrics, an academic advisor receives a dashboard notice indicating a student may drop out. The system suggests student-specific interventions.

Case 2: Administrative Reporting

Administrative staff prepare retention metrics reports at the term's conclusion to assess current activities and identify areas for improvement.

Case 3: Custom Querying by Educators

Educators can query the system for course engagement and performance trends to customize teaching approaches to improve student engagement and success.

These use cases demonstrate the system's capacity to suit user needs, from strategic administrative planning to direct educator support, creating an educational environment that improves student retention (Norris et al., 2021).

2.6 Data Visualization

This section explores avisualizations of the dataset to uncover patterns and distributions crucial for understanding key variables like retention rates and demographic factors, including marital status and gender. By using pie charts, histograms, and bar graphs, complex data was transformed into clear, visually engaging formats. This not only simplifies data interpretation but also enhances insight into the drivers of student retention, supporting informed decision-making in educational contexts.

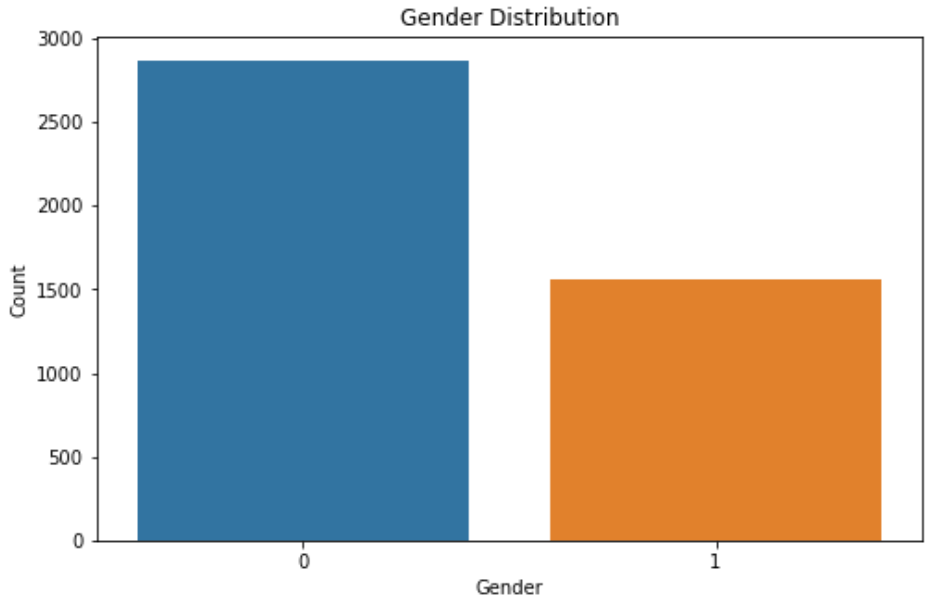


Fig. 1: Gender Distribution

The fig. 1 above shows a disparity between the numbers or proportions of males and females, with males being more prevalent or higher in number.

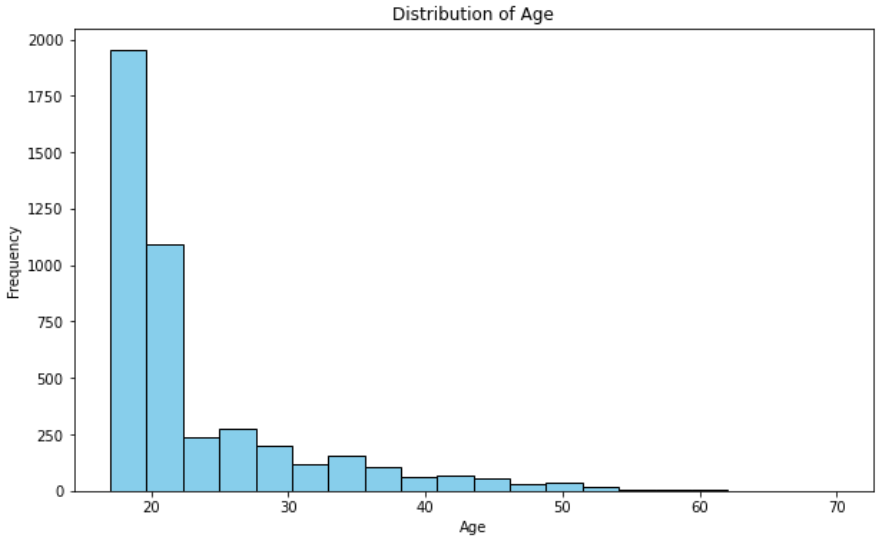


Fig. 2: Age Distribution

Fig. 2 above shows a histogram of students age distribution at enrolled with a series of bars that rapidly decrease in height from left to right. The tallest bar on the left suggests a high frequency for the lower end of the data spectrum.

