

Information Technology Project Management

Graduate Portfolio

Yegna Sai Anand Akella

1226108184

Dr. Tatiana Walsh

Arizona State University

Table of Contents

<u>Resume.....</u>	<u>3</u>
<u>Reflection.....</u>	<u>5</u>
<u>Overview.....</u>	<u>8</u>
<u>Accomplishments.....</u>	<u>10</u>
<u>Accomplishment1.....</u>	<u>10</u>
<u>Accomplishment2.....</u>	<u>18</u>
<u>Accomplishment3.....</u>	<u>23</u>
<u>References.....</u>	<u>28</u>

Resume

Yegna Sai Anand Akella

Education

Master of Science in Information Technology Graduating December 2024
(Information Technology Project Management)

Arizona State University, Tempe, Arizona, USA GPA: 3.96 /4

Technical Concepts

Data Visualization, Analyzing Big Data, Advanced DB Management Systems, Info Systems Development, Advanced Information Systems Security.

Technical Skills

Programming : Java, Springboot, Python, Shell, JavaScript.

Coding Tools : IntelliJ, MySQL, GitHub, Compass, AWS, Tableau.

Web Technologies : Node.js, Angular.js, Express.js, HTML, CSS.

Cloud Technologies : Amazon Web Services, Microsoft Azure, Google Cloud Platform.

Databases : SQL, MongoDB, DynamoDB.

Professional Work Experience

Product Support Specialist - GradRight, India May 2022 – Dec 2022

- Created RESTful APIs with BitBucket version control to process client requests.
- Used Splunk and dashboards to troubleshoot production issues and monitor the applications.
- Closely collaborated with cross-functional teams to enhance workflow and effective solutions.
- Using Spring Boot, APIs were developed to access and modify vital student data, facilitating effective information retrieval according to predetermined standards.

- JUnit test suites were created to ensure the robustness and dependability of the application.
- Jenkins was used to implement a CI/CD pipeline that streamlined software deployment.

Technical Support Engineer – Joveo Technologies, India

Nov 2021 – Apr 2022

- Created RESTful APIs with AWS serverless architecture, utilizing Node.js, AWS Lambda, and API Gateway.
- Reduced manual efforts by 75% through automation of repetitive tasks using Node.js and SQL, which improved accuracy and data integrity while minimizing manual data entry.
- Built highly scalable, low-latency, fault-tolerant, and high-performance architectural solutions for customer-facing web applications, demonstrating proficiency with DSA.
- Resolved complex functional issues within applications using Node.js, achieving maximum test coverage.
- Contributed significantly to the debugging of a web application using Node.js, RESTful APIs.

Production Software Engineer – Oracle Cerner Healthcare, India

Jul 2018 – Aug 2021

- Improved the implementation of server-side code, enabling patients to securely access confidential data.
- Developed and contributed to workflows that assist hospitals digitize each stage of the healthcare process for patients, doctors, and pharmacists using Java, Maven, SQL, Git and Agile methodologies.
- Created Docker images, coordinated the containerized application using Kubernetes.
- Proved ability to solve complicated technological problems by finding solutions, such as indexing, query, rewriting, and query plan analysis, which optimize database performance.
- Identified, resolved, and fixed a variety of functional, database, and workflow problems.
- Initiated the process of documenting workflows and trained colleagues on products, procedures, and services.
- Worked on fixing various bugs in existing solutions, which enhanced my knowledge of Java.

Reflection

My Graduate Experience

I concentrated on information technology during my graduate studies at Arizona State University. I was able to learn a great deal about database administration, cloud computing, and software engineering, among other aspects of IT, because to this curriculum. The projects and curriculum gave students the chance to learn and develop the fundamental abilities needed to create effective systems.

I now have a thorough understanding of enterprise-level software solutions and contemporary cloud-based architectures thanks to my emphasis on information technology. I gained a strong technical foundation that allowed me to stay up to date with industry changes and best practices while applying IT principles successfully.

Focus Area and Academic Journey

Cloud Architecture was my favorite course that I took. This course offered priceless knowledge about developing, deploying, and refining cloud-based systems. It provided a practical way to interact with cloud infrastructure, which improved my knowledge of service management. My confidence in designing dependable and scalable cloud systems increased as a result of the course's ability to apply theoretical knowledge to real-world applications. My enthusiasm for cloud computing was stoked by the focus on practical applications and critical thinking, which equipped me for the demands of these professions.

Academic Accomplishments

Achieving a GPA of 3.96 during my graduate studies was one of my major academic achievements. This accomplishment demonstrates my devotion to both academic brilliance and material mastery. It took a disciplined approach to juggling assignments, projects, and extra learning activities in order to maintain a high GPA.

Completing many cloud computing and big data initiatives is another noteworthy achievement. Through these initiatives, I was able to have practical expertise in working with big datasets and developing scalable cloud solutions. I improved my technical knowledge and problem-solving skills through these experiences, equipping me to face problems in the industry.

Post-Graduation Aspirations:

I'm applying right now for a job as a software development engineer. I believe I'm a good fit for this position because I have five years of expertise creating scalable software solutions and maximizing performance. My practical knowledge of cloud technologies and my technical expertise in languages like Java, Spring Boot, and JavaScript meet the criteria of this role.

I want to use my experience with automation, cloud architecture, and agile approaches in this position to help create software that is of a high caliber. My background in performance optimization, debugging, and teamwork has given me the abilities I need to succeed in this role.

Joining the Alumni Network:

Yes, please add me to the alumni group for information technology. I would be able to stay up to date on industry advancements and other experts by joining this club, which would also offer beneficial networking chances. After graduation, I'm excited to interact with other graduates and

give back to the community. To add me to the group or for any other correspondence, please contact me at anandakella777@gmail.com.

Overview

The main projects I finished throughout my two years at Arizona State University are displayed in my graduation portfolio. Being a part of such a distinguished organization makes me proud. This portfolio's three primary projects showcase my accomplishments and demonstrate my development. My goal is to become a Software Development Engineer (SDE). Together with my schoolwork, these projects have helped me better understand software development and have strengthened my resolve to work in this industry.

IFT 530: Advanced Database Management System - Soccer Database Management

To manage and store soccer-related data, including player information, team details, match statistics, and performance records, this project concentrated on building an extensive database. An effective system for arranging data from various sources and making it available for analysis was the aim.

Players, teams, coaches, and player health are among the important entities that are tracked by the database. For instance, the performance entity keeps track of statistics like games played, victories, and injuries, while the player entity contains information like name, team, and nation. Important questions like which coach has won the most trophies, which team has the most players from a particular nation, and which team has the most fans were addressed by the queries.

In addition to using NoSQL techniques for bucket creation and data insertion, the project used SQL for data creation, insertion, and retrieval. All things considered, this system offers teams, soccer analysts, and supporters' insightful information about player performance, team accomplishments, and other soccer-related metrics.

