C964: Computer Science Capstone Project

Western Governors University

Steffen Devich

Student ID: 005643564

January 11, 2025

C964: Computer Science Capstone Project

Table of Contents:

Part A: Letter of Transmittal

January 11, 2025

Mr. Tamaribuchi, President and Chairman of the Board
WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern
40701 Meieki, Nakamura-ku, Nagoya City, Aichi Prefecture 450-8711, Japan

Dear Sir,

I hope this letter finds you well and the wintertime in Nagoya is filled with seasonal delights. As you are aware, our software application consulting firm, Steffen Devich Software Consulting Solutions (SDSCS), a subsidiary of WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern, is working on a machine learning project to integrate into our current WGUSportsTracking sports betting application, YouBetWeLose for WGUApplications, Inc, in the hopes of one day including the project as an embedded customer sports betting tool. The design of the machine learning project, the NCAA Tournament Prediction Model (NTPM), will help in predicting the outcome of the annual United States NCAA Men's College Basketball tournament, better known as March Madness. In 2024, estimations show that $2.72B were bet on the outcome of this tournament (Statista Search Department, 2024). Providing such a tool to our customers will not only improve their betting experience, but it could also potentially increase our revenue generated through our sports betting application platform.

The NTPM is an innovative project designed to offer sports-related insights during one of the most widely watched sports events of the year. By using historical NCAA Men's Basketball team performance data and advanced machine learning algorithms, this model can accurately predict game outcomes, simulate tournament brackets, and identify potential NCAA Men's Basketball champions. These capabilities make the NTPM model a valuable tool for engaging sports betting fans. For WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern, the NTPM model will offer several benefits that could potentially help greatly with our company's bottom line. It will help derive customer engagement by enabling fans to interact with accurate predictions and simulated brackets, fostering greater participation in contests, promotions, and fan experiences during March Madness. Marketers can use the predictions to tailor campaigns around high-stakes matchups or compelling underdog stories, maximizing campaign impact and return on investment. Additionally, sponsorship and advertising strategies can be fine-tuned by focusing on teams or regions predicted to have the highest visibility as the tournament progresses. We can even develop premium offerings, such as

subscription-based insights or exclusive simulation results, to generate new revenue streams and provide unique value to our sports betting community.

The NTPM prediction model supports business decision-making by offering data-driven planning and scenario simulation. With such a model in place, stakeholders will be able to base strategic decisions regarding the YouBetWeLose betting platform, on robust analytics, reducing reliance on guesswork and outdated March Madness prediction strategies. The ability to simulate various tournament outcomes allows us to prepare multiple scenarios, such as targeting promotions if an unexpected underdog advances. Furthermore, insights gained from tournament engagement can uncover patterns in fan behavior and preferences, helping us develop long-term loyalty strategies. Our current sports betting customer analytics shows our customers invest heavily in March Madness, making it a prime opportunity for us to connect with our customers. The NTPM model transforms a traditionally unpredictable event into a powerful business advantage. By integrating the NTPM model into YouBetWeLose, we will be able to stay ahead of our competition by using machine learning to allow our customers to make smarter, faster, and more impactful decisions.

The project has an estimated total cost of $423,210, which covers hardware, software, personnel, and miscellaneous expenses. Hardware expenses include Dell servers, SAN disk arrays, a Cisco switch for networking, and high-performance workstations for staff, amounting to $74,300. Software costs are minimal, as open-source tools like Python and Jupyter Notebooks are used, with minor expenses for premium licenses for Tableau, Power BI, and PostgreSQL support, totaling $3,160.

Personnel costs are the largest part of the budget, with $377,600 allocated for roles such as data scientists, software engineers, reliability and DevOps engineers, database administrators, system analysts, testers, and managers. These roles are essential for data preparation, model development, system integration, testing, and project management. A completion bonus of $29,600 is included to incentivize on-time and high-quality delivery.

Finally, $5,150 is allocated for presentation tools and a contingency budget to cover other expenses that may come up as the project is developed. This budget ensures that the project can be developed and managed efficiently while supporting flexibility to address unexpected needs. By investing in these resources, we will be well-equipped to deliver the project on time and as promised, allowing you and your Executives to make inciteful and informed decisions.

SDSCS not only brings value, but we also offer a diverse range of expertise to the implementation of this project, ensuring its success at every stage. SDSCS specializes in advanced data science techniques, including machine learning model development and feature engineering, enabling the analysis and interpretation of complex datasets effectively. With a strong foundation in statistical analysis and predictive modeling, SDSCS is equipped to extract actionable insights

and deliver accurate predictions tailored to WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern business needs.

In addition to technical expertise, SDSCS excels in software development practices, such as integrating models into existing platforms, improving pipelines for scalability, and ensuring system stability through robust DevOps principles. The firm is proficient in using industry-standard tools, including Python, Jupyter Notebooks, PostgreSQL, and data visualization platforms like Tableau and Power BI, to create intuitive and impactful reporting dashboards.

Moreover, SDSCS offers comprehensive project management capabilities, demonstrated through its ability to coordinate cross-functional teams, align project objectives with business goals, and ensure prompt delivery of deliverables. The firm's consultants excel at translating technical results into business-friendly insights, bridging the gap between data science and strategic decision-making.

Finally, SDSCS emphasizes ethical considerations in data usage, ensuring compliance with legal standards and fostering trust among stakeholders. This comprehensive expertise positions SDSCS to deliver data-driven solutions that align with a company’s goals, enhance decision-making processes, and drive competitive advantage in any industry.

With the integration of the NTPM model into YouBetWeLose, we would be innovators in the age-old sports betting industry and place WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern as a leader in sports betting. We, at SDSCS, appreciate your valuable attention to this phase of potential growth in WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern, and look forward to your decision to move forward with this project.

Sincerely,

Steffen Devich, SDSCS

Part B: Project Proposal Plan

The NCAA Tournament Prediction Model (NTPM) project addresses the challenge of providing accurate, data-driven predictions for NCAA tournament outcomes, a critical feature for enhancing sports betting applications. WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern has identified the need for a robust machine learning model to improve customer engagement and decision-making accuracy within their sports betting application, YouBetWeLose. The current lack of predictive analytics in the platform limits its ability to attract and retain customers seeking competitive insights for betting on the NCAA Men's Basketball annual tournament.

WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern is positioning YouBetWeLose as a market leader in the sports betting industry. Their platform's customers range from casual sports fans to serious bettors, all of whom demand intuitive and reliable tools to inform their decisions. By incorporating NTPM into their platform, we will provide our customers with innovative predictive capabilities that use historical NCAA tournament data, advanced statistical modeling, and machine learning algorithms.

The primary deliverable of this project will be the NTPM, a machine learning-based prediction model that accurately forecasts NCAA tournament outcomes. It will include the integration of the model into the existing YouBetWeLose application, a customer-friendly interface to display predictions and insights, and comprehensive documentation for the system. Additional deliverables will include a feature importance analysis to highlight the transparency of the model's decisions and interactive data visualizations to enhance customer engagement. The project will also provide training materials and support for our IT and management teams to ensure the seamless adoption of the NTPM into their operations.