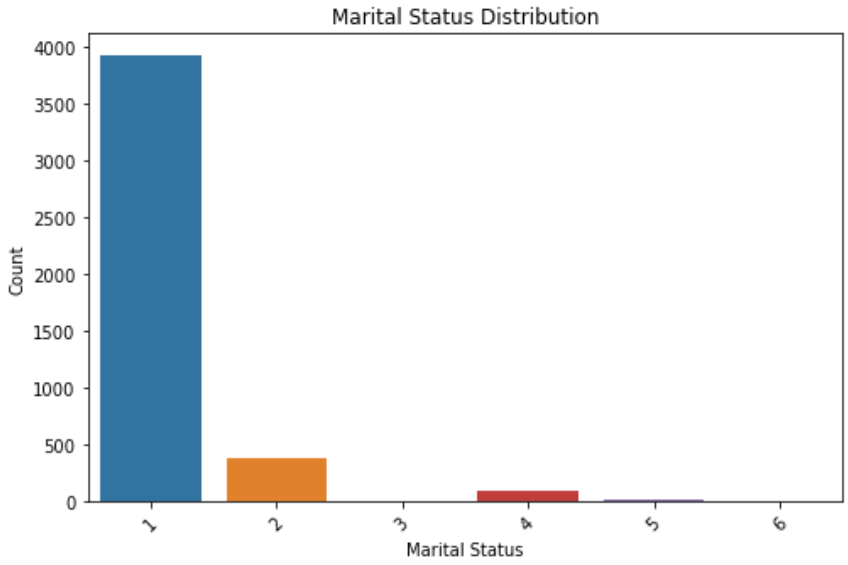


Fig. 3: Marital Status Distribution

The bar chart above categorizes marital status with varying heights: The tallest blue bar signifies singles as the most frequent group; the orange bar represents married individuals, noticeably fewer than singles; and the shortest red bar indicates those who prefer not to disclose their marital status, making it the least common category among respondents.

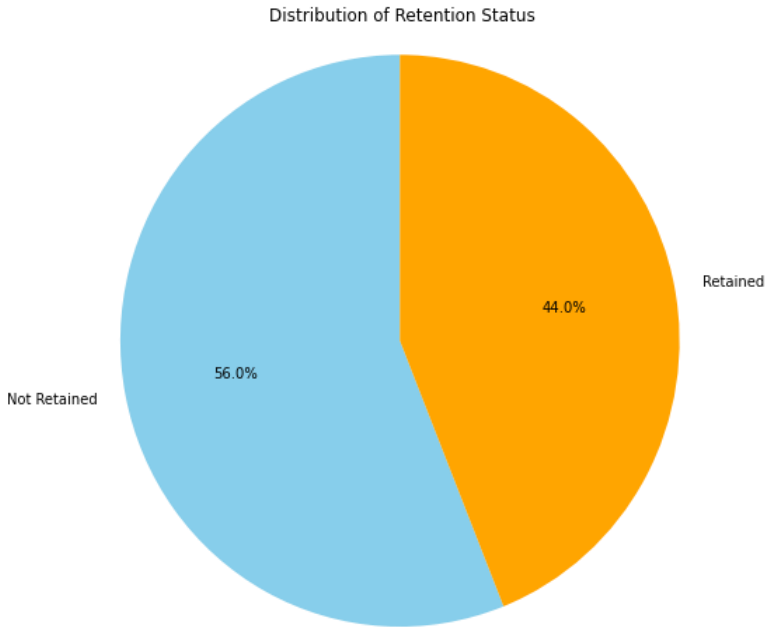


Fig. 3: Retention Status Distribution

The pie chart in fig. 3 above shows the distribution of the target variables, with 56% in the Not Retained category and 44% in the Retained category, indicating a relatively balanced split between the two groups.

