Estimation of Charging Costs of Electric Vehicles

By Lusine Baghdasaryan, Ph.D.

Summary

This application assists you evaluate electric vehicle options by estimating their charging costs. Based on your vehicle preferences, your geographical location and driving intensity, you will be able to compare charging costs of multiple vehicles. The application also provides visualizations of charging costs across different US cities.

Problem Statement

Electric vehicles have recently become popular due to their environmental friendliness, low emissions, and reduced dependence on gasoline. However, several key factors play a role in purchasing decisions. Battery performance, the range a vehicle can travel on a charge, charging infrastructure, make and model, and cost affordability, are not only important factors but also, they are interrelated. This app aids the user in this complex decision-making process by using vehicle battery capacities, expected annual travel distances, and local electricity prices to estimate vehicle charging costs over the timeframe of ownership. The user inputs the following parameters on the app form: a city from top 22 largest US cities provided, up to five vehicles at a time, annual miles and number of years expected to drive. The app then returns the estimated annual costs on a map and a breakdown of monthly costs on a chart.

Project

The app uses the following two types of datasets:

1) List of electric vehicles, along with technical configurations, are scraped from https://ev-database.org/ electric vehicle database. **Lambda Function 1** runs four times a month and saves “electric_vehicles.csv” file in “data” folder on S3 bucket. A copy of the function is stored in “codes/lambda function vehicle” folder. The app reads “electric_vehicles.csv” file from s3 during the runtime. Vehicle information selected by user is also provided to the user in the app as shown in the figure below:

Image		make	model	acceleration	topspeed	battery	erange_real	efficiency	fastcharge_speed	country_uk	img1_
Image 1	0	Tesla	Model Y Long Range Dual Motor	5.0 sec	217 km/h	75	435 km	172 Wh/km	670	£52,990	https:/
Image 2	1	BMW	iX xDrive40	6.1 sec	200 km/h	71	360 km	197 Wh/km	480	£69,905	https:/

Figure 1: Vehicle information scraped from vehicle database website and presented to the user.

2) The “data” folder on S3 folder houses “cities_series.csv” file for the top 22 largest US cities. **Lambda Function 2** reads that file and runs monthly to extract historical electricity prices per KWH from https://api.bls.gov/publicAPI/v2/timeseries/data/. A copy of Lambda

Function 2 is stored in “codes/ lambda function bls” folder. A Developer’s API key is used to extract the data and create time series models for each of the cities. At this point, Lambda Function 2 produces “cities_geocoded.csv” and “series_data1.csv” files. The latter one holds raw data from bls.gov API. These two files are saved for intermediate data checks. Then Lambda Function 2 imputes the data and creates time series models for each city. The “city_prices_imputed.csv” file holds imputed data, and the trained models are saved in “models” folder on S3. Lambda function then runs predictions for each city and saves them on “predictions” folder. This is also done for data checks. Actuals/predicted charts were created during analysis for each city. Even though the chart below is not shown to the user, it is provided here as part of the analysis. The blue is historical prices, and the green is the predicted prices per KWH for Phoenix.

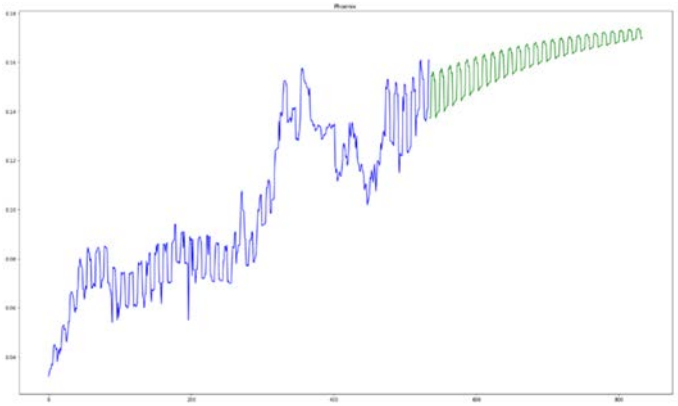

Figure 2. An example of forecasted electricity prices for Phoenix, AZ.

The app requests that the user select up to 5 vehicles and a city from the drop-down lists and provide annual distance and number of years to drive. The app reads “electric_vehicles.csv” file from “data” folder and the “.pkl” files from “models” folder. The app calculates the expected costs by utilizing battery energy requirements for the selected vehicles and the rest of the collected information. The app renders the calculation results on the screen and saves them as “.json” files in “vehicle_costs.csv” folder on S3 . A sample output is shown in Figure 3 below. As shown in Figure 3, the app provides a popup for the total costs of owning each of the vehicles for the years selected. The app also provides a chart with monthly estimated costs for each vehicle. Figure 4 shows the full data pipeline.