Implementing NTPM will provide numerous benefits for our customers. The machine learning model will improve the accuracy of predictions, enhancing the trust and engagement of the YouBetWeLose application customer base. By delivering insights based on years of historical data and sophisticated statistical techniques, the platform will stand out in a competitive market. Additionally, the application will empower middle management with a tool that requires limited technical expertise to use and maintain, ensuring accessibility across the organization. The enhanced customer experience and predictive reliability will lead to increased customer retention and revenue growth, justifying the project's investment.

This project will establish the technical foundation for the YouBetWeLose application's competitive edge in the sports betting industry while aligning with our goal to integrate innovative technologies that enhance customer satisfaction and engagement. With a focus on delivering actionable predictions, seamless implementation, and

long-term support, the NTPM will ensure that the YouBetWeLose platform becomes the premier choice for NCAA tournament predictions and overall sports betting.

Data Summary

The raw data for this project will be sourced from publicly available Kaggle "College Basketball Dataset" (Andrew Sundberg, 2024). This data includes team performance metrics, seeding, adjusted efficiencies, and other relevant attributes. The dataset has been carefully curated to ensure accuracy and completeness, and additional simulated data can be generated if needed to test specific edge cases. Data processing and management will be handled meticulously throughout the project's life cycle. During the design phase, the data will be cleaned, normalized, and prepared for machine learning workflows, addressing anomalies such as missing values and data outliers During development, the data will be split into training, validation, and testing subsets to ensure the model's reliability and generalizability. Maintenance will include regular updates to the dataset with new NCAA tournament data to keep the model relevant and accurate.

The chosen dataset meets the project's needs as it provides comprehensive and detailed historical records necessary for building a robust prediction model. The inclusion of advanced metrics, such as adjusted offensive and defensive efficiencies, historical tournament placements, and overall team rankings, ensures the machine learning algorithms have the depth needed to capture the intricacies of tournament dynamics. Any data anomalies will be cleaned up during preprocessing to maintain the model's accuracy and reliability.

From an ethical and legal perspective, the use of publicly available data ensures compliance with all relevant data privacy and copyright laws. As the data does not contain personally identifiable information (PII) or sensitive details, there are no significant ethical concerns. Additionally, all data processing and management steps will adhere to best practices for transparency and accountability. This ensures that the data and the resulting machine learning model align with the ethical standards expected in predictive analytics and sports-related applications.

Implementation

This project will adopt the SEMMA methodology (Sample, Explore, Modify, Model, and Assess) to ensure a structured and comprehensive approach to implementation. The SEMMA methodology is particularly suited for this project because it involves working with a complex dataset, such as NCAA basketball statistics, which requires extensive preprocessing and the development of predictive machine learning models to forecast game outcomes. By adhering to SEMMA, the project will systematically address all phases of the data science life cycle.

1. Sample:

   We will first gather datasets from our primary sources, including the Kaggle "College Basketball Dataset," which provides historical NCAA tournament data and related statistics. This dataset will be sampled to ensure it represents various game scenarios, such as upsets, predictable wins, and close contests. Additional simulated data may be generated to supplement gaps or edge cases.

2. Explore:

   In this phase, the sampled dataset will be explored to gain insights into its structure and content. Key statistics such as team seeding, adjusted efficiencies, and momentum will be analyzed to uncover patterns and relationships. Exploratory data analysis (EDA) techniques, such as visualizations and descriptive statistics, will be used to better understand how different variables interact and affect game outcomes.

3. Modify:

   The modification phase will focus on preprocessing the data and applying feature engineering. This includes cleaning the dataset by handling missing or inconsistent values, standardizing numeric features like adjusted offensive and defensive efficiencies, and eliminating redundant fields. Feature engineering will add derived metrics, such as differences in strength of schedule (SOS) and momentum, to improve the dataset's predictive capabilities. Noise will be added to overly dominant features, such as seeding differences, to ensure balanced model training.

4. Model:

   Machine learning models will be developed and trained using the preprocessed dataset. The primary algorithm for predictive modeling will be XGBoost, chosen for its high performance with tabular data. The model will be tuned using hyperparameter optimization to maximize accuracy. Predictive features such as seed differences, momentum, and adjusted efficiencies will be tested iteratively. Models will undergo rigorous evaluation to find the best-performing algorithms, which will then be refined for deployment.

5. Assess:

   The final phase will assess the accuracy and reliability of the developed models. Performance evaluation will focus on analyzing feature importance, particularly the contribution of metrics such as seed differences, strength of schedule, adjusted efficiencies, and momentum differences, to the predictions. The model's predicted outcomes will be compared to actual historical tournament results to measure prediction accuracy. Misclassified predictions will be reviewed to further refine the models. These results will be visualized using feature importance plots, accuracy distributions, and champion prediction frequency charts. All findings will be communicated to stakeholders through comprehensive visualizations and reports, providing actionable insights for decision-making.

By following the SEMMA methodology, this project will establish a structured and industry-standard approach to developing the NTPM. This ensures that every step, from data collection to model evaluation, aligns with the overarching goal of providing accurate game outcome predictions for integration into our sports betting application, YouBetWeLose.

The project implementation plan for integrating the NCAA Tournament Prediction Model (NTPM) into the YouBetWeLose platform follows a structured timeline divided into five key milestones. The first phase begins with the acceptance of the proposal, scheduled to start on February 8, 2024, and conclude on February 12, 2024. This phase includes a kickoff meeting where objectives are finalized, and resources are allocated to ensure the project is on track from its inception.

The second phase focuses on developing and presenting a technical proof of concept, which spans from February 15 to February 26, 2024. During this phase, initial datasets will be gathered, and preliminary data analysis will be conducted. Key findings and insights will be shared with stakeholders to demonstrate the feasibility of the project and gather feedback.

The third milestone, scheduled from February 29 to March 23, 2024, involves the development and testing of the predictive model. This critical phase will include training and refining the machine learning algorithms and evaluating them against defined performance metrics. The results will be submitted for review to ensure the model meets the project's goals and stakeholder expectations.

The fourth phase, taking place from March 28 to April 8, 2024, focuses on preparing the deliverables. This phase will involve finalizing the predictive model and creating a proof-of-concept API or script for integration into the YouBetWeLose platform. All technical documentation and implementation guides will also be completed during this period.

Finally, the deliverables will be submitted between April 11 and April 15, 2024. This phase involves presenting all project deliverables, including the finalized model, detailed reports, and stakeholder presentations. This structured timeline ensures the project progresses efficiently, meeting all key milestones and delivering a robust machine learning solution.

Timeline

The projected timeline of the project, to include start and end dates, duration of each step, and a description of each milestone is provided below.

