# Statistical Modelling with Python

Tue Jan 17

W03D5 (1)
2 hrs
Model Evaluation and Intro to Project

Project Details »

POWERED BY
LIGHTHOUSE LABS


Welcome to your Statistical Modelling with Python project! This project is an opportunity for you to combine all of the Python skills you have learned up until this point.

## My Submissions

| Status | Marker | Stats | Date Submitted | Project | |
|---|---|---|---|---|---|
| In Queue | | | w02d2 2023-01-17 | View Statistical Modelling with Python | Cancel |

Description | Eval Rubric

## Evaluation Rubric

Evaluators use this form and select one of the options and provide written feedback (where applicable) for each section below. If any of the sections are marked as *Unsatisfactory*, the evaluation is rejected.

| | Exceptional | Acceptable | Needs Work | Unsatisfactory |
|---|---|---|---|---|
| **Python** | Starting to implement more advanced Python programming techniques such as object oriented programming (creating custom objects). | Implements Python best practices to create functions. | Can distinguish between Python objects and can create basic functions but does not always observe best practices. | Can recognize Python as a language but cannot differentiate between different Python objects. |
| **SQL** | Implements advanced SQL techniques/skills (i.e. changing data type or creating custom field within SQL query) | Uses SQL queries to access data they need and is able to join data from multiple tables. | Can create basic SQL queries but cannot yet join data from separate tables. | Does not have enough syntax knowledge to use SQL to access the data they need. |
| **Data Visualization for EDA** | Creates optimal visualizations and draws initial insights into potential relationships in the data. | Chooses correct chart type to support EDA but struggles with identifying patterns or relationships demonstrated in visualization. | Creates visualizations to understand data but visualization type cannot be used to gain any insights. | Does not use visualization as part of EDA process. |
| **API Interaction** | Independently searches for, identifies and connects to APIs to access relevant and reliable open data sources. | Can find and connect to specified APIs. | Able to access a specified API but has limited technical skills to use an API. | Can identify the steps required to connect to an API but cannot connect to the API. There is limited understanding of how an API works. |
| **Data Acquisition** | Finds relevant data sources with thought given to source reliability and the data quality. Parses data efficiently. | Finds and sets up connection with data sources to access relevant data tables and parses unstructured data to solve a problem. | Can find and work with simple data sources, such as basic data found online and parse simple structures. | Must be provided with data files, does not have technical skills to access and parse any data source. |
| **Data Audit** | Identifies potential issues with quality and quantity of data and fixes issues at the source of the data . Generates hypothesis of what can be done with the data. | Assesses quality of data or data source and identifies potential issues by checking for missing values and duplicate rows in addition to other standard techniques | Conducts data quality assessment but has limited understanding of the implications of any quality assessment (uses incomplete QA techniques or incorrectly applied). | Cannot identify steps to follow to assess quality of data. |
| **Exploratory Data Analysis** | Uses patterns found during EDA to answer hypotheses generated during audit process through statistics or appropriate visualization techniques. | Identifies basic patterns in data and selects suitable analysis techniques based on data type. | Implements EDA but does not yet understand full value of insights to be gained from EDA. | Can speak to EDA as a process at a high level but does not have the skills to execute. |
| **Data Cleaning** | Thinks beyond basic data cleaning techniques to consider additional data issues (i.e. capitalization of text) | Cleans data to address potential issues by using outlier detection techniques, replacing null values with correct information, and deduplicating data. | Implements incomplete set of data cleaning techniques. | Aware of data cleaning techniques but lacks technical skills to implement them. |
| **Data Wrangling** | Able to parse data, work with a variety of data files including JSON, and CSV files and combine data from different sources. Can create one source table that can be leveraged for visualization and modelling purposes. | Able to parse data, work with a variety of data files including JSON, and CSV files and combine data from different sources. Can create one source table that can be leveraged for visualization and modelling purposes, however, does not remove unnecessary data from the source table. | Attempts to work with a variety of data files, but incorrectly combines data from a variety of sources resulting in an unreliable DataFrame. | Aware that different data file types exist but can only work with one file type at a time and cannot convert one to another. |
| **Interpreting Model Output** | Has strong understanding of model outputs and can leverage outputs to draw insights, enhance data understanding and make predictions in a business context. | Interprets model correctly but does not incorporate business context into meaning of results. | Leverages outputs to draw insights, but Interprets model output incorrectly. | Has limited understanding of how to interpret model outputs. Cannot leverage outputs to draw insights from data. |
| **Tool Installation & Set Up** | Recommends problem specific tools and can set up any tool stack required. | Aware of a wide range of industry-standard tools and their pros and cons. Can set up and work with basic tools required for the project. | Can install pre-determined tool stack, and has basic understanding of the set up and tool environment. | Cannot install tools or set up environment. |
| **Coding Comprehension** | Comments code using best practices for variable names, syntax etc. Further explanation of code is not required. | Writes well commented code that others can understand independently. Has little syntax errors but applies best practice inconsistently. | Code and comments need further clarification. | Code is not sufficiently comprehensible without an explanation. Missing or unclear comments. |
| **Organization** | README.md is well written, easily understood, and precise instructions are included. Files created can easily be shared with others. File organization allows for easy collaboration with others. | Keeps files organized and appropriately named. Able to maximize Github file management to easily collaborate with others. Uses generic README.md that provides adequate description. | Files are somewhat organized but README.md file doesn't match content of folder, does not have enough explanations. Naming of files and file structure could be improved so work is more easily shared. | Files are not properly organized, could not be shared easily with others. Teammates unable to work with files created. Missing files. |