

Component/Design Specification

Software Components

1) Visualization Component:

Purpose: The visualization component's main purpose is to give a clear idea of how many alternative-fuel charging stations exist in the nation and the breakdown of each kind of station that exists.

Input Required: Input required to enable this component is broken-down government data about the nation's charging stations as well as user clicks and hovers around our map

Output: The output of this visualization is a clear state-by-state visual analysis so users can know in what states what kind of charging stations exist for their car.

2) Classification Component

Purpose: The classification component is in charge of recommending the user an electric vehicle based on the "survey" that the user takes at the beginning.

Input: Input required to enable this component is the user's answers to the survey of preferences when it comes to their ideal car

Output: A recommendation for the brand of electric car they should based on their preferences as well as a short summary of why that particular car is the best choice for them.

3) Data Cleaning Component:

Purpose: All things related to initial data cleaning and data loading will be included in this component.

Input: Data that has been scraped from the web and loaded in using the pandas library.

Output: Well-structured datasets that can be used for our two main tools, classification and visualization.

Interactions to accomplish use cases

Our system's two-pronged approach with classification and visualization will take into account things that a user can control well, (i.e. their car preferences) and things that a user cannot control well (where they live). Our data cleaning

component will make sure our data is in a good environment for analysis with our classification and visualization tools importing our data straight from that component. Our visualization will be mostly centered around making the user more comfortable with their electric vehicle choices by detailing the charging stations available to them in their particular state. After that initial information is given, the user will then pick their preferences for what they want and like in a car, and the classification model will use that information to recommend a car and give a short summary of why that car might be the best fit for them.

Preliminary plan

- 1) **Data collection:** We are planning to use 4 different datasets from various sources at the base level before aggregation. We get our data downloaded and ready in our repository and our workspace.
- 2) **Data aggregation:** Load in all our datasets into different Python notebooks in the Google CoLab suite and examine the shape
- 3) **Data cleaning:** Merge our datasets as necessary, clean the data into formats we will need for visualization and machine learning analysis
- 4) **Visualization Building:** Import visualization libraries and our visualization-based dataset and begin experimenting ways in which we can create meaningful, interactive, and informative visualizations
- 5) **Feature Building for Classification:** For our classification tool, we have to make our features readable and clean for the users. This step is to pre-process those features.
- 6) **User Interaction for Classification:** Create a “survey” for the user and have their responses be stored in a dictionary we will use in our classification later.
- 7) **Format Analysis for User Responses:** Clean our user’s responses so that they are translated into input values into our backend machine learning model.
- 8) **Choose a classification model:** Utilize accuracy scores for several different classification models to determine which is the best one to use going forward.
- 9) **Feed inputs into model:** Input the user’s responses into our chosen models and summarize results for the user
- 10) **Building Unit Testing:** Build out the unit tests for user input validation, our data cleaning tools, and visualization’s data access
- 11) **Program Run:** Make sure we have one *setup.py* file that will be able to execute our entire project (classification and visualization) together in a streamlined and aesthetic way

12) Directory structure: Transfer all our notebooks, unit tests, data, and documents into our GitHub repository and make sure our structure is up to date