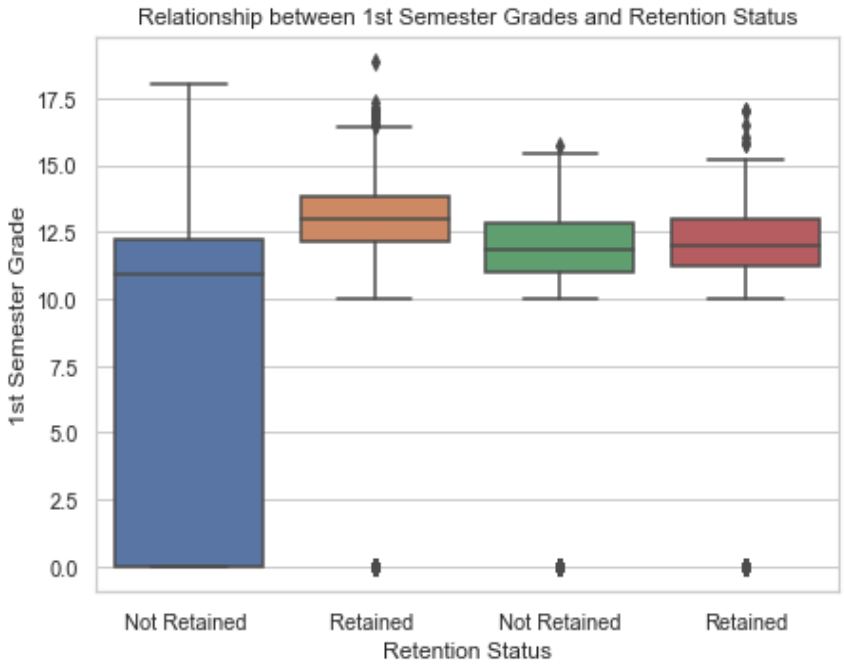


Fig. 4: Relationship Between 1st Semester Grades and Retention Status

The boxplot in Fig. 4 compares grades and retention status for the 1st semester, and it suggests that students who are retained (continued their studies) generally have higher grades than those not retained. This indicates that academic performance might be a significant predictor of retention.

3. Product Development

3.1 Software Tools and Platforms Selections

Python is the main programming language for this project, with Tableau for data visualization. Python is ideal for predictive analytics applications due to its adaptability and strength in data manipulation, statistical analysis, and machine learning. According to McKinney (2021), Python's vast libraries like Pandas, NumPy, and Scikit-learn provide sophisticated data analysis and machine learning tools needed to build predictive models. Streamlit was selected as the tool of choice for creating an interactive web application for model deployment. According to Khorasani et al (2022), Streamlit simplifies the process of building web applications for machine learning models and data visualization, making it accessible to users without extensive web development experience. With its user-friendly interface, rapid prototyping and deployment of predictive analytics tools becomes a breeze. This makes it easier for everyone to access and utilize these tools.

3.2 The Development Methodology

I used agile software development to create the student retention prediction model. Its flexibility, iterative nature, and focus on continual improvement made it suitable. Agile

methods such as Scrum enable regular feedback loops and incremental development, allowing for faster stakeholder feedback and insights (Seitsamo, 2021).

The first phase, requirement gathering, outlined project goals, and identified important stakeholders. Following this, I collected and pre-processed data using Python packages like Pandas. I used Matplotlib and Seaborn for exploratory data analysis (EDA) to find data patterns and linkages.

Consequently, I used Scikit-Learn to create a variety of machine learning models, such as Logistic Regression, K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, and Random Forest. According to Qumer et al. (2023), Agile iteration allowed for model development and enhancement based on evaluation outcomes.

3.3 System Testing Method

The student retention prediction system testing included unit testing, integration testing, and validation.

Unit Testing: I separately tested data pretreatment procedures and algorithms to ensure model accuracy. Python's unit test framework was used.

Integration Testing: I examined the interaction between various components, such as the integration of preprocessing stages with machine learning models, to ensure a fluid workflow and data consistency.

Cross-validation and validation: I assessed model performance using cross-validation techniques to ensure robustness and generalizability. Cross-validation evaluates the model on several data subsets, eliminating overfitting and ensuring dependability.

Deployment Testing: Finally, Streamlit deployed the model and performed end-to-end testing to ensure the web application worked properly without errors. User acceptance testing (UAT) verified the model's stakeholder performance.

3.4 The User Evaluation Plan

A qualitative and quantitative user evaluation plan is created to assess product effectiveness. End-user surveys will focus on usability, functionality, and product satisfaction. User satisfaction surveys help uncover product improvements and measure user engagement, according to Albert and Tullis (2023).

Usage data will also be analysed to see how often and efficiently product features are used. Login frequency, report production rates, and feature utilization will give quantifiable data that, when combined with qualitative input, will show how the product improves student retention methods.

