Capstone Project Expectations

GENERAL EXPECTATIONS

Area	Successful Project	Incomplete Project
Process Documentation	Project has a README file clearly documenting each step in a data science workflow, as well as data understanding, project goals, etc.	The README is missing or incomplete, or some workflow steps are not documented fully.
Reproducible Science	Any data scientist could reproduce your work by following the simple instructions found in the README.	Instructions to reproduce the work are incomplete or confusing, or manual steps are required to prepare data.
Narrative	Project has a clearly defined story, set out in the README and other documentation, that ties the goals of the project together with the steps taken to accomplish those goals.	Project is thrown together, with data science concepts and techniques applied but no real sense of why you undertook your project or how you are tackling the defined problem.
Copy Editing	All documentation, including the README, are well written with proper spelling and grammar.	One or more obvious typos or grammar errors exist in the README or elsewhere.
Sourcing	When applicable, citations provide the origin of other people's code, images, etc.	Project contains plagiarized or uncited content.

BUSINESS UNDERSTANDING

Area	Successful Project	Incomplete Project
Problem Definition	The README clearly explains the problem that the project sets out to solve.	The project has vague or unclear goals, or no sense of what problem they are trying to solve.
Success/Evaluation Criteria	Criteria for evaluation are laid out in the Business Understanding section of the README.	The README does not describe evaluation criteria, or does not explain why metrics were used.
End Result	The data product/result directly answers the business question or addresses the specified problem.	There is no final data product, or the data product is not relevant to the stated business question.

DATA UNDERSTANDING

Area	Successful Project	Incomplete Project
Data Acquisition	The README explains where and how the data was obtained, such that the data acquisition process could be reproduced.	The README fails to explain where or how someone could get the data.
Example Data	If the dataset is public/scraped is sufficiently small, the repo contains the dataset. If the file is not included, a sample of data (potentially fake or anonymized data) is provided so that others can replicate the pipeline for illustrative purposes.	If the dataset is not included, the README does not explain the reasoning behind withholding data (such as confidential data, or dataset is too large) and no example data is provided. Or, arguably worse, private data is included without anonymization.

DATA PREPARATION

Area	Successful Project	Incomplete Project
Data Preparation	Original raw data is preserved. All steps to clean and prepare data are explained, contained within functions, and reproducible.	Raw data was manually altered, or some data preparation steps are not documented in code. Data are cleaned and processed piecewise in notebooks.
Cross Validation	Cross validation is used correctly to evaluate model performance.	Cross validation is not used, or is used incorrectly.
Data Leakage	All models and transformers are fit only on the training data, not on the entire dataset. The model is not trained on features that leak information about the target variable for the entire dataset.	Model features leak information during model selection and hyperparameter tuning, or transformers and models are fit on the entire dataset.

MODELING

Area	Successful Project	Incomplete Project
Baseline	Simplest model (or model-less baseline) is attempted first before trying more complex or less interpretable models. Models are compared to the baseline consistently and coherently.	No sense of how well the final model does compared to a simpler model. More complex models are not compared to the baseline or justified.
Documentation	Documentation and markdown cells explain modeling decisions and processes, so another data scientist can understand the reasoning throughout the project.	The only form of documentation is in-line comments, or no explanations are provided.

PRESENTATION

Area	Successful Project	Incomplete Project
Visualizations	Discussion points are presented visually whenever possible, and materials are prepared and polished for presentation.	Presentation materials feature unpolished content that distracts from the presentation (lack of legends, unlabeled axes, etc.).
Video	A non-technical presentation is recorded as a 3-7 minute video, linked in the README.	Presentation was not recorded or was not shared. Video is too technical or too long.
Sharing	Presentation materials, including any slides and the recorded video, are available online.	The presentation materials are incomplete or are not publicly available.

CODE QUALITY

Area	Successful Project	Incomplete Project
Python Files	Custom functions and classes are contained in .py files which are then imported.	Some or all of the Python code needed to reproduce the work is written within various notebooks.
Docstrings	Each Python module, function, or class has a docstring containing at least a one-line description.	Some or all docstrings are missing, incomplete, misleading, or out of date. Or, comments are used in place of docstrings.
Naming Style	Object names follow PEP8 conventions (i.e. classes are written in CamelCase, functions are written in snake_case).	The naming of objects is inconsistent. Object names use a confusing or arbitrary naming style, e.g. ‘MyFunction’ or ‘f.’
Bonus: Class Creation	Functions are bundled into class objects that can be reused throughout analysis. Where applicable, the sci-kit style and method calls are implemented.	

VERSION CONTROL

Area	Successful Project	Incomplete Project
Project Organization	All project code is contained in a GitHub repo with separate subdirectories to organize source code and data.	The project repo is incomplete, or the organization of the repo is confusing.
Repository Name	The repo name is creative and original or describes what the project does to solve a problem.	"Capstone" or "Project" appears in the repo name, or the name does not describe the project.
File Naming	File names are not used for version control. File names follow a consistent naming convention (i.e.: lowercase, no spaces).	File names follow an inconsistent naming convention, or file names contain words like "new" or "final" or "v2".
Git Commits	The project shows a steady commit history during the project period. Each commit message is written clearly and professionally, and describes the high-level intent of a specific change.	The commit history is sporadic, or large portions of work are contained within a few monolithic commits. Commit messages are written in an unprofessional style, or do not describe the changes.
.gitignore File	The project contains a .gitignore file based on GitHub's default .gitignore file for Python projects, and no personal information is accidentally shared.	There is no .gitignore file, or the repo contains unnecessary directories and files such as .DS_Store. API keys or other sensitive details are in the repo.
Branching	The final project lives on the master branch of the repo. Other branches, such as unmerged feature branches or historical checkpoints, may exist.	The final project is stored somewhere other than the master branch, or it is necessary to view multiple branches to see the entire project.

NEXT STEPS

Area	Successful Project	Incomplete Project
Model Improvement	Documentation or README includes potential ways to improve the model, such as through feature engineering, parameter tuning, etc.	No reflection providing ideas for how the model could be improved.
Product Improvement	Project documentation includes ideas to continue the project that further address the original business problem.	The project does not include ideas for future improvements or any reflection on ways to address the original problem.
Web Presence	The project is deployed online (e.g.: pythonanywhere, Heroku), or written up publicly on a blog or personal site.	The project has no presence on the web beyond GitHub.