The proposal is accepted. The project begins with a kickoff meeting to finalize objectives and allocate resources.	4 Days	8-Feb-24	12-Feb-24
A technical proof of concept is presented. Initial datasets are gathered, and preliminary analysis is shared with stakeholders.	11 Days	15-Feb-24	26-Feb-24
Submitted for review. The predictive model is developed, tested, and submitted for evaluation against defined performance metrics.	23 Days	29-Feb-24	23-Mar-24
Deliverables are prepared. The model is finalized, and a proof-of-concept API/script for platform integration is completed.	11 Days	28-Mar-24	8-Apr-24
Deliverables are submitted. All project deliverables, including the final report and presentation, are provided to stakeholders.	4 Days	11-Apr-24	15-Apr-24

Evaluation Plan

Stage	Verification Method
Proposal Acceptance	Confirm alignment of objectives with client expectations through documented meeting minutes. Verify allocation of resources and responsibilities by reviewing the project plan and resource assignment matrix.
Proof of Concept Presentation	Evaluate the completeness, accuracy, and relevance of initial datasets through exploratory data analysis. Confirm the results of preliminary findings by cross-referencing with client expectations and project objectives. Document stakeholder feedback for necessary adjustments.
Submission for Review	Test the predictive model against performance metrics such as accuracy, precision, and recall. Verify consistency across multiple simulations by reviewing results. Address any identified issues through iterative improvements and stakeholder feedback.
Deliverable Preparation	Verify that the finalized model, proof-of-concept API/script, and documentation are complete and functional. Ensure readiness for integration by conducting system compatibility tests. Review deliverables with stakeholders and incorporate their feedback, as necessary.
Deliverables Submission	Validate all final deliverables, including the report, presentation, and technical components, against project objectives. Confirm that they align with stakeholder expectations and meet the agreed-upon quality standards.

Upon completion of the project, the validation methods will focus on assessing the model's performance, its integration into the YouBetWeLose platform, and its overall business value. First, the model predictions will be validated against historical NCAA Tournament outcomes to ensure predictive accuracy. Stakeholder validation will be conducted by presenting the results and securing approval for the model's integration into the platform. Additionally, the business value of the model will be evaluated by comparing its predictions with existing company benchmarks and other internal tools. Finally, the usability of the API or script will be validated through customer acceptance testing, ensuring that IT teams and other stakeholders can seamlessly integrate the model into the platform. This comprehensive validation process will confirm the model's effectiveness and its alignment with our companies needs and expectations.

Resources and Costs

Resource	Description	Units	Cost
Hardware			
Dell Servers	3 robust Dell servers for on-premises development and deployment.	3 units @ $4,000 each	$12,000.00
VMware License	License for virtualization to manage multiple environments.	1 license	$2,500.00
Red Hat Linux Licenses	Licenses for Red Hat Linux OS.	3 units @ $600 each	$1,800.00
Cisco Switch	1 Cisco switch with fiber connections to support high-speed networking and interconnectivity.	1 unit @ $8,000	$8,000.00
SAN Disk Arrays	Storage area network (SAN) disk arrays for each Dell server.	3 units @ $5,000 each	$15,000.00
Cloud Backup Storage	Off-site storage (cloud-based) for backups and disaster recovery operations.	1 subscription	$3,000.00
Robust Workstations	High-performance workstations for FTEs and 2 spare units, equipped with i7 processors, 32GB RAM, and SSDs.	16 units @ $2,187.50 each	$35,000.00
Software			
Python Environment	Python libraries such as pybaseball, scikit-learn, pandas, matplotlib.	Open source	$0.00
Jupyter Notebooks	Interactive environment for development and documentation.	Open source	$0.00
IDE	Software like PyCharm, VS Code, or Jupyter Lab for coding and debugging.	Free or 1 premium license	$200.00
PostgreSQL Support	Commercial support for the PostgreSQL database used in the project.	1 license	$1,000.00
Power BI Licenses	Licenses for data visualization and reporting using Microsoft Power BI.	2 licenses @ $140 each annually	$280.00
Tableau Licenses	Licenses for data visualization and reporting using Tableau.	2 licenses @ $840 each annually	$1,680.00

Personnel/Work Hours			
Data Scientists	Data preparation, model development, and evaluation at $75/hour.	2 @ 400 hours each	$60,000.00
Software Engineers	Integration of the model into the WGUSportsTracking platform at $60/hour.	2 @ 400 hours each	$48,000.00
Reliability Engineer	Ensures system stability and performance at $60/hour.	1 @ 400 hours	$24,000.00
Database Administrator	Manages database operations and ensures data integrity at $80/hour.	1 @ 400 hours	$32,000.00
DevOps Engineer	Manages the on-premises development environment and deployment pipeline at $50/hour.	1 @ 400 hours	$20,000.00
System Analyst	Requirements gathering and alignment with project goals at $50/hour.	1 @ 400 hours	$20,000.00
Testers	Validates and tests the system to ensure quality at $40/hour.	2 @ 400 hours each	$32,000.00
Project Manager	Coordinates the project and communicates with stakeholders at $100/hour.	1 @ 400 hours	$40,000.00
Product Manager	Manages the overall vision and ensures product-market fit at $80/hour.	1 @ 400 hours	$32,000.00
Completion Bonuses	Bonuses for all personnel if the project is delivered on-time and as proposed (10% of total wages).	Lump Sum	$29,600.00
Miscellaneous			
Presentation Tools	Software for preparing final reports and stakeholder presentations (e.g., PowerPoint).	Free or 1 premium license	$150.00
Contingency Budget	Reserved for unforeseen costs, such as additional hardware, licenses, or personnel hours.	Lump Sum	$5,000.00
		Total	**$423,210.00**

Part C: Application

Application Files

\dataSets

\cbbAll.csv

\debugging

\simulation_debug.log

\requirements.txt

\ Main.ipynb

Part D: Post-implementation Report

Solution Summary

The project aimed to address the challenge of providing accurate, data-driven predictions for the NCAA Men's Basketball Tournament. The problem was that the existing sports betting application, YouBetWeLose, lacked advanced predictive analytics, limiting its ability to engage customers seeking reliable tools for betting decisions. Without such capabilities, the platform faced difficulties in attracting and retaining both casual sports fans and serious bettors.

To solve this, the project developed the NTPM, a machine learning-based system designed to forecast tournament outcomes with high accuracy. The solution used historical NCAA data, advanced statistical modeling, and algorithms like XGBoost to predict game results, simulate tournament brackets, and identify potential champions. The model was integrated into YouBetWeLose through a customer-friendly interface and supported by interactive visualizations, enhancing customer engagement and decision-making. The project also provided technical documentation and training to ensure seamless adoption and long-term usability.

By transforming a traditionally unpredictable event into a data-driven opportunity, the NTPM elevated YouBetWeLose as a competitive player in the sports betting market. It empowered customers with actionable insights while offering WGU Business Enterprise and Tamaribuchi Heavy Manufacturing Concern a strategic advantage in increasing customer retention, revenue, and market presence.

The application provided a solution to the problem by integrating the NTPM into the YouBetWeLose sports betting platform, addressing the lack of advanced predictive analytics. The NTPM utilized historical NCAA tournament data and machine learning algorithms, such as XGBoost, to generate accurate predictions for game outcomes, simulate

tournament brackets, and identify likely champions. This offered customers reliable insights for their betting decisions during March Madness.

Through a customer-friendly interface, the application presented predictions in an accessible format, supported by interactive data visualizations that engaged both casual fans and serious bettors. Features such as feature importance analysis enhanced transparency, allowing customers to understand the factors driving the predictions. Additionally, the application's seamless integration into the existing platform ensured minimal disruption while introducing advanced functionalities.