4. Project Management

4.1 The Time Management Technique

Time management is essential for data science product delivery (Larson and Chang, 2020). The Gantt chart in fig. 4 below illustrates the timeline and milestones of this project. After project kick-off, the chart covers requirements gathering, design, development, testing, and deployment. Final product launch, end-user testing, and prototype completion are important milestones. This visual tool keeps team members informed of project deadlines and deliverables (Kerzner, 2022).

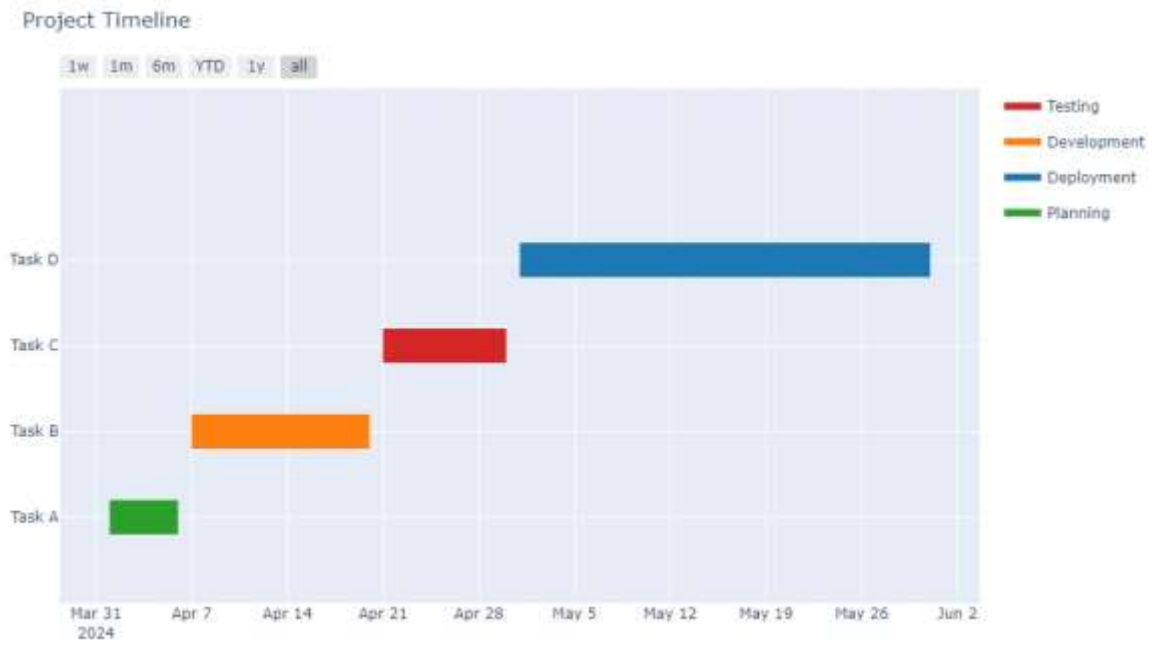


Fig. 4: Gantt Chart for Timeline and Milestone

4.2 Risk Assessment

Protecting personal information and guaranteeing data security are major risks in the development of a data science product. Breaches could compromise user trust and cause legal concerns. Unauthorized data access, data loss, and data leakage are common concerns, according to Cheng et al. (2021). Using strong encryption for data storage and transport, frequent security audits, the General Data Protection Regulation (GDPR), and HIPAA compliance will reduce these threats (Gordon et al., 2020).

4.3 Quality Control

Quality control in software development guarantees the product fulfils customer standards. Coding standards, peer reviews, and continuous integration ensure high-quality software development, according to Soares et al. (2022). Automatic testing was utilized extensively to

find bugs early. I also used Test-Driven Development (TDD), which promotes simple designs and confidence (Beck, 2022).

4.4 Management of Customer and User Relationships

Any product's success depends on end-user relations. To adapt communications and support, this project will use a CRM system to store user data and interaction history as suggested by (Peppers and Rogers, 2022). Regular updates and feedback loops will be built to ensure user satisfaction. Personalized training will improve user comfort and product proficiency, according to Homburg et al. (2021).

4.5 Marketing Strategy

The data science product's marketing approach will target potential clients, particularly educational institutions aiming to boost student retention. Content marketing will comprise blogs and white papers to highlight thought leadership and the product's value proposition (Kotler, 2019). To increase product launch awareness, social media campaigns will be used. We will also partner with educational experts who can promote the product in the industry (Kotler and Armstrong, 2020).