IFT 533: Data Visualization – Healthcare Analytics Dashboard for Patient Insights.

To give healthcare professionals, administrators, insurers, and researchers useful information about patient demographics, medical conditions, doctor visits, insurance data, and other topics, this project creates an extensive, interactive dashboard. The dashboard, which is based on a rich dataset, shows

important healthcare metrics such as trends in test results, top doctors by patient visits, insurance spending patterns, and age distribution across medical conditions.

The dashboard enables users to investigate important questions like differences in medical conditions by age and gender, the frequency of prescribed medications, and insurance provider comparisons using a variety of visual elements like bar charts, line graphs, and pie charts. It is an effective tool for data-driven decision-making because of its interactive features, which allow for smooth navigation across multiple patient attributes and dynamic filtering by medical condition. The dashboard supports policy planning, improves patient care, and optimizes resource allocation by providing healthcare organizations with these insights.

IFT 511: Analyzing Big Data – Credit Risk Classification with Decision Tree Modeling

Using the Credit Card Default dataset from the UCI Machine Learning Repository, this project, Credit Risk Classification with Decision Tree Modeling, creates a decision tree model to forecast the risk of credit card default. Records from Taiwanese credit card users between April and September 2005 are included in the dataset, which includes information on credit limit, demographics, and payment history. To ensure data quality for analysis, the project starts by cleaning the data to eliminate missing values. The dataset is divided to balance classes in training and testing using a five-fold Stratified K-Fold cross-validation, which improves the precision of performance evaluation.

Two impurity measures, Gini and Entropy, are used to train the model using decision trees, and GridSearchCV optimizes the parameters. The best impurity measure is determined by averaging accuracy scores across folds; the results indicate that the Entropy measure is the most successful in forecasting default risk. This project demonstrates a comprehensive approach to data mining, covering everything from planning to model evaluation and selection.

Accomplishments

Accomplishment 1 - (IFT 530 - Advanced Database Management Systems)

Project Title: Soccer Database Management 2023

Establishing a centralized platform for gathering, storing, and evaluating soccer-related data from several sources is the goal of the Soccer Database Creation and Data Storage initiative. This project collects information from media outlets, sporting groups, and official sports websites.

After being gathered, the data is arranged and kept in a single database, giving users quick access to a variety of soccer information, such as event schedules, standings, scores, and statistics. It also contains comprehensive data about specific players and teams, including player biographies, career statistics, team rankings, and histories.

The project uses data validation methods including cleaning and normalization to preserve the data's accuracy and dependability. Data may be retrieved quickly and effectively thanks to the database's easy accessibility, scalability, and security features. This is supported by a strong database management system, which provides effective database management along with necessary capabilities for security, scalability, and configuration.

By providing a unified platform for accessing and analyzing soccer data, this project seeks to transform soccer analysis for researchers, analysts, fans, and sports organizations. It could improve performance analysis, increase fan involvement, and facilitate better sport decision-making.

Importance:

Maintaining a soccer database is essential for several key reasons:

Historical Insights and Trend Analysis

Tracking and analyzing historical data is a major advantage of keeping a soccer database. The database provides insightful information about historical patterns, accomplishments, and performances—all of which are essential for comprehending how teams and players have changed

over time. A greater comprehension of the game and its strategic benefits results from analysts and teams being able to make well-informed decisions based on historical trends and data-driven insights.

Data-Driven Strategy and Predictive Modeling:

A basis for statistical analysis that can inform game plans is provided by soccer databases. Statisticians can develop predictive models that estimate future events, player performance indicators, and tactical plans by examining the data that has been gathered. This data gives coaches and teams the tools they need to make strategic choices based on solid, data-driven insights, increasing team performance and success rates.

Enhancing player performance:

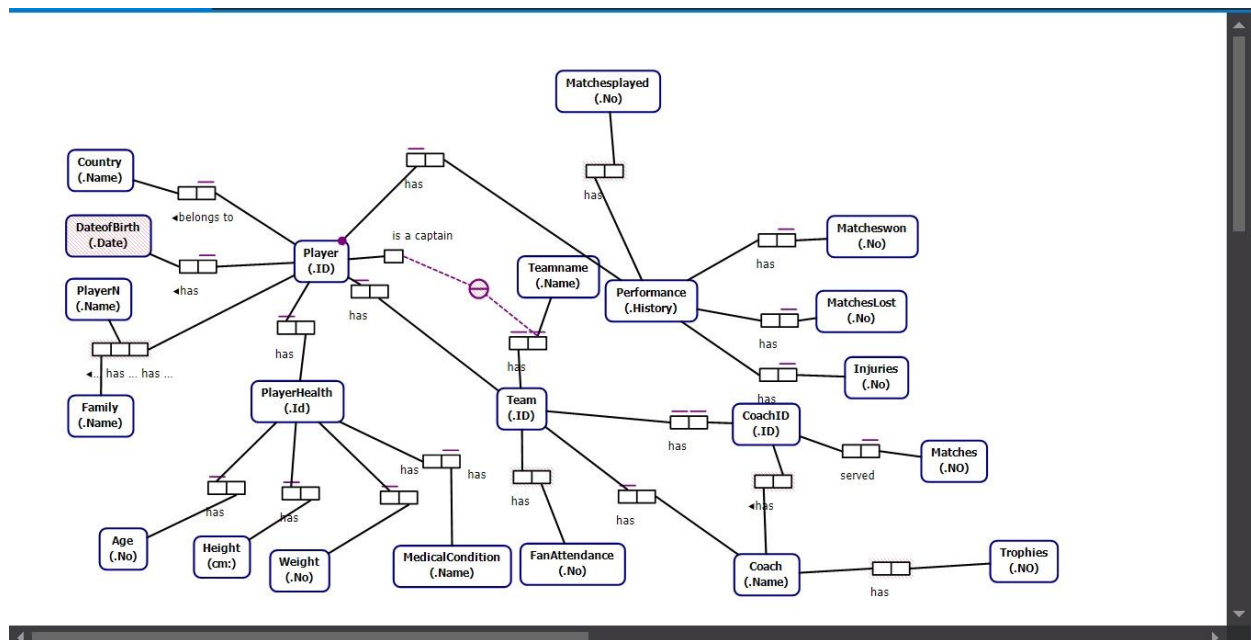
By monitoring each player's statistics and comparing them to team-wide standards, a thorough soccer database is essential for regulating each player's performance. This enables coaches to make specialized training and development plans and pinpoint the precise areas in which athletes need to improve. Improved on-field performance and a more cohesive team chemistry can result from effective player management based on data-driven insights.

