ISO 9001:2015

COLLEGE *of* COMPUTER STUDIES

Final Project and Final Exam
CSEC 413-MODELING AND SIMULATION

Modeling and Simulation with Python

Project Goal

This project will explore the concepts of modeling and simulation using Python. The goal is to gain hands-on experience with Python libraries and tools (such as NumPy, Pandas, Scikit-learn, Matplotlib, etc.) commonly used for modeling and simulation tasks.

Project Options

A group must **choose one (1)** of the following two options for their project.

Option 1: Synthetic Data Modeling (General Application)

This project involves creating a synthetic dataset and applying modeling and simulation techniques to it.

- **Focus:** Exploratory Data Analysis (EDA), feature engineering, and applying standard modeling techniques (e.g., regression, classification, time series).
- **Task:**
  1. Generate a synthetic dataset with known properties (e.g., specific distributions, correlations) using libraries like `numpy` and `scikit-learn`.
  2. Perform EDA to "discover" the characteristics of your data.
  3. Apply a suitable modeling technique (`scikit-learn`, `statsmodels`, `tensorflow`) to the data.
  4. Use the fitted model to generate simulated outcomes or predictions.
  5. Evaluate the model's performance by comparing simulated outcomes against the original data's known properties.
- **Example:** The sample app provided (`https://projectsyntheticdatageneration.streamlit.app/`) is a good example of this option.

Option 2: Stochastic Game Simulation (Perya or Casino Game)

This project focuses on modeling and simulating a system based on probability, specifically a simple casino or Filipino "perya" game (e.g., Color Game, Simple Dice Roll, basic Slot Machine).

- **Focus:** Modeling probabilistic systems, Monte Carlo simulation, and analyzing the impact of "tweaking" probabilities to create a house edge.
- **Task:**
  1. **Model a "Fair Game":** Define the rules, probabilities, and payout structure for a game where the player and house have (roughly) equal odds.
  2. **Model a "Tweaked Game":** Create a second version of the model where you introduce a "house edge" by subtly changing the rules, probabilities, or payouts.
  3. **Simulate:** Run both models thousands of times (e.g., 10,000+ plays) using Monte Carlo simulation to generate outcome data (e.g., player win/loss, house profit).
  4. **Perform EDA:** Analyze the *simulated outcomes* from both models.
  5. **Compare & Evaluate:** Compare the results from the "fair" vs. "tweaked" simulations. Quantify the house edge and analyze the impact of your "tweaks."
- **Example:** The sample app provided (https://stochasticgamesimulation.pythonanywhere.com/) is a good example of this option.

Suggested Techniques for Option 2 (Tweaking)

If you choose Option 2, you must implement a "tweak." Here are some suggestions:

- **Weighted Probabilities:** Instead of a fair die (`numpy.random.randint(1, 7)`), simulate a weighted one. Use `numpy.random.choice()` with the `p` parameter to assign different probabilities to each outcome (e.g., making the "house" number appear 20% of the time instead of 16.67%).
- **Modifying Payouts:** Keep the game probabilities fair but change the payout structure. For example, a win that should pay 2-to-1 is modified to pay 1.9-to-1, creating a small, consistent edge for the house.
- **Adjusting Mean/Standard Deviation:** If your game has a variable payout (like a slot machine), you can model the payout as a normal distribution. The "tweak" could be setting the *mean* of the payout distribution to be slightly negative (e.g., -0.05 per play) while using a **standard deviation** to ensure variance (occasional big wins) but an overall loss for the player over time.

Project Documentation and Steps

Your final `Documentation.pdf` and video presentation must follow this structure, adapted for the option you chose.

1. **Introduction:**
    - Provide an overview of your project.
    - State which option you chose and why.
    - Explain the goal of your specific project (e.g., "to model a synthetic customer dataset..." or "to simulate the 'Color Game' and analyze the effect of weighted dice...").
2. **Project Overview:**
    - Explain the different steps you took to complete the project, from generation/modeling to analysis.
3. **Data Generation / Model Definition:**
    - **(Option 1):** Describe the synthetic data you generated. What are its features? What known properties (distributions, correlations) did you build into it? Show code snippets.
    - **(Option 2):** Define the rules, probabilities, and payout structures for *both* your "fair" and "tweaked" game models. Justify your design choices.
4. **Exploratory Data Analysis (EDA):**
    - **(Option 1):** Perform EDA on your generated static dataset. Use visualizations (`matplotlib`, `seaborn`) to explore statistical properties, correlations, and patterns.
    - **(Option 2):** Run your simulations (10,000+ plays) and perform EDA on the *simulated outcomes*. Visualize the results (e.g., player profit over time, distribution of wins/losses, final house profit).
5. **Modeling and Simulation:**
    - **(Option 1):** Describe the modeling technique you chose (e.g., Linear Regression, K-Means Clustering, ARIMA). Explain why it is appropriate. Show the process of fitting the model.
    - **(Option 2):** This section should detail the *simulation* itself. Show how you ran the 10,000+ trials. Most importantly, detail the *implementation* of your "tweak." Show the code that differentiates the "fair" model from the "tweaked" model.
6. **Evaluation and Analysis:**
    - **(Option 1):** Evaluate the performance of your model using appropriate metrics (e.g., Accuracy, R-squared, MSE). Compare the simulated outcomes with the original data's known properties.
    - **(Option 2):** This is the most critical part. **Compare the simulation results** from the "fair" vs. "tweaked" models. Use your EDA results to show the impact of your tweak. Quantify the house edge (e.g., "In the fair model, the player lost an average of 0.01 per play, while in the tweaked model, they lost 0.08 per play").
7. **Conclusion:**
    - Summarize your key findings.
    - Discuss the importance of what you learned (e.g., "this demonstrates how models can be evaluated..." or "this shows how small probabilistic tweaks can guarantee a profit in a casino...").
    - Encourage further exploration.

Submission and Deliverables

You must submit a single link to a cloud drive folder (Google Drive/OneDrive).

- **Folder Name:** `LASTNAME_LASTNAME_LASTNAME_SECTION`
    - (Example: `IBO_GARCIA_DELACRUZ_3B`)

The folder **must** contain the following three items:

1. `github.txt`
   - A plain text file containing the public GitHub link for your project's source code.
2. `Project_web_link.txt`
   - Published link of the project
3. `Documentation.pdf`
   - Your complete project report, following the 7 steps outlined above.
4. `Video.mp4`
   - A 10-15 minute video (MAX) that includes a discussion, presentation of your findings (from the documentation), and a demonstration of your code.