

Switching to Electrical Vehicles   
MIS 4460 Business Analytics Project

Zahra Khan Vasudha Gulati Kayla Luchenbach Nitya Gatla

# Executive Summary

The demand for fuel-efficient vehicles has increased recently, owing to the rise in gas prices. This is attributed to depleting fossil fuel reserves and the uncertain political climate. These factors give rise to the need for advanced fuel-efficient technologies, leading to a surge in demand for electrically powered vehicles for travel. For this project, we chose a data set from Kaggle that contains the key attributes to consider when purchasing an Electric Vehicle. Our analysis compared Electric Vehicle market prices and features for multiple brands and made EV recommendations for potential buyers thus aiding in decision making. When creating the decision support tool, we focused on getting inputs from the customer to base the recommendations on their preferences and needs. These inputs included family size or number of seats needed in the vehicle, preferred body style of the vehicle, cost of electricity per KwH (if known), preferred range (in miles), and the daily commute distance of the customer (in miles).

With the decision support system we’ve developed, potential EV owners can now see their top options and get a good sense of the running costs they can expect for each EV.

# Target Users or Analysis Consumers

Our Electric Vehicle Purchasing Decision Support System is intended to support **potential buyers** in the decision-making process for the acquisition of an electric vehicle

# Technical Summary

**Primary software used**: Microsoft Excel and VBA

**Dataset/Data Source:** We leveraged an Electric Car Dataset from [Kaggle](https://www.kaggle.com/datasets/geoffnel/evs-one-electric-vehicle-dataset)

Chart

Description automatically generated with low confidence

**Data Model Design:**

1. Applied formulas to convert to USD and Miles
2. Leveraged a user form in VBA to input values
3. Stored incoming values in variables
4. Used subroutines to:

* Calculate cost per mile, running costs and yearly mileage
* Highlight EV rows that meet user criteria
* Copied highlighted rows to separate output sheet

1. Relaunched the user form once user was done viewing results

**Design Details:**

* In our EVgo model the column headings that are in blue are the original columns from the dataset and the columns with yellow heading are fields that we calculated so that the data would make more sense to Americans. For example, we converted KmH to MpH and Euros to USD
* We created a VBA user form to have the customer input their desired values for family size or number of seats needed in the vehicle, preferred body style of the vehicle, cost of electricity per KwH (if known), preferred range (in miles), and the daily commute distance of the customer (in miles)
* These numbers were stored into variables and displayed on our output worksheet
* Stored values were then used to calculate additional metrics like yearly mileage, running costs, cost per mile etc.
* We created a subroutine that highlights any rows in the Electric Car Dataset that match the user inputs. For non-matching conditions the subroutine also displays an error message
* Once rows are highlighted in the dataset, another subroutine was created to copy all the colored rows over to the output worksheet
* The output worksheet then shows the user criteria and the top choices that match
* Additionally, there is also a relaunch button on the output worksheet if the customer would like to re-run the tool with a new set of criteria

If we had more time to advance our model, we would have wanted to write code to display “n” top choices for the customer.

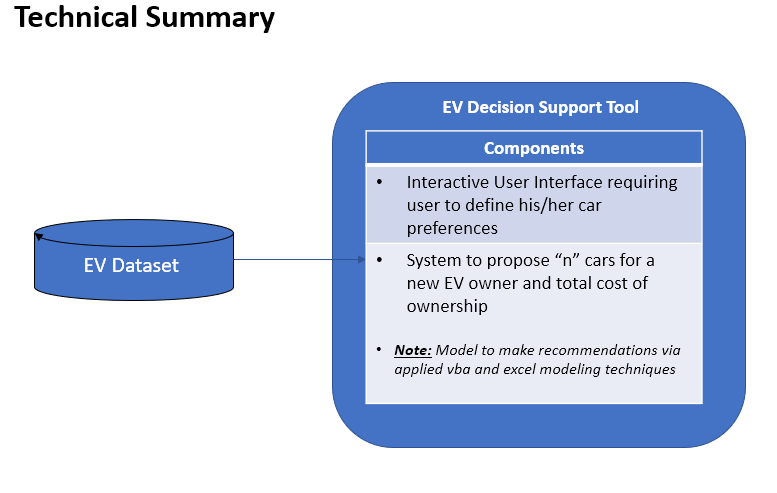


Figure Technical Diagram

# Data Needs and Sources

The [Electrical Vehicles Dataset](https://www.kaggle.com/geoffnel/evs-one-electric-vehicle-dataset) was obtained from Kaggle. The dataset was based on European data, therefore the speed, acceleration, efficiency, and range used Km as the unit of measure. We converted all these values to miles so that the American user would be able to understand the data being presented. Also, the price of each vehicle was originally shown in Euros which we converted to USD for easy understanding as well. All the data we are using is stored in our file under the worksheet named “Electric Car Dataset.”