5. Conclusion

5.1 Summary of Key Points

This study recounts the development of an innovative data science product designed to increase student retention at higher education institutions. This product uses advanced predictive analytics to analyse Kaggle data on demographics, academic performance, and engagement levels. Completely Python-based for data manipulation and analysis, the product delivers deep insights into student behaviours for effective retention methods.

The product's user-centric design makes complex data accessible to administrators and

educators without deep technical understanding. Agile development promotes adaptation and continual improvement through iterative feedback and quick prototyping. This guarantees the product satisfies educational institutions' current demands and adapts to changing educational contexts.

Risk management was a primary priority, with strong tactics to protect personal information, assure data security, and maintain trust and regulatory compliance. Comprehensive system testing and user evaluation processes ensure the product meets the highest software quality and usability standards.

5.2 Future Directions

There are various promising ways to improve the product and expand its impact:

1. **Incorporation of Machine Learning Algorithms:** Using machine learning algorithms that can dynamically respond to changes in data over time, such as changes in educational trends or student demographics, could make the product more responsive and tailored to institutional needs.

2. **Integration of Natural Language Processing (NLP):** Numerical input from students and instructors could enrich the dataset and provide more nuanced student experiences and retention variables.

3. **Mobile Platform Compatibility:** Sending administrators and staff real-time notifications and updates on their mobile devices could enhance user engagement.

4. **Cross-Institutional Studies:** Testing the product in multiple education institutions could confirm its efficacy and increase its resilience and scalability.

In conclusion, this data science product is a beacon of educational technology innovation, built through rigorous analysis and a deep grasp of educational needs. Due to research and user input, it will continue to improve higher education student retention and success.

REFERENCES

Albert, B. and Tullis, T., (2023). *Measuring the user experience: collecting, analyzing, and presenting usability metrics*. Newnes.

Bean, J.P., (2020). Dropouts and turnover: The synthesis and test of a causal model of student attrition. *Research in higher education*, *12*, pp.155-187.

Beck, K., (2022). *Test driven development: By example*. Addison-Wesley Professional.

Bertino, E. and Ferrari, E., (2019). Big data security and privacy. In *A comprehensive guide through the Italian database research over the last 25 years* (pp. 425-439). Cham: Springer International Publishing.

Cheng, L., Liu, F. and Yao, D., (2021). Enterprise data breach: causes, challenges, prevention, and future directions. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *7*(5), p.e1211.

Delen, D., (2020). A comparative analysis of machine learning techniques for student retention management. *Decision Support Systems*, *49*(4), pp.498-506.

Gordon, L.A., Loeb, M.P. and Sohail, T., (2020). Market value of voluntary disclosures concerning information security. *MIS quarterly*, pp.567-594.

Homburg, C., Jozić, D. and Kuehnl, C., (2021). Customer experience management: toward implementing an evolving marketing concept. *Journal of the Academy of Marketing Science*, *45*, pp.377-401.

Jones, T.S. and Richey, R.C., (2020). Rapid prototyping methodology in action: A developmental study. *Educational Technology Research and Development*, *48*(2), pp.63-80.

Kahu, E.R. and Nelson, K., (2022). Student engagement in the educational interface: Understanding the mechanisms of student success. *Higher education research & development*, *37*(1), pp.58-71.

Keevers, T.L., (2019). Cross-validation is insufficient for model validation. *Joint and Operations Analysis Division, Defence Science and Technology Group: Victoria, Australia*.

Kerzner, H., (2022). *Project management metrics, KPIs, and dashboards: a guide to measuring and monitoring project performance*. John wiley & sons.

Khorasani, M., Abdou, M. and Hernández Fernández, J., (2022). Web Application Development with Streamlit. *Software Development*, pp.498-507.

Kotler, P. and Armstrong, G., (2020). *Principles of marketing*. Pearson education.

Kotler, P., (2019). *Marketing management*. Pearson Education India.

Larson, D. and Chang, V., (2020). A review and future direction of agile, business intelligence, analytics and data science. *International Journal of Information Management*, *36*(5), pp.700-710.

McKinney, W., (2021). Python for Data Analysis. OReilly Media. *Inc., Sebastopol*.

Myers, G.J., (2022). *The art of software testing*. John Wiley & Sons.

Norris, D., Baer, L., Leonard, J., Pugliese, L. and Lefrere, P., (2021). Action analytics: Measuring and improving performance that matters in higher education. *EDUCAUSE review*, *43*(1), p.42.

Peppers, D. and Rogers, M., (2022). *Managing customer experience and relationships: A strategic framework*. John Wiley & Sons.

