**FinTech Unit 13 AWS Homework Grading Rubric**

**Both Options**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Ratings** | | | |
| **Coding Conventions/Formatting**  • Appropriate header, name, short description at top of the notebook • Imports are at the top of the file, just after any headers or subheads. • Files read in from relative file path • Functions and variable names are descriptive, lowercase, with words separated by underscores • Clean code, no repetition, maintainable and highly reusable code. • Appropriate code wrapping and cell sizes • Appropriate subheads as needed | **10 Points Mastery** | **9 Points Approaching Mastery** | **8 Points Progressing** | **8 > 0 Emerging** |
| **Deployment/Submission**  • Files submitted in personal repo • Appropriate directory structure with correct files needed to run scripts • Appropriate commit messages • Appropriate README | **10 Points Mastery** | **9 Points Approaching Mastery** | **8 Points Progressing** | **8 > 0 Emerging** |
| **Documentation/Comments**  • Code is well commented with concise, relevant comments | **10 Points Mastery** | **9 Points Approaching Mastery** | **8 Points Progressing** | **8 > 0 Emerging** |

**Option 1**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Ratings** | | | |
| **Initial Robo Advisor Configuration**  • **RoboAdvisor** created with proper parameter. • **RecommendPortfolio** created and configured with proper name utterances. • **RiskLevel** custom slots created with proper card slots. • **RoboAdvisor** tested after build with error handling configuration. | **35 Points Mastery** • Completed 4 out of 4 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenarios  • Code is free of bugs | **34 > 28 Points Approaching Mastery** • Completed 3 out of 4 of requirements • Code runs without error • Code produces results as expected 80% of the time | **28 > 23 Points Progressing** • Completed fewer than 2 out of 4 requirements • Code runs without error  • Code produces results, but not necessarily the correct results | **23 > 0 Emerging** • Completed 1 or none out of the 4 requirements • No submission • Code runs with error |
| **Enhance RoboAdvisor with Amazon Lambda Function**  • **User Input Validated.** • **Investment Portfolio Recommendation** given on **selected** **risks.** • **Lambda Function** tested with **sample test cases. • Lambda Function** integrated to the **RoboAdvisor.** | **35 Points Mastery** • Completed 4 out of 4 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenarios  • Code is free of bugs | **34 > 28 Points Approaching Mastery** • Completed 3 out of 4 of requirements • Code runs without error • Code produces results as expected 80% of the time | **28 > 23 Points Progressing** • Completed 2 out of 4 requirements • Code runs without error  • Code produces results, but not necessarily the correct results | **23 > 0 Emerging** • Completed 1 or none out of the 4 requirements • No submission • Code runs with error |

**Option 2**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Criteria** | **Ratings** | | | |
| **Data Preprocessed**  **•** Data loaded using Pandas DataFrame into **crypto\_df. •** Data Preprocessed with the assigned preprocessing tasks.  **Data Dimension Reduced**  **•** **PCA** algorithm from **sklearn** used to reduce dimensions. **•** **pcs\_df** DataFrame created and crypto\_df.index used as the index for **pcs\_df** DataFrame. | **35 Points Mastery** • Completed 4 out of 4 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenarios • Code is free of bugs | **34 > 28 Points Approaching Mastery** • Completed 3 out of 4 of requirements • Code runs without error • Code produces results as expected 80% of the time | **28 > 23 Points Progressing** • Completed 2 out of 4 requirements • Code runs without error  • Code produces results, but not necessarily the correct results | **23 > 0 Emerging** • Completed 1 or none out of the 4 requirements • No submission • Code runs with error |
| **Cryptocurrency Clustered**  • **K-Means** used to cluster the cryptocurrencies using **PCA** data. • **Elbow Curve** used to find the best value for **k**, using the **pcs\_df** DataFrame. • Kmeans algorithm used to predict the **k** cluster for cryptocurrency data. • New DataFrame created named **clustered\_df**, includes assigned columns and index.  **Visualizing Results**  • 3D-Scatter plotted created using **Plotly** using **clustered\_df** DataFrame, paramaters used as directed. • Data table created using **hvplot.table** for all current tradable cryptocurrencies, columns used as directed. • Scatter plot created using **hvplot.scatter,** to present clustered data with directed parameters. | **35 Points Mastery** • Completed 7 out of 7 requirements • Code runs without error and produces the assigned results • Code accounts for all possible scenarios • Code is free of bugs | **34 > 28 Points Approaching Mastery** • Completed 5 out of 7 of requirements • Code runs without error • Code produces results as expected 80% of the time | **28 > 23 Points Progressing** • Completed 3 out of 7 requirements • Code runs without error  • Code produces results, but not necessarily the correct results | **23 > 0 Emerging** • Completed less than 3 of the 7 requirements • No submission • Code runs with error |
| **Optional Challenge**  • Jupyter notebook uploaded to **Amazon SageMaker** and deployed. | **20 Points Mastery** | **15 Points Approaching Mastery** | **10 Points Progressing** | **5 Emerging** |