Simplifying procedures and increasing productivity:

Complex duties like player management, tactical planning, and performance analysis are made easier by the integrated database system, which also saves staff members time and lessens their manual workload. The database helps speed numerous operational duties inside sports organizations by effectively organizing vast volumes of data, freeing up staff members to concentrate more on strategic projects and increasing overall productivity.

The integrated system supports complex tasks like performance analysis and player management. By organizing data efficiently, it saves time and reduces manual workloads for staff members.

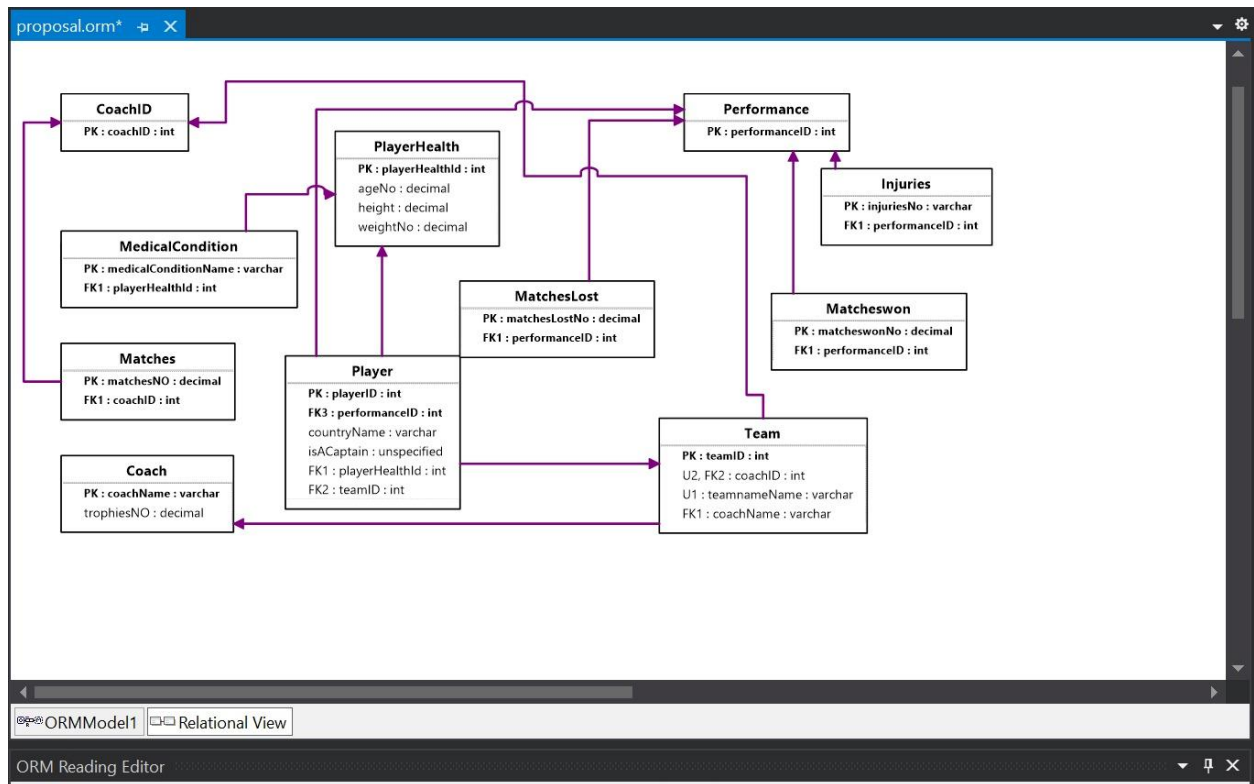
The ORM Diagram:



Our soccer database's project management tool includes a number of entities, such as players, teams, coaches, and performance indicators. Specific properties describe each entity:

- **Player:** This part contains attributes like nation, team affiliation, first and last names, and player ID. Teams and players continue to have a many-to-one relationship.
- **Team:** The team ID, name, coach, nation, and attendance statistics of the fans are among the attributes for this entity.
- **Coach:** Coach ID, full name, team affiliation, age, number of awards won, and matches coached make up this entity.
- **Performance:** This keeps tabs on player performance indicators, such as the quantity of games played and victories.
- **Player Health:** Medical conditions and other health-related features are included in this entity.

Relational Database Diagram:



Several tables connected by the primary and foreign keys make up the relational structure:

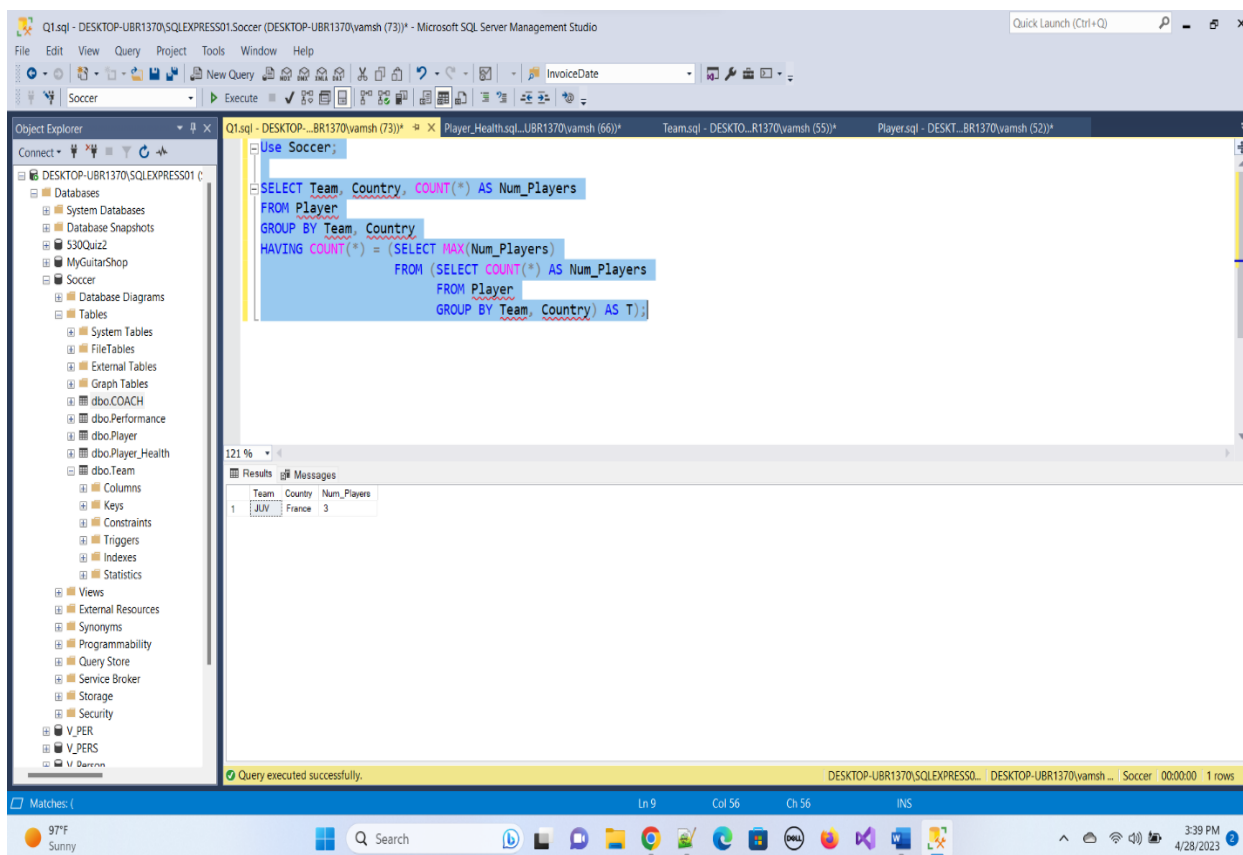
- **Player Table:** Contains player data, with player ID serving as the primary key.
- **Team Table:** Includes information on teams, with the team ID serving as the main key.
- **Coach Table:** Holds coach-related data, with coach ID acting as the main key.
- **Performance Table:** keeps track of player performance metrics.
- **Player Health Table:** keeps track of players' health information.

