| **Instructions:**  Evaluate the student’s submitted Final Project assignment and presentation against the outlined criteria in the rubric below and assign a rating to each criterion. Add points earned across all criteria and convert the total points to a letter grade using the *Recommended Final Project Scoring Breakdown*.  **Note:**  We encourage students to collaborate and share ideas during the project weeks. Therefore, you may notice shared code, documentation, and/or write-up explanations across student submissions. This is acceptable and should be a consideration when assigning a rating to the student’s performance. | **Recommended Final Project**  **Scoring Breakdown**   | **Total Rubric Points Achieved** | **Project Grade** | | --- | --- | | 90 or more | A | | 80–89 | B | | 70–79 | C | | 60–69 | D | | 59 or less | F | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|

**Rubric for Skill Drills:**

|  | **Proficiency**  **20 points** | **Approaching Proficiency**  **17 points** | **Developing Proficiency**  **14 points** | **Emerging**  **12 points** | **Incomplete** |
| --- | --- | --- | --- | --- | --- |
| **Data Model Implementation** | Student produces an analytical model in Python that fulfills all the following specifications:  √ Script initializes, trains, and evaluates a model, or loads a pretrained model from hyperparameter tuning  √ Script cleans, normalizes, and standardizes input data prior to modeling  √ Model utilizes data retrieved from a relational database or big data source (SQL or Spark)  √ Model demonstrates meaningful predictive power (>75% classification accuracy, >80 R-squared) | Student produces an analytical model in Python that fulfills all the following specifications:  √ Script initializes, trains, and evaluates a model, or loads a pretrained model from hyperparameter tuning  √ Script cleans, normalizes, and standardizes input data prior to modeling  √ Model utilizes data retrieved from a relational database or big data source (SQL or Spark) | Student produces an analytical model in Python that fulfills all the following specifications:  √ Script initializes, trains, and evaluates a model, or loads a pretrained model from hyperparameter tuning  √ Script cleans, normalizes, and standardizes input data prior to modeling | Student produces an analytical model in Python that fulfills the following specifications:  √ Script initializes, trains, and evaluates a model, or loads a pretrained model from hyperparameter tuning  -OR-  √ Script cleans, normalizes, and standardizes input data prior to modeling | No submission was received  -OR-  Submission was empty or blank  -OR-  Submission contains evidence of academic dishonesty |
| **Data Model Optimization** | √ Clear, well-documented evidence of model optimization and performance evaluation in the form of one of the following:  √ A CSV/Excel table showing model designs, testing parameters, and model performance  √ A Python script that utilizes hyperparameter tuning logic  -AND-  √ Overall model performance is printed or displayed at the end of the script | √ Some evidence of model optimization and performance testing within Python scripts  -AND-  √ Overall model performance is printed or displayed at the end of the script | √ Overall model performance is printed or displayed at the end of the script | √ Performance of the model is unknown/unclear |
| **Project and Documentation Uploaded to GitHub** | √ Successfully uploaded to GitHub; demonstrating professional quality of presentation  √ GitHub repository is free of unnecessary files and folders and has an appropriate .gitignore in use  √ The README is customized to a professional level | √ Successfully uploaded to GitHub; demonstrating professional quality of presentation  √ GitHub repository has minimal unnecessary files and folders (no more than two) and has an appropriate .gitignore in use  √ The README is customized to a basic level | √ Successfully uploaded to GitHub; demonstrating professional quality of presentation  √ GitHub repository has minimal unnecessary files and folders (no more than three)  -OR-  √ Does not use a .gitignore text file  √ The README is minimally customized | √ Unsuccessful uploads to GitHub  √ Does not use a .gitignore text file  √ The README has no customization |
| **Group Presentation** | √ All group members spoke during presentation  √ Group was well prepared    √ Presentation was relevant to material  √ Presentation maintains audience interest | √ All group members spoke but didn’t split time equally  √ Group was mostly prepared, with minor hiccups  √ Presentation was almost entirely relevant | √ Some group members barely spoke, others spoke for much longer  √ Group was fairly well prepared but encountered some major hiccups  √ Presentation was mostly relevant | √ Not all group members spoke during presentation  √ Group seemed unprepared, presentation was scattered or confusing  √ Presentation was not relevant to material |
| **Slide Deck** | √ Slides are visually clean and professional  √ Slides are relevant to material  √ Slides effectively demonstrate project  √ Slides are clear and maintain audience interest | √ Slides are visually clean and professional but contain minor areas for improvement  √ Slides are almost entirely relevant to material  √ Slides are mostly effective at demonstrating project | √ Slides are visually clean and professional but contain areas for improvement  √ Slides are somewhat relevant to material  √ Slides are somewhat effective at demonstrating project | √ Slides are not visually clean and professional and contain substantial areas for improvement  √ Slides are not relevant to material  √ Slides do not effectively demonstrate project |