The application also empowered company management by providing tools that required little technical expertise to use and maintain, ensuring accessibility across the organization. This enhanced customer experience, combined with the predictive reliability of the NTPM, increased customer trust and engagement. By addressing the limitations in predictive analytics, the application established YouBetWeLose as a competitive leader in the sports betting industry and drove significant business value through increased customer retention and revenue growth.

Data Summary

The raw data for the project was sourced from the publicly available College Basketball Dataset on Kaggle, curated by Andrew Sundberg. This dataset included detailed historical records of NCAA Men's Basketball tournament performance metrics, such as team seeding, adjusted offensive and defensive efficiencies, and strength of schedule. The data was collected from publicly accessible sources and compiled to provide a comprehensive view of historical tournament outcomes.

Throughout the development of the NTPM, data processing and management were meticulously handled across the design, development, and maintenance phases. In the design phase, the raw data was sourced from the publicly available Kaggle College Basketball Dataset, which provided historical tournament metrics such as team seeding, adjusted efficiencies, and postseason results. Initial processing included mapping categorical postseason stages to numerical values for easier model interpretation:

```
postseason_mapping = {"R64": 0, "R32": 1, "S16": 2, "E8": 3, "F4": 4, "2ND": 5, "Champions": 6}
data['POSTSEASON'] = data['POSTSEASON'].map(postseason_mapping)
```

Feature engineering was also a critical part of this phase. Metrics like historical success (HISTORICAL_SUCCESS) and strength of schedule (SOS) were calculated to enhance predictive capabilities. For instance, the average postseason performance of teams was computed using:

```
data['HISTORICAL_SUCCESS'] = data.groupby('TEAM')['POSTSEASON'].transform('mean')
```

Additionally, SOS metrics were derived by comparing team efficiencies against conference averages:

```
def calculate_strength_of_schedule(data):
    conf_means = data.groupby(['CONF'])[['ADJOE', 'ADJDE']].mean()
    data = data.merge(conf_means, on='CONF', suffixes=('', '_CONF_AVG'))
    data['SOS_ADJOE'] = data['ADJOE'] - data['ADJOE_CONF_AVG']
    data['SOS_ADJDE'] = data['ADJDE'] - data['ADJDE_CONF_AVG']
    return data
```

During the development phase, extensive preprocessing ensured the dataset's reliability. Duplicate teams within the same year were identified and removed to prevent biased outcomes, and self-matchups (where a team was paired against itself) were excluded:

```
if year_data.duplicated(subset=['TEAM']).any():
    logging.warning(f"Duplicate teams found in year {year}. Removing duplicates.")
    year_data = year_data.drop_duplicates(subset=['TEAM'])
```

Pairwise matchups were generated for tournament teams, calculating features such as seed differences, momentum, and SOS differences for each matchup:

```
matchups.append({
    'Year': year,
    'Team1': team1.TEAM,
    'Team2': team2.TEAM,
    'Seed_diff': team1.SEED - team2.SEED,
    'SOS_ADJOE_diff': team1.SOS_ADJOE - team2.SOS_ADJOE,
    'HISTORICAL_SUCCESS_diff': 4 * (team1.HISTORICAL_SUCCESS - team2.HISTORICAL_SUCCESS),
    'Winner': 1 if team1.POSTSEASON > team2.POSTSEASON else 0
})
```

To prepare data for training, features were standardized using StandardScaler, ensuring consistency in scale across all inputs:

```
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

The XGBoost classifier was selected for its superior performance with tabular data, and class weights were computed to handle imbalanced classes:

```
class_weights = compute_class_weight('balanced', classes=np.unique(y_train), y=y_train)
xgb_model = XGBClassifier(
    n_estimators=200, learning_rate=0.1, max_depth=5, colsample_bytree=0.8,
    subsample=0.8, random_state=42, scale_pos_weight=class_weights[0])
xgb_model.fit(X_train_scaled, y_train)
```

During the maintenance phase, logging and debugging systems were implemented to capture issues during simulations. Logs were cleared at the start of each run to ensure clean debugging sessions:

```
log_file = "debugging/simulation_debug.log"
if os.path.exists(log_file):
```

```
    os.remove(log_file)
logging.basicConfig(
    filename=log_file, level=logging.INFO,
    format="%(asctime)s - %(levelname)s - %(message)s"
)
```

Simulations were conducted to predict tournament outcomes, with steps to eliminate self-matchups and ensure unique winners at each round:

```
for idx, game in current_round.iterrows():
    if team1 == team2:  # Skip self-matchups
        continue
```

The final script visualized feature importance and champion frequency, providing stakeholders with actionable insights into the model's behavior:

```
importances = xgb_model.feature_importances_
plt.barh(sorted_features, sorted_importances)
```

Key troubleshooting steps included addressing duplicate teams, avoiding self-matchups, and managing imbalanced classes. These systematic processes ensured the data's integrity, enabling the NTPM to deliver accurate, data-driven predictions for March Madness.

Machine Learning

1. Supervised Learning

Supervised learning, the foundation of this project, involves training a model on labeled data where the outcomes are known. In this case, historical NCAA tournament data served as the labeled dataset, with the target variable indicating the winner of each matchup. This method was developed by mapping categorical postseason outcomes (e.g., "R64", "Champions") to numerical labels, enabling the model to predict winners based on input features.

The choice of supervised learning was justified as the problem required mapping input features like seed differences and adjusted efficiencies to a binary outcome (win/loss). This approach allowed the model to learn from historical patterns and make accurate predictions. The transformation of categorical outcomes to numerical values was performed as follows:

```
postseason_mapping = {"R64": 0, "R32": 1, "S16": 2, "E8": 3, "F4": 4, "2ND": 5, "Champions": 6}
data['POSTSEASON'] = data['POSTSEASON'].map(postseason_mapping)
```

2. Gradient Boosting with XGBoost

XGBoost, a gradient boosting algorithm, was used as the core predictive model. This method builds an ensemble of decision trees iteratively, optimizing for reduced prediction error at each step. The model was developed by training it on input features engineered from historical data, such as seed differences, strength of schedule (SOS), and momentum.

XGBoost was chosen for its high performance with tabular data, ability to handle feature interactions, and efficiency in terms of computation. Hyperparameter tuning ensured the model's robustness and adaptability:

```
class_weights = compute_class_weight('balanced', classes=np.unique(y_train), y=y_train)
xgb_model = XGBClassifier(
    n_estimators=200, learning_rate=0.1, max_depth=5, colsample_bytree=0.8,
    subsample=0.8, random_state=42, scale_pos_weight=class_weights[0])
xgb_model.fit(X_train_scaled, y_train)
```

The justification for using XGBoost lies in its proven ability to perform well with imbalanced datasets, as demonstrated here by incorporating class weights to handle discrepancies in win/loss frequencies.

3. Feature Engineering

Feature engineering involved creating new variables to enhance the model's predictive power. For instance, historical success was computed as the average postseason performance of a team, and strength of schedule was calculated as the difference in adjusted efficiencies relative to conference averages. Momentum was derived as a team's win ratio during the season.

Feature engineering was justified because raw data lacked variables that could directly model team performance in matchups. By incorporating domain knowledge, the engineered features provided meaningful insights to the machine learning model:

```
data['HISTORICAL_SUCCESS'] = data.groupby('TEAM')['POSTSEASON'].transform('mean')

def calculate_strength_of_schedule(data):
    conf_means = data.groupby(['CONF'])[['ADJOE', 'ADJDE']].mean()
    data = data.merge(conf_means, on='CONF', suffixes=('', '_CONF_AVG'))
    data['SOS_ADJOE'] = data['ADJOE'] - data['ADJOE_CONF_AVG']
    data['SOS_ADJDE'] = data['ADJDE'] - data['ADJDE_CONF_AVG']
    return data
```

Validation

1. Validation for Supervised Learning

To validate the supervised learning method, calculate classification metrics like accuracy, precision, recall, and F1-score using the test dataset.

Code:

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, confusion_matrix

# Predict on test set
y_pred = xgb_model.predict(X_test_scaled)

# Calculate validation metrics
accuracy = accuracy_score(y_test, y_pred)
precision = precision_score(y_test, y_pred)
recall = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

# Confusion matrix
conf_matrix = confusion_matrix(y_test, y_pred)

# Display results
print(f"Accuracy: {accuracy:.2f}")
print(f"Precision: {precision:.2f}")
print(f"Recall: {recall:.2f}")
print(f"F1 Score: {f1:.2f}")
print("Confusion Matrix:")
print(conf_matrix)

#Results after running ten simulations
Accuracy: 0.95
Precision: 0.99
Recall: 0.94
F1 Score: 0.97
Confusion Matrix:
[[ 641   10]
 [  82 1283]]
```

2. Validation for Gradient Boosting with XGBoost

To validate the gradient boosting model, use cross-validation with stratified folds to ensure balanced classes across splits.

Code:

```
from sklearn.model_selection import StratifiedKFold, cross_val_score

# Define cross-validation strategy
cv = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)

# Perform cross-validation
cv_scores = cross_val_score(xgb_model, X_train_scaled, y_train, cv=cv, scoring='accuracy')

# Display results
print(f"Cross-Validation Scores: {cv_scores}")
print(f"Mean CV Accuracy: {cv_scores.mean():.2f}")

#Results after running ten simulations
Cross-Validation Scores: [0.96552579 0.96329365 0.96329365 0.9625496  0.96155754]
Mean CV Accuracy: 0.96
```

3. Validation for Feature Engineering

To validate feature engineering, compute the correlation coefficients or use permutation importance to check if engineered features improve model performance.

Code:

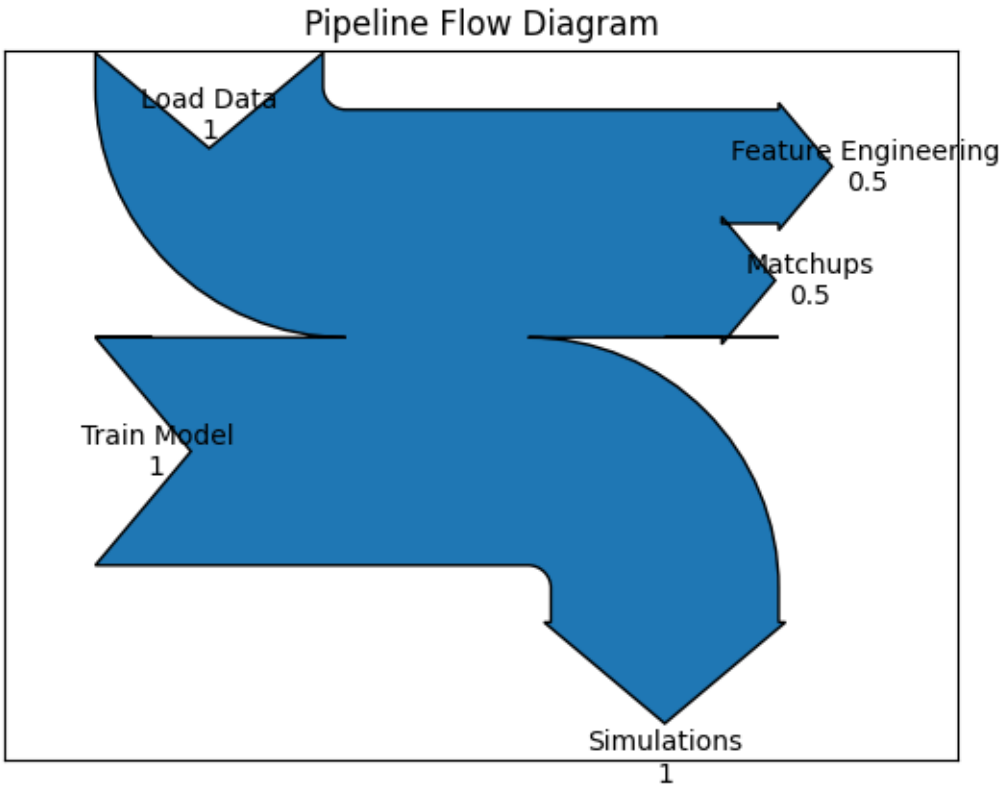
```
from sklearn.inspection import permutation_importance

# Compute permutation importance
perm_importance = permutation_importance(xgb_model, X_test_scaled, y_test, n_repeats=10,
random_state=42)

# Display results
for i in perm_importance.importances_mean.argsort()[::-1]:
    print(f"Feature: {model_features[i]}, Importance: {perm_importance.importances_mean[i]:.4f}")

#Results after running ten simulations
Feature: Rank_diff, Importance: 0.3641
Feature: Seed_diff, Importance: 0.1258
Feature: BARTHAG_diff, Importance: 0.0822
Feature: SOS_ADJDE_diff, Importance: 0.0159
Feature: HISTORICAL_SUCCESS_diff, Importance: 0.0098
Feature: MOMENTUM_diff, Importance: 0.0065
Feature: SOS_ADJOE_diff, Importance: 0.0055
```

These validation methods will ensure each part of the project is reliable, effective, and aligned with the objectives.

Visualizations

The Sankey diagram was created early in the script during the pipeline visualization stage. This diagram visualized the flow of the data processing pipeline, from loading data to feature engineering, matchup generation, model training, and simulations. It was used to offer a high-level view of the project's workflow.