SQL Implementation

SQL Server Management Studio was used to create the database:

- Tables with the required constraints were created for every entity.

- To extract certain insights, like identifying top-performing players or figuring out which teams had the most fans, SQL queries were created.
- To automate changes and preserve data integrity, triggers and stored procedures were used.



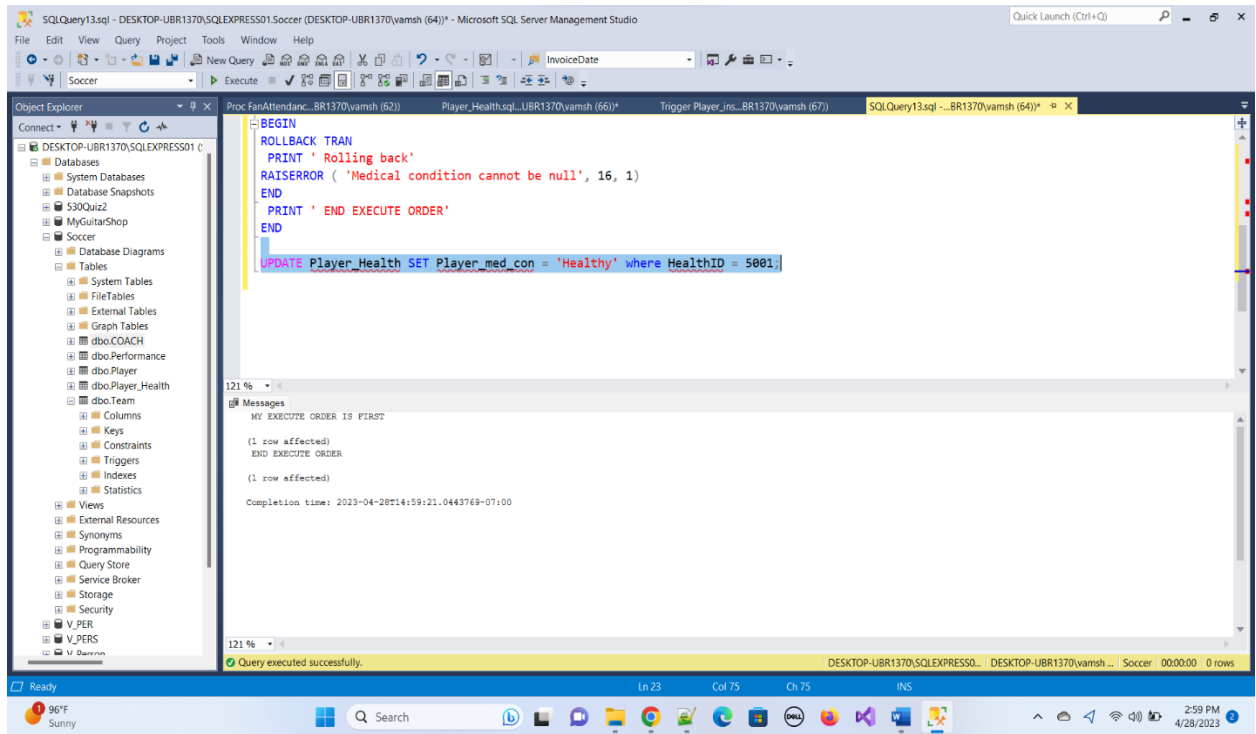
The screenshot displays the Microsoft SQL Server Management Studio (SSMS) interface. The main window shows a SQL query in the 'Query Editor' pane, which is titled 'Q1.sql - DESKTOP-UBR1370\SQLEXPRESS01 Soccer (DESKTOP-UBR1370\vamsh (73))'. The query is as follows:

```
USE Soccer;
SELECT Team, Country, COUNT(*) AS Num_Players
FROM Player
GROUP BY Team, Country
HAVING COUNT(*) = (SELECT MAX(Num_Players)
FROM (SELECT COUNT(*) AS Num_Players
FROM Player
GROUP BY Team, Country) AS T);
```

The 'Object Explorer' pane on the left shows the database structure, including 'Databases', 'System Databases', 'Database Snapshots', 'S30Quiz2', 'MyGuitarShop', 'Soccer', 'Database Diagrams', 'Tables', 'System Tables', 'FileTables', 'External Tables', 'Graph Tables', 'dbo.COACH', 'dbo.Performance', 'dbo.Player', 'dbo.Player_Health', 'dbo.Team', 'Columns', 'Keys', 'Constraints', 'Triggers', 'Indexes', 'Statistics', 'Views', 'External Resources', 'Synonyms', 'Programmability', 'Query Store', 'Service Broker', 'Storage', 'Security', 'V_PER', 'V_PERS', and 'V_Datam'. The 'Results' pane at the bottom shows the output of the query, which is a single row with the following data:

Team	Country	Num_Players
1	JUV	France

The status bar at the bottom indicates that the query was executed successfully, and the results show 1 row.



NoSQL Implementation

Additionally, a Couchbase NoSQL database was created:

- Buckets were setup for entities such as players and teams.
- Data was indexed to facilitate efficient querying using SQL++.
- Without the limitations of a set schema, flexible data management was made possible by the NoSQL architecture.

The screenshot shows the C2 Analytics interface with a query editor and results. The query editor contains a SQL query that joins a Player table with a Team table. The query results are displayed in JSON format, showing a list of players and their associated team information.