Qumer, A. and Henderson-Sellers, B., (2023). A framework to support the evaluation, adoption and improvement of agile methods in practice. *Journal of systems and software*, *81*(11), pp.1899-1919.

Rodrigues, A.G., Demion, B. and Mouawad, P., (2019). *Master Apache JMeter-From Load Testing to DevOps: Master performance testing with JMeter*. Packt Publishing Ltd.

Seitsamo-Räsänen, S., (2021). Building an Agile Approach to Individual Feedback.

Soares, E., Sizilio, G., Santos, J., da Costa, D.A. and Kulesza, U., (2022). The effects of continuous integration on software development: a systematic literature review. *Empirical Software Engineering*, *27*(3), p.78.

Tinto, V., (2022). Reflections on student persistence. *Student Success*, *8*(2), pp.1-8.

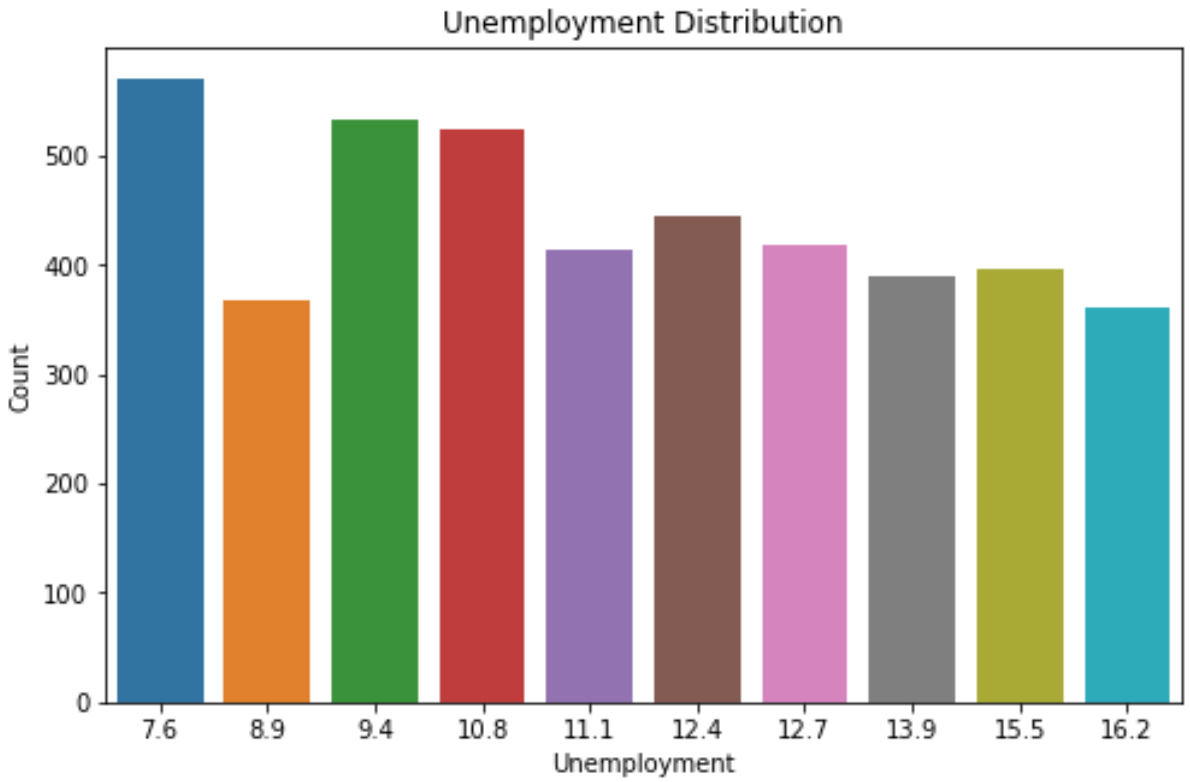
Vemula, S.R. and Moraes, M., (2024). Learning Analytics Dashboards for Advisors--A Systematic Literature Review. *arXiv preprint arXiv:2402.01671*.

Vemulapalli, G., (2023). Architecting for Real-Time Decision-Making: Building Scalable Event-Driven Systems. *International Journal of Machine Learning and Artificial Intelligence*, *4*(4), pp.1-20.