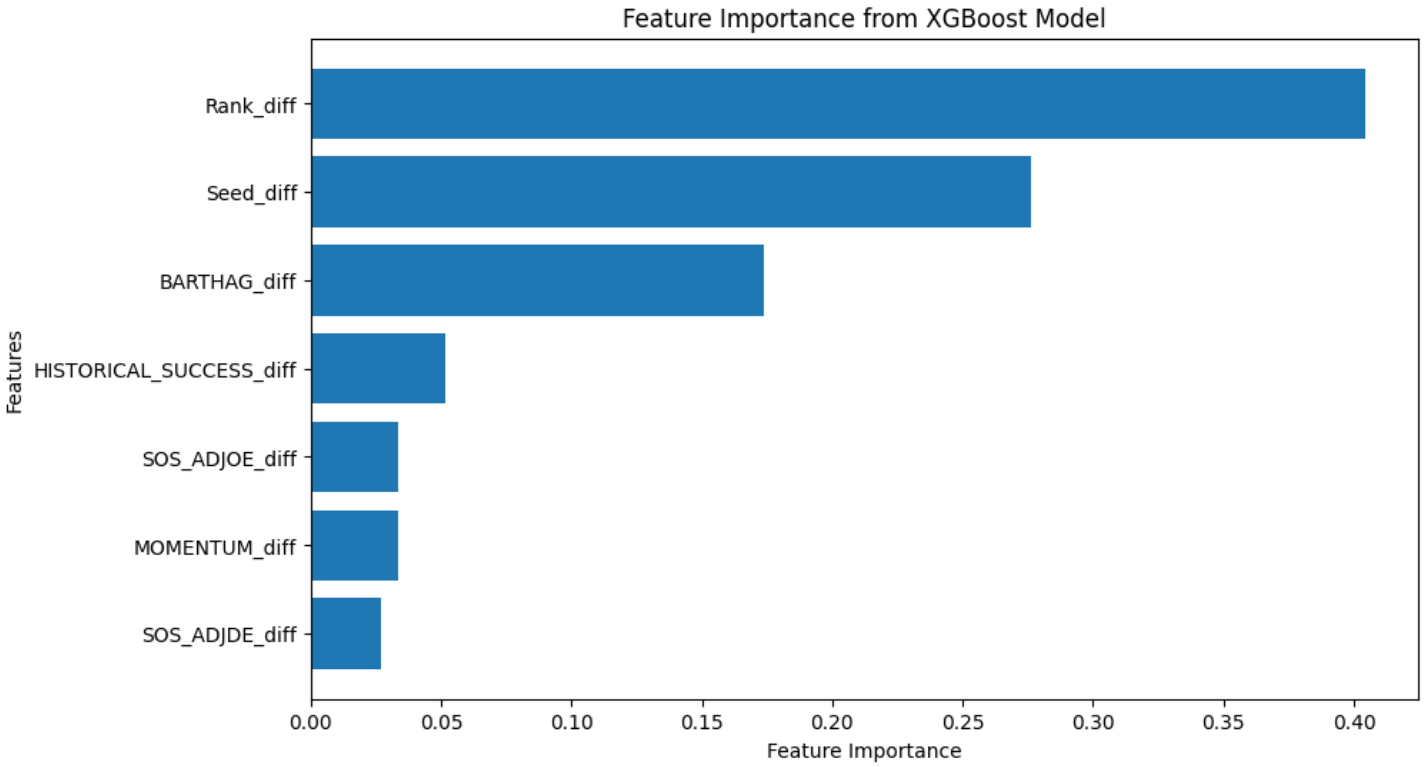
```
Sankey(flows=[1, -0.5, -0.5, 1, -1],
    labels=['Load Data', 'Feature Engineering', 'Matchups', 'Train Model', 'Simulations'],
    orientations=[1, 0, 0, 0, -1]).finish()
plt.title('Pipeline Flow Diagram')
plt.show()
```



The feature importance chart was generated after training the XGBoost model to validate the relevance of input features, such as HISTORICAL_SUCCESS_diff and Rank_diff. Both were created to add weight to each teams yearly schedule outcome and tournament showings. It showcases the relative importance of each feature in the XGGoost model. It showed which features had the greatest impact on predictions. This visualization was important for interpreting the model and adjusting any feature based on expected outcomes.

```
importances = xgb_model.feature_importances_
feature_names = model_features
```

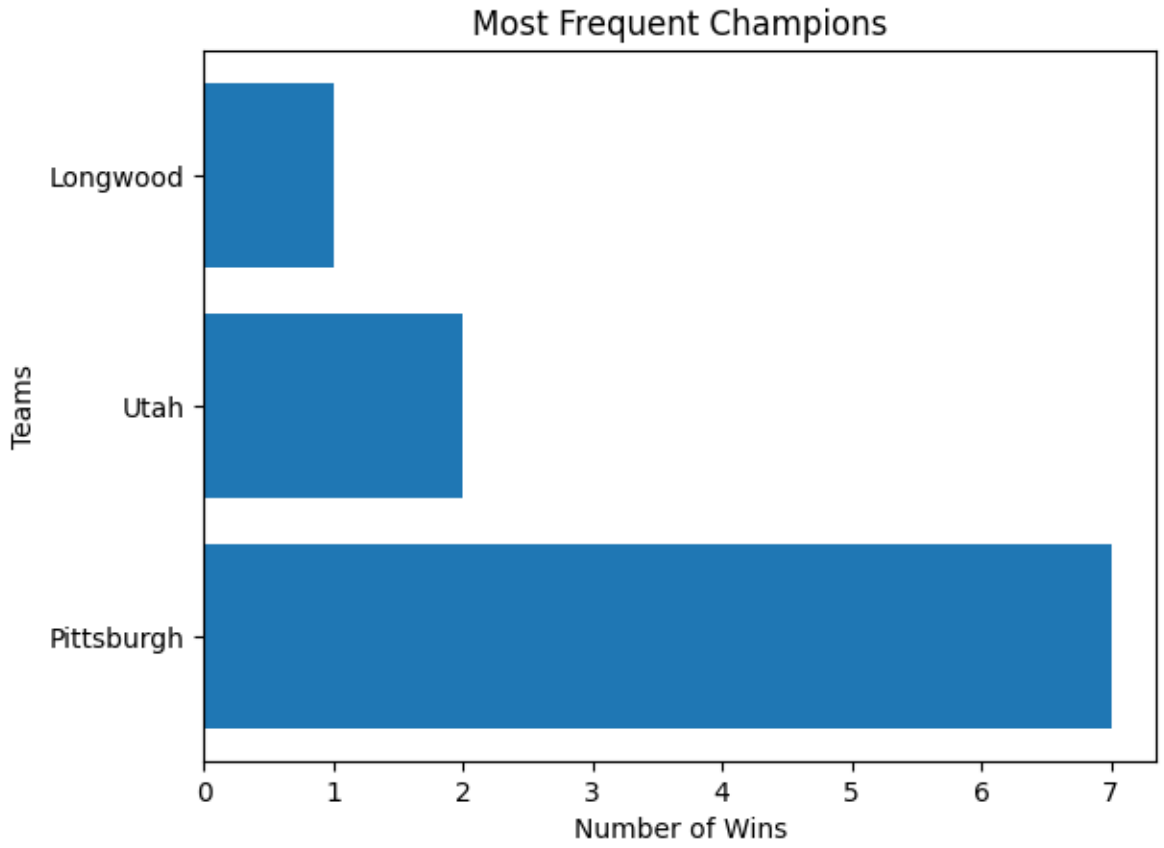
```
sorted_idx = np.argsort(importances)[::-1]
sorted_features = [feature_names[i] for i in sorted_idx]

plt.figure(figsize=(10, 6))
plt.barh(sorted_features, sorted_importances, align='center')
plt.xlabel("Feature Importance")
plt.ylabel("Features")
plt.title("Feature Importance from XGBoost Model")
plt.gca().invert_yaxis()
plt.show(
```



The champion frequency bar chart was produced during the simulation analysis stage, after running the predetermined number of tournament simulations. This bar chart summarized the frequency of predicted champions across all the simulations. It highlighted the most likely tournament winners, which provided stakeholders with an idea of possible tournament outcomes.

```
champions = [simulate_bracket(filtered_matchups, xgb_model) for _ in range(num_simulations)]
champion_counts = Counter(champions)

teams, counts = zip(*champion_counts.most_common(num_top_champions))
plt.barh(teams, counts)
plt.title(f"Most Frequent Champions")
```

```
plt.xlabel("Number of Wins")
plt.ylabel("Teams")
plt.show()
```



User Guide

This project is run on a Jupyter notebook from an Anaconda distribution. This provides an easy way to manage and run python scripts.