Query Editor

```
1 SELECT *
2 FROM Player as p
3 INNER JOIN Team as t
```

Query Results

```
1 {
2   {
3     "p": {
4       "Country": "Argentina",
5       "Family name": "Messi",
6       "Teamname": "PSG",
7       "id": "1227",
8       "name": "Lionel"
9     },
10    "t": {
11      "Coach": "Tow",
12      "Country": "France",
13      "Fan Attendance in percentage": 89,
14      "Teamname": "PSG",
15      "id": "1227"
16    }
17  },
18  {
19    "p": {
20      "Country": "Portugal",
21      "Family name": "Ronaldo",
22      "Teamname": "MUTD",
23      "id": "1229",
24      "name": "Cristiano"
25    },
26    "t": {
27      "Coach": "Mike",
```

Analytics Scopes, Links, & Collections

- Coach_default
- Local
- Customers_default
- Local
- Default
- Local
- Health_default
- Local
- Orders_default
- Local

The screenshot shows the C2 Analytics interface with a query editor and results. The query editor contains a SQL query that selects all columns from the Team table, ordered by Fan Attendance in percentage. The query results are displayed in table format, showing a list of teams and their associated fan attendance information.

Query Editor

```
1 select * from Team order by "Fan Attendance in percentage"
```

Query Results

Coach	Country	Fan Attendance in percentage	Teamname	id
Tom	France	89	PSG	1227
Mike	UK	81	MUTD	1229
Jason	Brazil	72	Juventus	1230
Robert	Sweden	78	Arsenal	1231

Analytics Scopes, Links, & Collections

- Coach_default
- Local
- Customers_default
- Local
- Default
- Local
- Health_default
- Local
- Orders_default
- Local

Summary

This project demonstrated how well SQL and NoSQL databases can be integrated to manage data pertaining to soccer. While the NoSQL database offered the flexibility required to handle unstructured data, the SQL database made structured data storage easier with pre-established schemas. In-depth data analysis was made possible by this combination, enabling more dynamic queries and real-time insights. Consequently, it improved sports organizations' ability to make decisions and provided a scalable system to handle future data growth and evolving analytics requirements.

Accomplishment 2 - (IFT 533: Data Visualization and Reporting for IT)

Project Title: Healthcare Analytics Dashboard for Patient Insights

Introduction

A comprehensive tool for visualizing patient health data, medical conditions, doctor visits, and insurance information is the Healthcare Dashboard. By offering a comprehensive perspective of important healthcare metrics through interactive visualizations, it seeks to support healthcare professionals in making data-driven decisions.

Dashboard Overview

The dashboard incorporates a number of plots that let users examine insights like:

- Distribution of ages by medical conditions
- Top physicians according to patient visits
- Prevalence of medical conditions by gender
- Trends in insurance coverage and admissions over time

The dashboard's intuitive design facilitates interaction and allows for smooth transitions between various visual components. This flexibility enables researchers, analysts, and healthcare administrators to improve patient care and extract useful insights.

Dataset Details:

Rich patient-specific data, such as demographics, medical history, treatment specifics, and billing amounts, are included in the healthcare dataset. Important characteristics include:

- Age: Constant numbers
- Gender: Representation in categories
- Doctor: Health care providers' names
- Names of insurance companies that offer insurance
- Medical Condition: Summaries of the health conditions of the patients

In-depth investigation of patient journeys and healthcare trends is supported by the inclusion of temporal data on admission and discharge dates, which enhances the dataset.

Key Insights and Visualizations:

The control panel makes it possible to investigate important queries, such as:

1. Distribution of ages by medical conditions
2. Changes in test results over time for particular conditions
3. Comparisons of medication frequency for different medical conditions
4. Annual trends in admissions
5. An examination of insurance companies' finances

Visualizations include:

- Line graphs that display admissions over a six-year period
- Bar charts: comparing the distribution of ages by type of admission
- Pie charts: Showing the distribution of blood groups for particular conditions
- Comparing the expenditures of insurance providers using stacked bar graphs