APPENDIX

```
import pandas as pd

# Load the dataset
df = pd.read_csv('desktop/dataset.csv')

# Display the first few rows of the dataframe to understand its structure
print(df.head())
# Display dataframe info to understand the columns and data types
print(df.info())

# Generate descriptive statistics
print(df.describe())
import matplotlib.pyplot as plt

# Since'age at enrollment' is a column in the dataset
plt.figure(figsize=(10, 6))
plt.hist(df['Age at enrollment'], bins=20, color='skyblue', edgecolor='black')
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
# Gender distribution
plt.figure(figsize=(8, 5))
sns.countplot(x='Gender', data=df)
plt.title('Gender Distribution')
plt.xlabel('Gender')
plt.ylabel('Count')
plt.show()
# Marital status distribution
```

```
plt.figure(figsize=(8, 5))
sns.countplot(x='Marital status', data=df)
plt.title('Marital Status Distribution')
plt.xlabel('Marital Status')
plt.ylabel('Count')
plt.xticks(rotation=45)  # Adjust rotation depending on label length
plt.show()
# Unemployment distribution
plt.figure(figsize=(8, 5))
sns.countplot(x='Unemployment rate', data=df)
plt.title('Unemployment Distribution')
plt.xlabel('Unemployment')
plt.ylabel('Count')
plt.show()
```

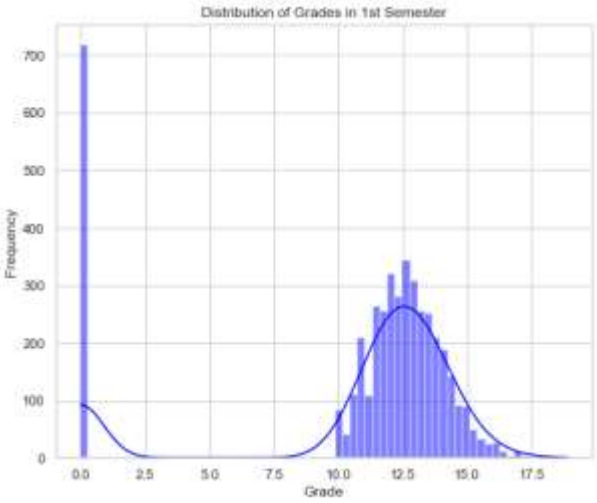


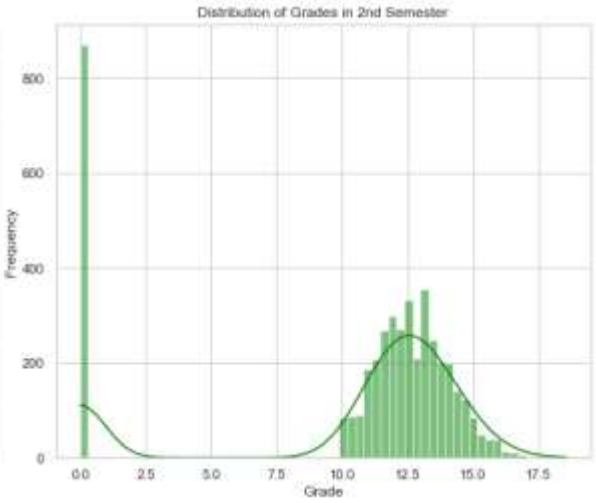

```
# Load the dataset
df = pd.read_csv('desktop/dataset.csv')
```

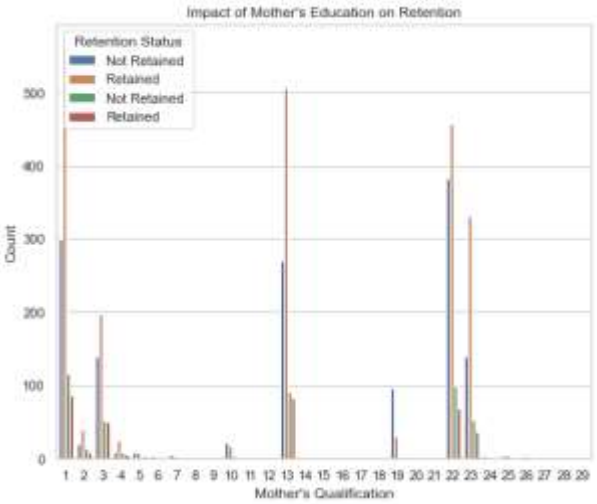
```
# Print unique values in the 'target' column
print("Unique target values:", df['Target'].unique())
# Filter to include only 'Retained' and 'Not Retained'
df = df[df['Target'].isin(['Retained', 'Not Retained'])]

# Check the counts again
print("Counts of target categories:", df['Target'].value_counts())
# Create a pie chart with corrected data
target_counts = df['Target'].value_counts()
plt.figure(figsize=(8, 8))
plt.pie(target_counts, labels=target_counts.index, autopct='%1.1f%%', startangle=90,
colors=['skyblue', 'orange'])
plt.title('Distribution of Retention Status')
plt.axis('equal')
plt.show()
pip install plotly
import plotly.figure_factory as ff
import datetime

# Define a list of dictionaries, each representing a task in the Gantt chart
tasks = [
    {"Task": "Task A", "Start": "2024-04-01", "Finish": "2024-04-06", "Resource":
"Planning"},
    {"Task": "Task B", "Start": "2024-04-07", "Finish": "2024-04-20", "Resource":
"Development"},
    {"Task": "Task C", "Start": "2024-04-21", "Finish": "2024-04-30", "Resource": "Testing"},
    {"Task": "Task D", "Start": "2024-05-01", "Finish": "2024-05-31", "Resource":
"Deployment"}
]

# Create a Gantt chart
```