To manage software version compatibility, we recommend using the software listed below.

1. This application requires three files to function as developed:
    a. Main.ipynb – This is the Jupyter Notebook file that holds the source code.
    b. requirements.txt – Contains a list of all the required Python packages needed to run the application.
    c. cbbAll.csv – Contains the dataset.
2. The Anaconda distribution we used during development was Anaconda3-2022.10-Windows-x86_64.exe. To download this distribution, go to https://repo.anaconda.com/archive/. This will work for Windows 10/11 and

Windows Server 2019-2022. For other Operating Systems, pick an installer from the same date. This distribution comes with Python version 3.9, which will work with the other required Python packages used in the project.

3. Once you have downloaded the installer, double click the file to start the installation.

4. Accept any administrator requirements, as needed.

5. Follow the installer prompts to finish the installation.

6. The following is the required folder structure and application file placement. It is recommended that the root folder for the project be a customer folder that has the required permissions to run an Anaconda Python project.

   During development, in a Windows environment, c:\customers\<customer name>\Documents, worked well.

Application Files

\dataSets

\cbbAll.csv

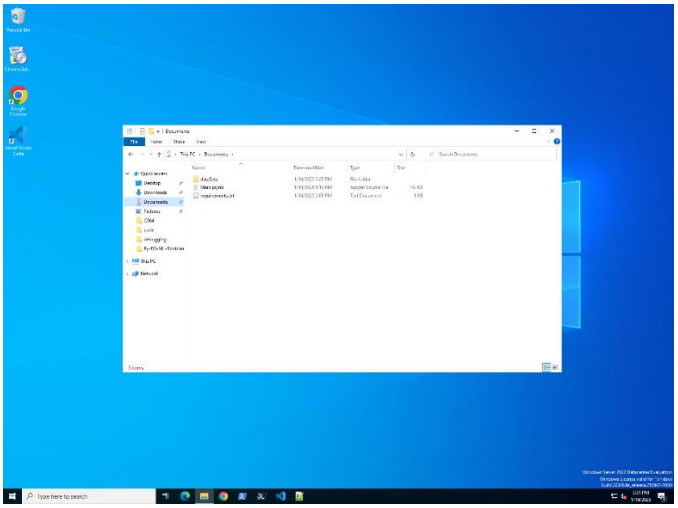
\debugging

\simulation_debug.log

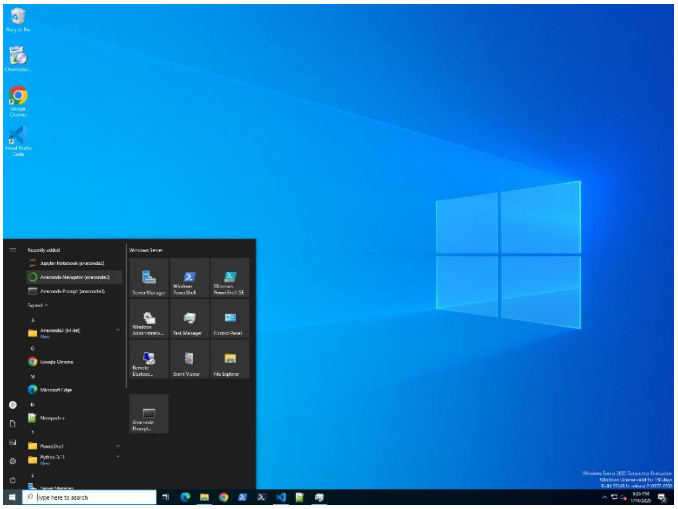
\requirements.txt

\ Main.ipynb

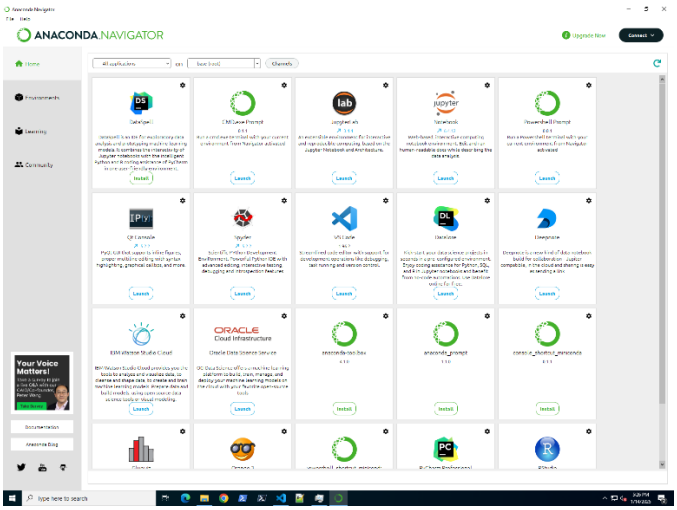
7. Place the project files.



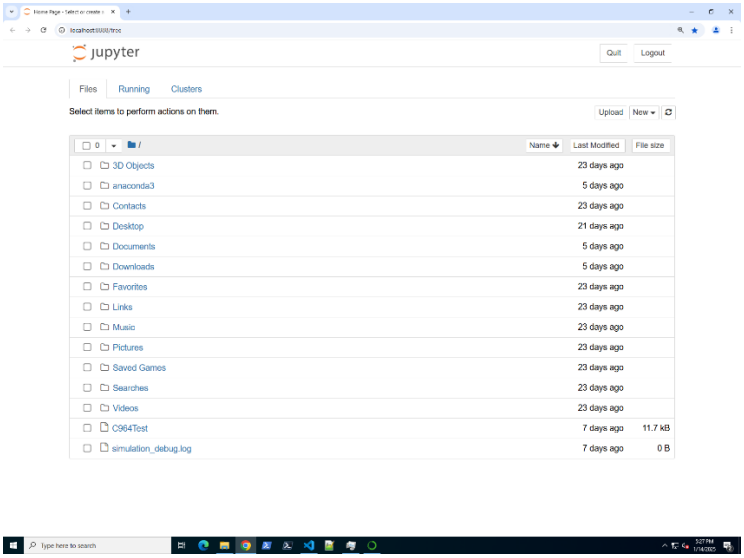
8. Run the Anaconda Navigator application.