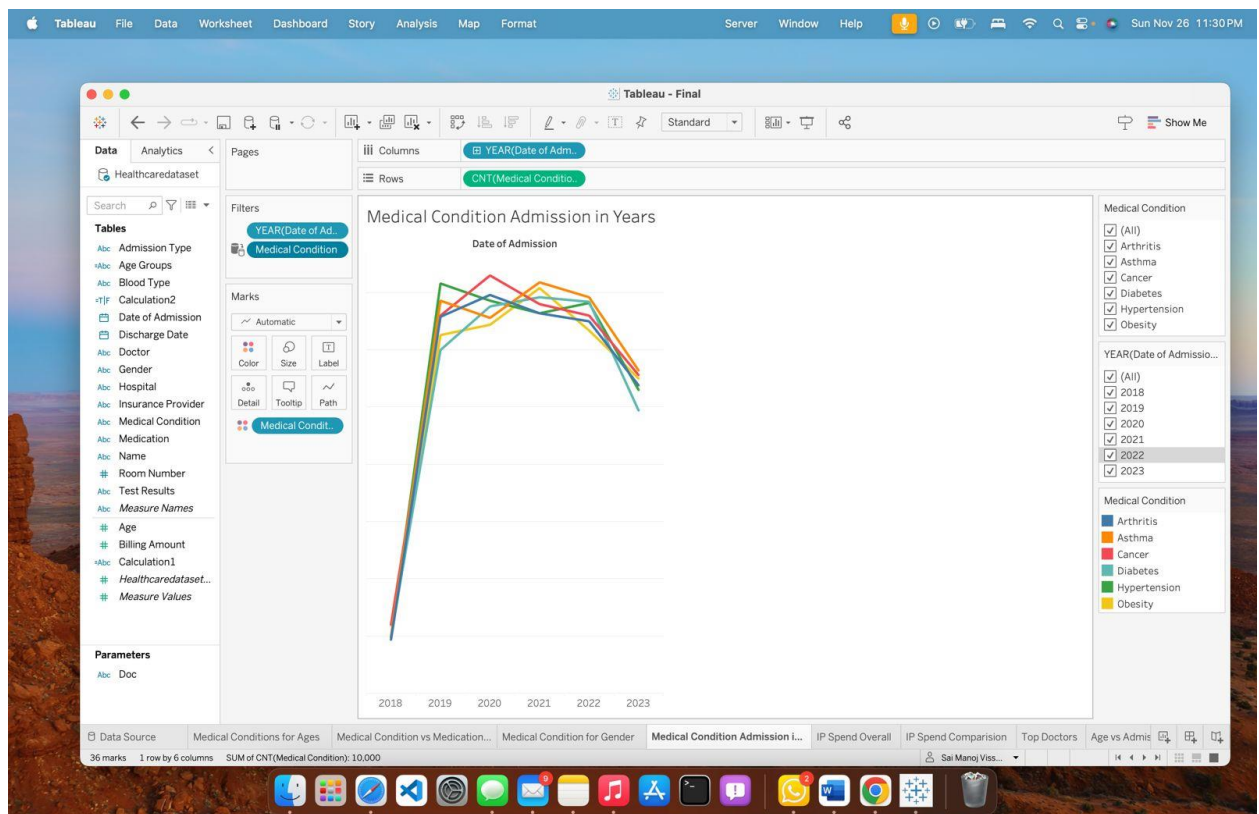
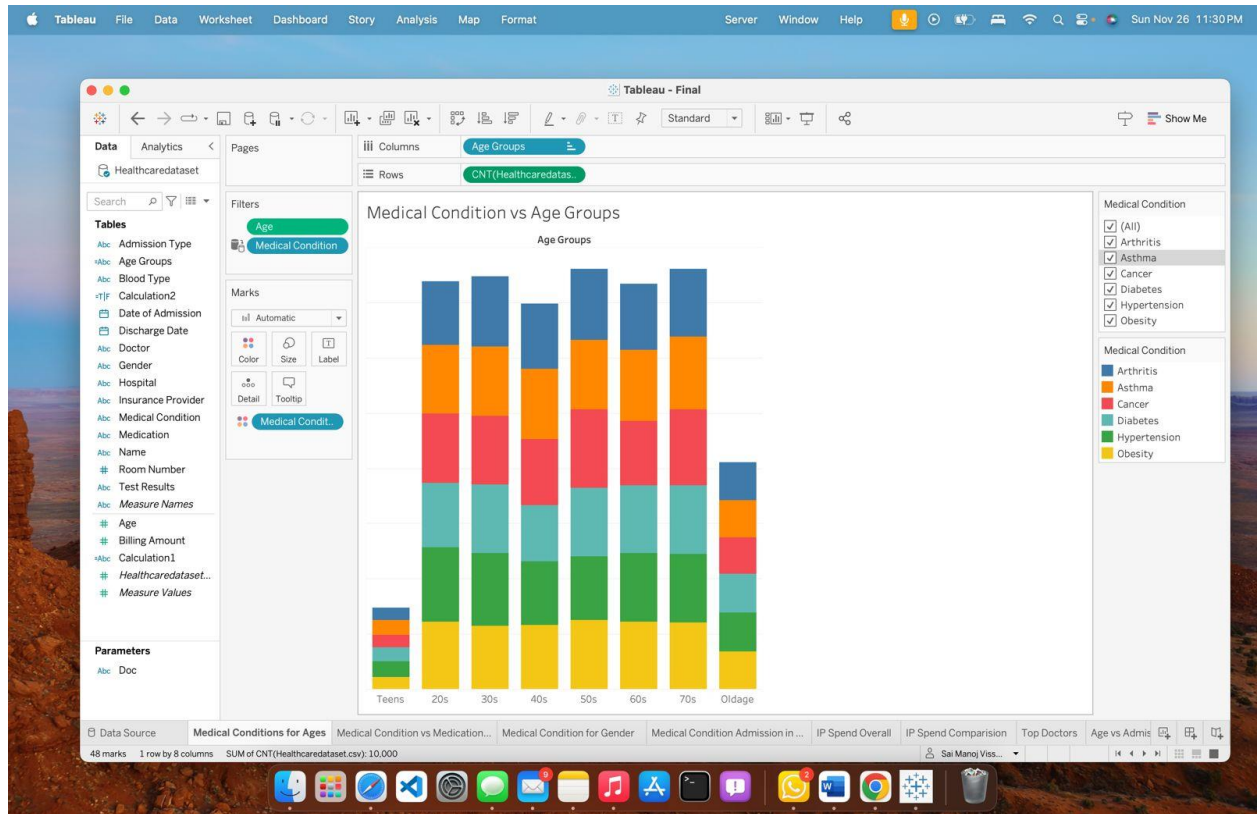
Interactivity Features:

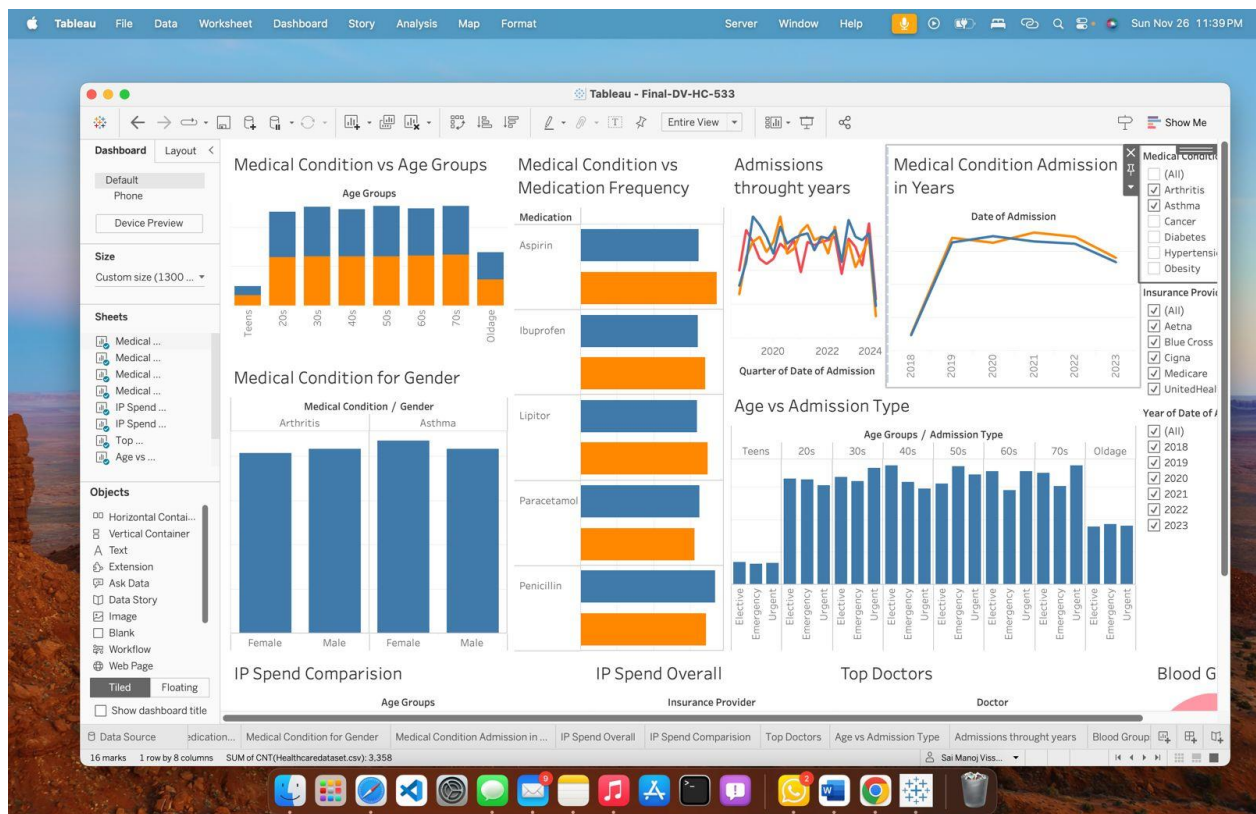
Customized analyses are made possible by interactive controls that let users filter data by medical conditions. Deeper insights into particular patient demographics and healthcare trends are made possible by this functionality, which improves the user experience.