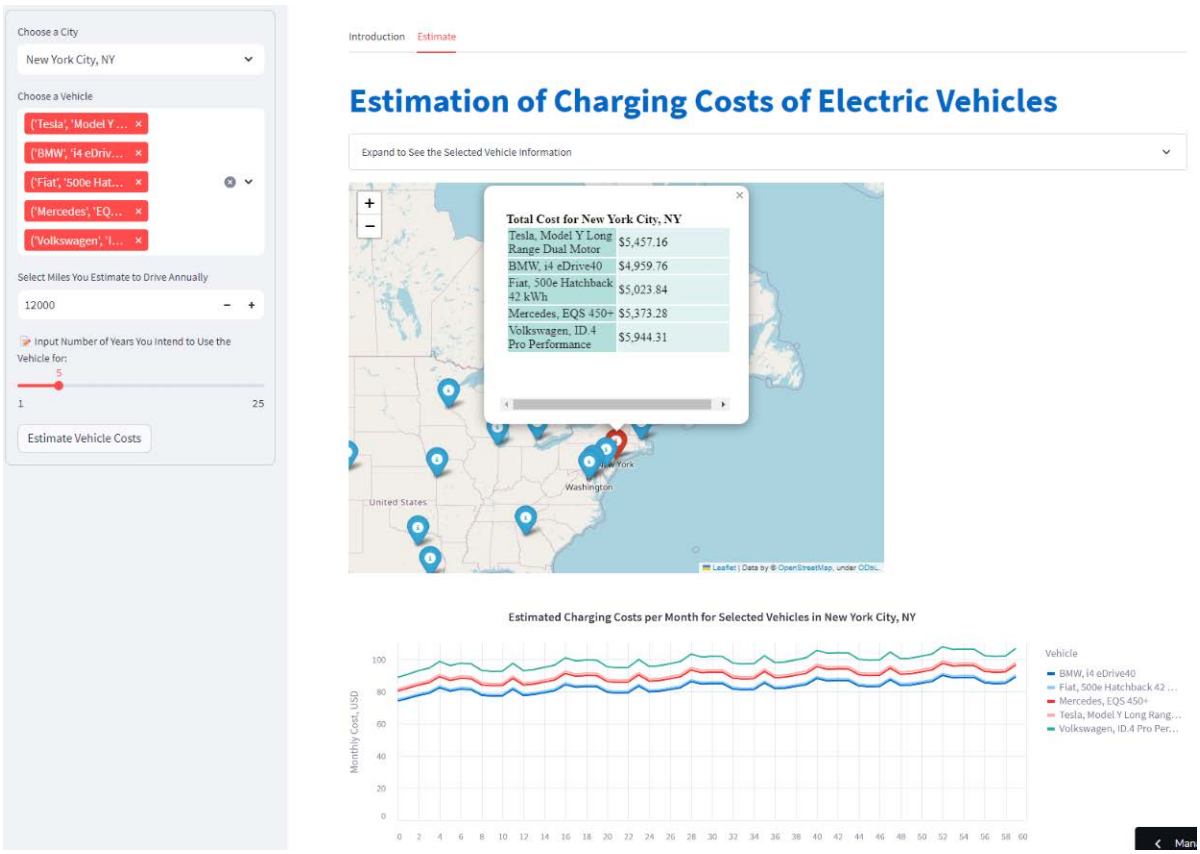

Figure 3. A screenshot of the app.

Data Pipeline of Vehicle Cost App

Lambda function 1:

Scrapes https://ev-database.org/ and writes to AWS S3.

Lambda function 2, runs monthly, does the following:

1) Reads the list of cities, geocodes and writes back S3.
2) Extracts historical electricity consumer prices from https://www.bls.gov/ with API
3) Imputes data to prepare for time series modeling
4) Creates time series models with pmdarima
5) Saves trained models for each city to S3
6) Runs predictions for each city and writes to S3

Streamlit APP:

1) The app reads vehicle data and ARIMA models from AWS
2) User makes selections in the app
3) Costs are calculated in the APP
4) Cost data are stored on AWS S3

Figure 4. The Data Pipeline

Deliverables

The app is built in streamlit and deployed by https://share.streamlit.io/. The lambda functions, the app code and supporting files are housed in Github.

https://tdi-capstone-electric-vehiclesgit-u3y3kobjpb.streamlit.app/
https://github.com/blusine/tdi-capstone-electric-vehicles.git