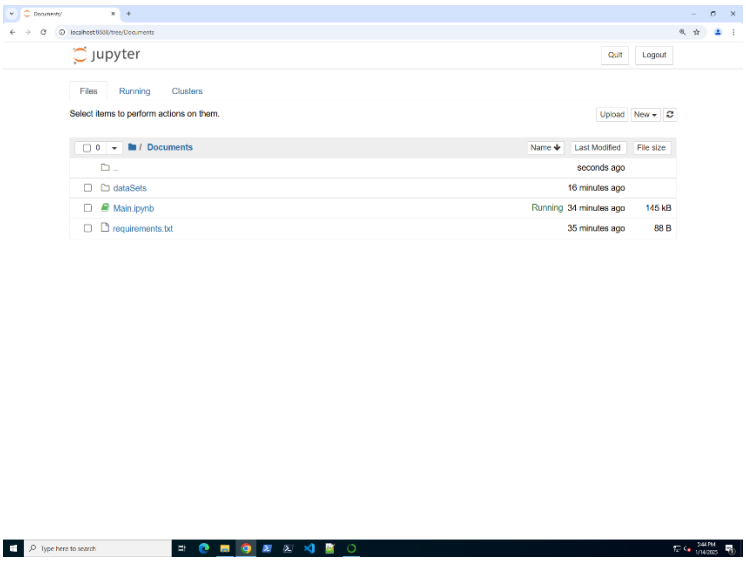
9. Find the Jupyter Notebook application and click Launch.



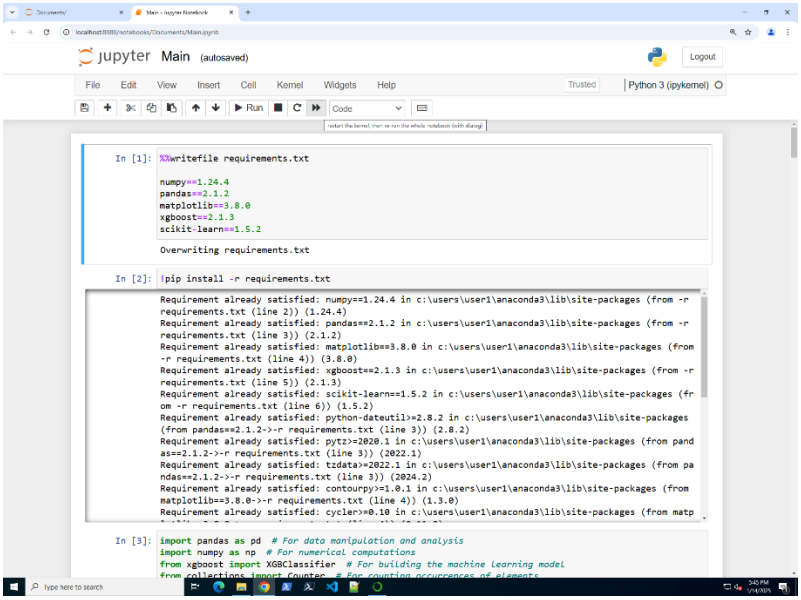
10. Once open, on the left side of the file list click on Documents.



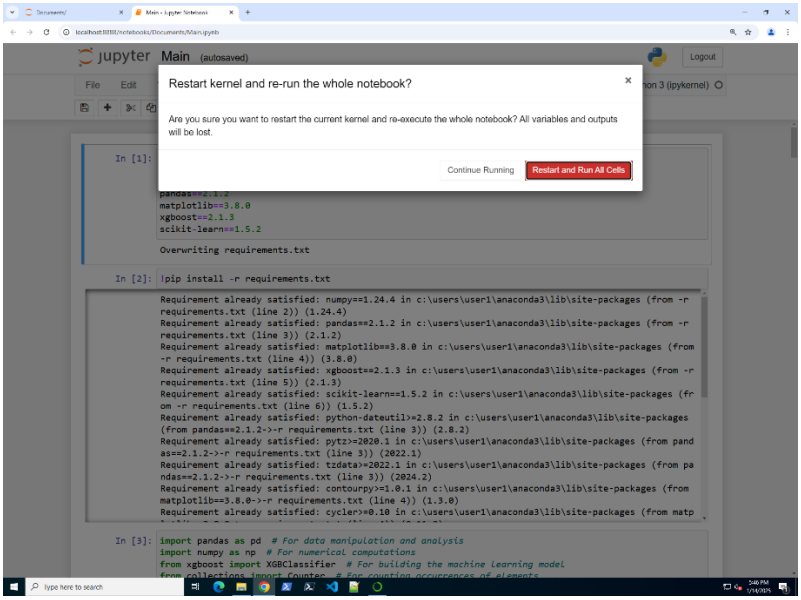
11. Find the Main.ipynb file and click it to open the project.



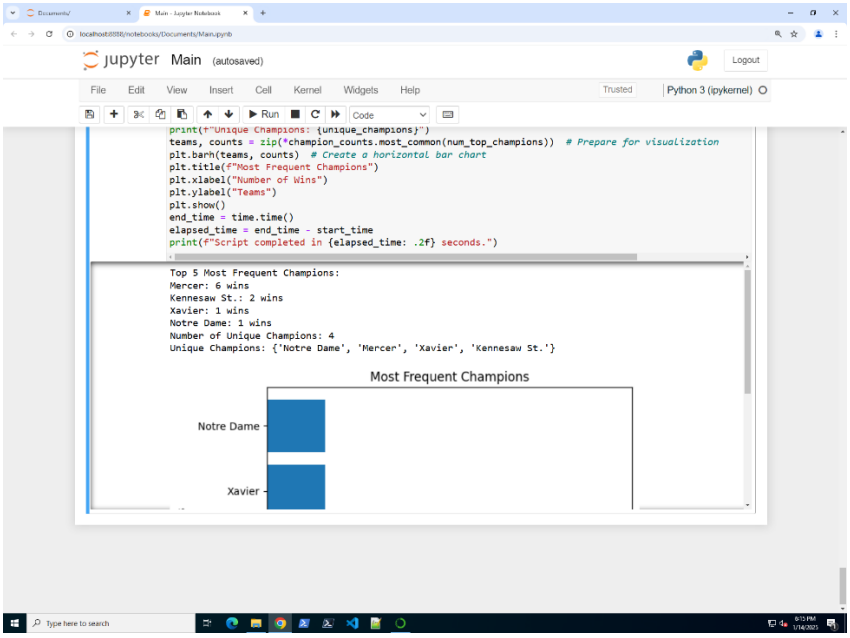
12. The file will open with a list of cells showing python code.
13. To run the project, we recommend clicking the double-arrows from the middle menu. This will restart the Python kernel associated with the notebook and re-run the entire notebook.



14. In the information window that pops up, choose Restart and Run All Cells.



15. The notebook will begin to run the project, moving down the cells as each section is successfully run and completed.
16. As the project finishes, cell number 16 will show the following output as the simulated tournament winner(s):



17. To run the simulations again, click the double arrows from the middle menu.

Reference Page

Statista Search Department (2024, July 19) *Estimated amount of money bet on March Madness in the United States in 2019 and 2024* [Infographic]. Statista. https://www.statista.com/statistics/1296462/total-amount-bet-march-madness/

Andrew Sundberg. (March 2024). College Basketball Dataset, Version 1. Retrieved January 11, 2025, from https://www.kaggle.com/datasets/andrewsundberg/college-basketball-dataset/.