Final Dashboard:

Click on link below to access the public dashboard. For healthcare administrators and analysts, the dashboard is an effective tool that supports strategic decision-making and raises the standard of healthcare services.

<https://public.tableau.com/app/profile/sai.manoj.vissavajhula/viz/Final-DV-HC-533/Dashboard1?publish=yes>





Summary

The goal of the Healthcare Dashboard project is to give users a thorough visual depiction of healthcare data so they can extract useful information about insurance coverage, doctor visits, medical conditions, and patient health. This interactive dashboard, which was created as a component of the IFT 533 course under the direction of Dr. Asmaa Elbadrawy, combines a number of plots and metrics to enable dynamic investigation of important healthcare issues.

The dashboard answers important questions like how age groups are distributed across medical conditions, how effective various medications are, and trends in hospital admissions over time by using a rich dataset that includes patient demographics, medical histories, and billing information. Pie charts, line graphs, and bar charts are important visualizations that are made for easy navigation and interaction.

The project gives healthcare administrators, analysts, and researchers the tools they need to improve patient care, make better decisions, and pinpoint areas that need more research. In the end, the Healthcare Dashboard is an invaluable tool for comprehending healthcare patterns and enhancing the provision of services in medical environments.

Accomplishment 3 - (IFT 511: Analyzing Big Data)

Project Title: Clustering Phoenix Schools for Improved Equity and Resource Allocation

Overview

The goal of this project is to use the Credit Card Default dataset from the UCI Machine Learning Repository to examine and forecast the risk of credit card default by utilizing machine learning techniques. The dataset, which includes a variety of characteristics like credit limits, payment histories, and demographic data, documents the financial activity of Taiwanese credit card users between April and September 2005. In order to help financial institutions make data-driven decisions that improve their lending processes and risk management strategies, this project uses sophisticated analytical techniques to find trends that increase credit risk.

The goal of our team is to create a prediction model that correctly categorizes credit card users according to their propensity to miss payments. We seek to offer practical insights that might help financial institutions customize their credit offerings and enhance their overall risk assessment procedures by identifying the major drivers of credit risk. By improving knowledge of consumer profiles and behaviors, this effort not only attempts to reduce monetary losses brought on by defaults but also encourages responsible lending.

Data Acquisition and Preparation

The Credit Card Default dataset was sourced and prepared for in-depth analysis as part of the data collecting process. A thorough data cleaning procedure was part of this process, where rows and columns with missing values were either removed or addressed using imputation techniques. Numerical features were standardized to guarantee consistency in scaling, and categorical attributes were encoded to aid machine learning procedures. To preserve the integrity of the dataset, outliers were found and handled, and duplicate items were removed to avoid model bias. The dataset was structured and prepared for efficient analysis and modeling thanks to these

painstaking preprocessing procedures.

Modeling Approaches and Implementation

A variety of machine learning methods, such as Decision Trees, Random Forests, and Gradient Boosting Machines, were used to forecast credit card defaults. Because it made the decision-making process interpretable, the Decision Tree classifier was very important. By carefully adjusting variables like maximum depth and impurity criteria, we were able to ensure robust performance by optimizing the models' hyperparameters using approaches like GridSearchCV. In order to verify model performance and guarantee that the outcomes were trustworthy across various data subsets, stratified K-Fold cross-validation was employed.

Evaluation Metrics and Insights

Metrics that offer a thorough understanding of the predictive power of each machine learning model, including accuracy, precision, recall, and F1-score, were used to evaluate each model's performance. If appropriate, the silhouette score was also used to assess the clustering's quality. In terms of prediction accuracy and robustness, a comparison analysis revealed that the Random Forest model performed better than the Decision Tree and Gradient Boosting models. This research highlights how machine learning may improve credit risk assessment and give financial organizations useful information to better customer relationship management and risk management tactics. Organizations can use these findings to design educational initiatives that encourage financial responsibility and to establish focused interventions for at-risk clients.

Snapshots:

Jupyter Untitled Last Checkpoint: 29 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [19]: import pandas as pd

# Load the dataset
df = pd.read_excel("C:/Users/sarma/Downloads/default of credit card clients.xls")
df
```

Out[19]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2
0	1	20000	2	2	1	24	2	2	-1	-1	...	0	0	0	0	689
1	2	120000	2	2	2	26	-1	2	0	0	...	3272	3455	3261	0	1000
2	3	90000	2	2	2	34	0	0	0	0	...	14331	14948	15549	1518	1500
3	4	50000	2	2	1	37	0	0	0	0	...	28314	28959	29547	2000	2019
4	5	50000	1	2	1	57	-1	0	-1	0	...	20940	19146	19131	2000	36681
...
29995	29996	220000	1	3	1	39	0	0	0	0	...	88004	31237	15980	8500	20000
29996	29997	150000	1	3	2	43	-1	-1	-1	-1	...	8979	5190	0	1837	3526
29997	29998	30000	1	2	2	37	4	3	2	-1	...	20878	20582	19357	0	0
29998	29999	80000	1	3	1	41	1	-1	0	0	...	52774	11855	48944	85900	3409
29999	30000	50000	1	2	1	46	0	0	0	0	...	36535	32428	15313	2078	1800

30000 rows x 25 columns

```
In [20]: # Assuming that your dataset is loaded into a Pandas dataframe called "df"
m_rows = df.isnull().any(axis=1).sum()
```

Jupyter Untitled Last Checkpoint: 29 minutes ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel)