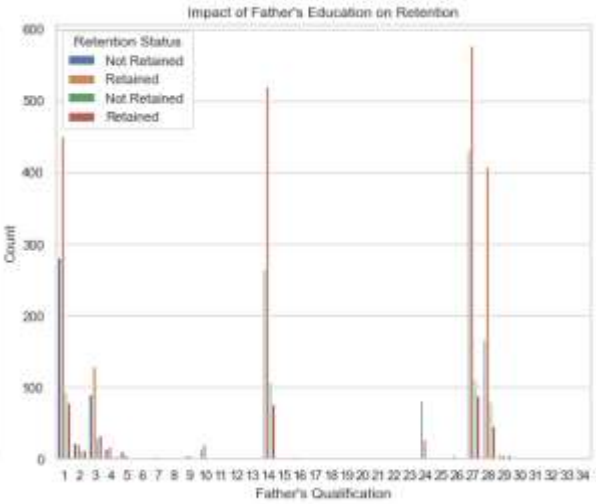
```
fig = ff.create_gantt(tasks, index_col='Resource', title='Project Timeline',
show_colorbar=True,
                      showgrid_x=True, showgrid_y=True)
fig.show()



import matplotlib.pyplot as plt
import seaborn as sns

# Setting the aesthetic style of the plots
sns.set(style="whitegrid")

# Creating visualizations to explore relationships in the data

# 1. Distribution of grades for the 1st and 2nd semester
fig, ax = plt.subplots(1, 2, figsize=(14, 6))

sns.histplot(data=data, x="Curricular units 1st sem (grade)", kde=True, ax=ax[0],
color="blue")
ax[0].set_title('Distribution of Grades in 1st Semester')
ax[0].set_xlabel('Grade')
ax[0].set_ylabel('Frequency')

sns.histplot(data=data, x="Curricular units 2nd sem (grade)", kde=True, ax=ax[1],
color="green")
ax[1].set_title('Distribution of Grades in 2nd Semester')
ax[1].set_xlabel('Grade')
ax[1].set_ylabel('Frequency')

plt.tight_layout()
```

```
# 2. Relationship between 1st semester grades and retention status
plt.figure(figsize=(7, 5))
sns.boxplot(x="Target", y="Curricular units 1st sem (grade)", data=data)
plt.title('Relationship between 1st Semester Grades and Retention Status')
plt.xlabel('Retention Status')
plt.ylabel('1st Semester Grade')

plt.show()

# 3. Impact of mother's and father's education on retention
fig, ax = plt.subplots(1, 2, figsize=(14, 6))

sns.countplot(x="Mother's qualification", hue="Target", data=data, ax=ax[0])
ax[0].set_title("Impact of Mother's Education on Retention")
ax[0].set_xlabel("Mother's Qualification")
ax[0].set_ylabel("Count")
ax[0].legend(title='Retention Status')

sns.countplot(x="Father's qualification", hue="Target", data=data, ax=ax[1])
ax[1].set_title("Impact of Father's Education on Retention")
ax[1].set_xlabel("Father's Qualification")
ax[1].set_ylabel("Count")
ax[1].legend(title='Retention Status')

plt.tight_layout()
plt.show()
```


Distribution of Grades in 1st Semester
Frequency
Grade


Distribution of Grades in 2nd Semester
Frequency
Grade


Impact of Mother's Education on Retention
Retention Status
Not Retained
Retained
Not Retained
Retained
Count
Mother's Qualification


Impact of Father's Education on Retention
Retention Status
Not Retained
Retained
Not Retained
Retained
Count
Father's Qualification