```
In [19]: import pandas as pd

# Load the dataset
df = pd.read_excel("C:/Users/sarma/Downloads/default of credit card clients.xls")
df
```

Out[19]:

	ID	LIMIT_BAL	SEX	EDUCATION	MARRIAGE	AGE	PAY_0	PAY_2	PAY_3	PAY_4	...	BILL_AMT4	BILL_AMT5	BILL_AMT6	PAY_AMT1	PAY_AMT2
0	1	20000	2	2	1	24	2	2	-1	-1	...	0	0	0	0	689
1	2	120000	2	2	2	26	-1	2	0	0	...	3272	3455	3261	0	1000
2	3	90000	2	2	2	34	0	0	0	0	...	14331	14948	15549	1518	1500
3	4	50000	2	2	1	37	0	0	0	0	...	28314	28959	29547	2000	2019
4	5	50000	1	2	1	57	-1	0	-1	0	...	20940	19146	19131	2000	36681
...
29995	29996	220000	1	3	1	39	0	0	0	0	...	88004	31237	15980	8500	20000
29996	29997	150000	1	3	2	43	-1	-1	-1	-1	...	8979	5190	0	1837	3526
29997	29998	30000	1	2	2	37	4	3	2	-1	...	20878	20582	19357	0	0
29998	29999	80000	1	3	1	41	1	-1	0	0	...	52774	11855	48944	85900	3409
29999	30000	50000	1	2	1	46	0	0	0	0	...	36535	32428	15313	2078	1800

30000 rows x 25 columns

```
In [20]: # Assuming that your dataset is loaded into a Pandas dataframe called "df"
m_rows = df.isnull().any(axis=1).sum()
```

```
In [21]: # Step 2: Split the data into K folds
from sklearn.model_selection import StratifiedKFold
k = 5
skfold = StratifiedKFold(n_splits=k, shuffle=True)
kfolds = list(skf.split(X=df.drop('Y', axis=1), y=df['Y']))
kfolds
```

Out[21]:

```
[(array([ 0, 1, 2, ..., 29997, 29998, 29999]),
 array([ 5, 15, 17, ..., 29988, 29998, 29999])),
 (array([ 0, 1, 2, ..., 29997, 29998, 29999]),
 array([ 4, 7, 10, ..., 29989, 29992, 29996])),
 (array([ 0, 1, 2, ..., 29996, 29997, 29998]),
 array([ 6, 8, 12, ..., 29991, 29995, 29999])),
 (array([ 0, 1, 3, ..., 29995, 29996, 29999]),
 array([ 2, 13, 16, ..., 29987, 29997, 29998])),
 (array([ 2, 4, 5, ..., 29997, 29998, 29999]),
 array([ 0, 1, 3, ..., 29963, 29977, 29994]))]
```

```

In [27]: # Step 3: Train and test the decision tree classifier with different parameter settings
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import GridSearchCV
import numpy as np

param_grid = {'criterion': ['gini', 'entropy'], 'max_depth': [10*df.shape[1]]}

giniaccuracy = np.zeros(k)
entropyaccuracy = np.zeros(k)

for i, (train_index, test_index) in enumerate(kfolds):
    X_train, X_test = df.drop('Y', axis=1).iloc[train_index], df.drop('Y', axis=1).iloc[test_index]
    y_train, y_test = df['Y'].iloc[train_index], df['Y'].iloc[test_index]

    g_search = GridSearchCV(DecisionTreeClassifier(), param_grid=param_grid, cv=4)
    g_search.fit(X_train, y_train)

    c_gini = DecisionTreeClassifier(criterion='gini', max_depth=10*df.shape[1])
    c_gini.fit(X_train, y_train)
    giniaccuracy[i] = c_gini.score(X_test, y_test)

    c_entropy = DecisionTreeClassifier(criterion='entropy', max_depth=10*df.shape[1])
    c_entropy.fit(X_train, y_train)
    entropyaccuracy[i] = c_entropy.score(X_test, y_test)
print(giniaccuracy)
print(entropyaccuracy)

[0.72483333 0.72866667 0.732      0.7225      0.71633333]
[0.73666667 0.7335      0.73066667 0.72183333 0.72733333]

```

In [15]: # Step 4: Compute the overall accuracy

```

In [15]: # Step 4: Compute the overall accuracy
ginimean_accuracy_ = giniaccuracy.mean()
entropymeans_accuracy_ = entropyaccuracy.mean()

print(f"Overall accuracy for Gini: {ginimean_accuracy_}")
print(f"Overall accuracy for Entropy: {entropymeans_accuracy_}")

Overall accuracy for Gini: 0.7247
Overall accuracy for Entropy: 0.7293000000000001

In [18]: # Step 5: Compare the results
if ginimean_accuracy_ > entropymeans_accuracy_:
    print("The DT classifier with Gini impurity gives the best results.")
else:
    print("The DT classifier with Entropy impurity measure gives the best results.")

The decision tree classifier with Entropy impurity measure gives the best results.

In [ ]:

```

Summary:

By using machine learning techniques on the Credit Card Default dataset from the UCI Machine Learning Repository, this research aims to improve credit risk assessment. Credit limits, payment histories, and demographic information are among the features included in the dataset, which covers Taiwanese credit card customers from April to September 2005. The dataset was prepared for analysis by carefully cleaning and preprocessing it, guaranteeing its structure and dependability.

The probability of credit card defaults was predicted using a number of machine learning methods, such as Decision Trees, Random Forests, and Gradient Boosting Machines. GridSearchCV was used for hyperparameter optimization, and stratified K-Fold cross-validation was used to verify model performance. According to the findings, the Random Forest model fared better than the

others in terms of robustness and prediction accuracy.

In the end, this project shows how machine learning may improve credit risk assessment and give financial institutions useful information to improve their risk management plans. The approach promotes responsible lending practices by identifying important determinants of credit risk, which enables better informed lending decisions and focused interventions for at-risk clients.

References

Halpin, Terry. *Object-Role Modeling Fundamentals: A Practical guide to data modeling with ORM*. Technics Publications, 2015

Syverson, B., & Murach, J. (2012b). *Murach's SQL Server 2012 for Developers (Training & Reference) (1st ed.)*. Mike Murach & Associates

Couchbase Product Marketing. (n.d.). *The Couchbase Blog*. *The Couchbase Blog*.

<https://blog.couchbase.com/>

SQL++ For SQL Users: A Tutorial by Don Chamberlin Copyright © September 2018 by Couchbase, Inc.

Healthcare Dataset. (2024, May 8). Kaggle.

<https://www.kaggle.com/datasets/prasad22/healthcare-dataset>

Public Dashboard Link.

<https://public.tableau.com/app/profile/sai.manoj.vissavajhula/viz/Final-DV-HC-533/Dashboard1?publish=yes>

Provost, F., & Fawcett, T. (2013). *Data Science for Business: What You Need to Know about Data Mining and Data-Analytic Thinking*. <https://dl.acm.org/citation.cfm?id=2564781>

UCI Machine Learning Repository. (n.d.).

<https://archive.ics.uci.edu/dataset/350/default+of+credit+card+clients>

Mishra, S. (2024, August 18). Unsupervised learning and data clustering - towards data science.

Medium. <https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a>

Jain, A. K. (n.d.). Cluster Analysis: basic concepts and algorithms. In *Chapter 8* (pp. 488–490).

<https://www-users.cse.umn.edu/~kumar001/dmbook/ch8.pdf>