Graphical user interface, table

Description automatically generated

Figure Electric Car Dataset

# Outputs

* When a user launches EVgo model, he/she should see the following screen come up on opening the excel file

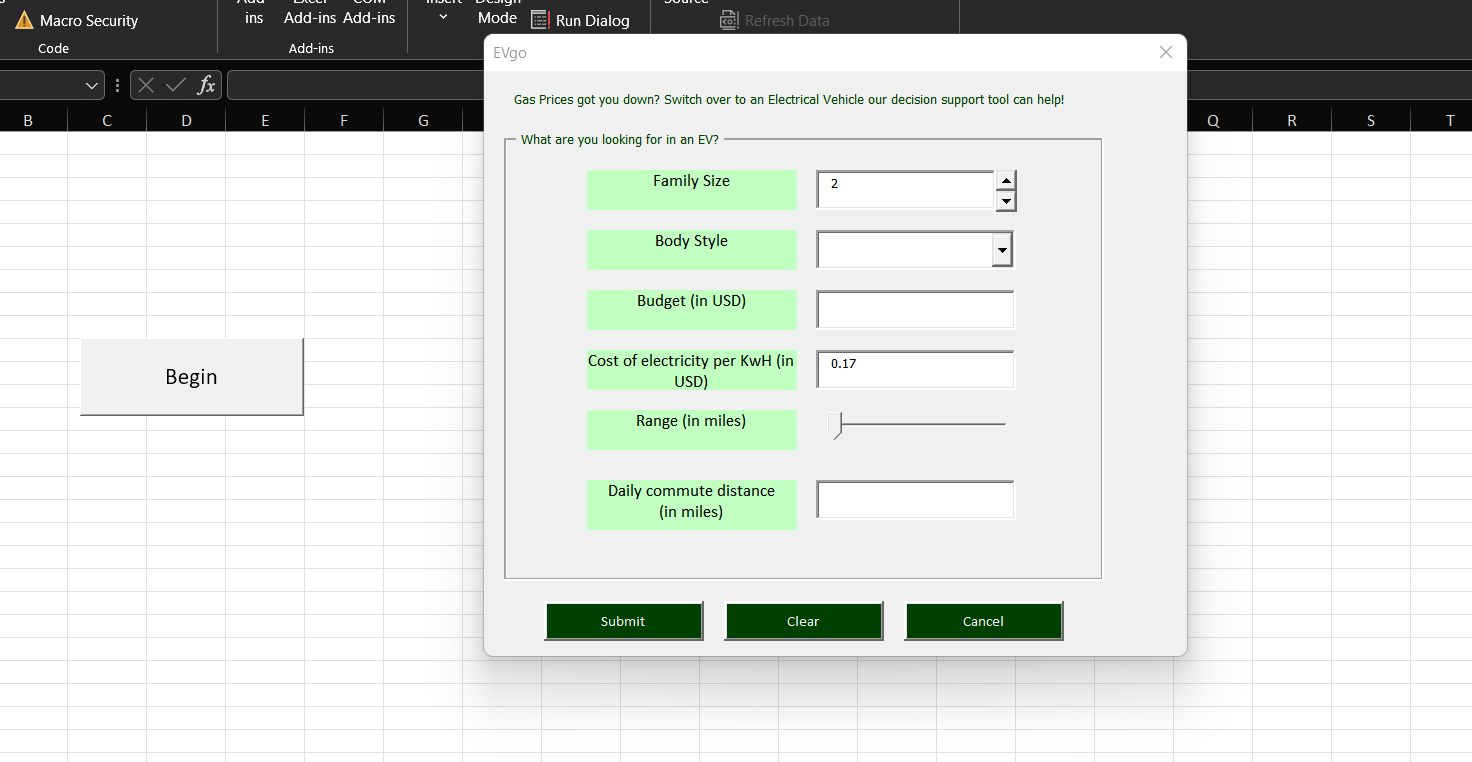


Figure User form displayed on opening the model

* User is prompted to fill out the form

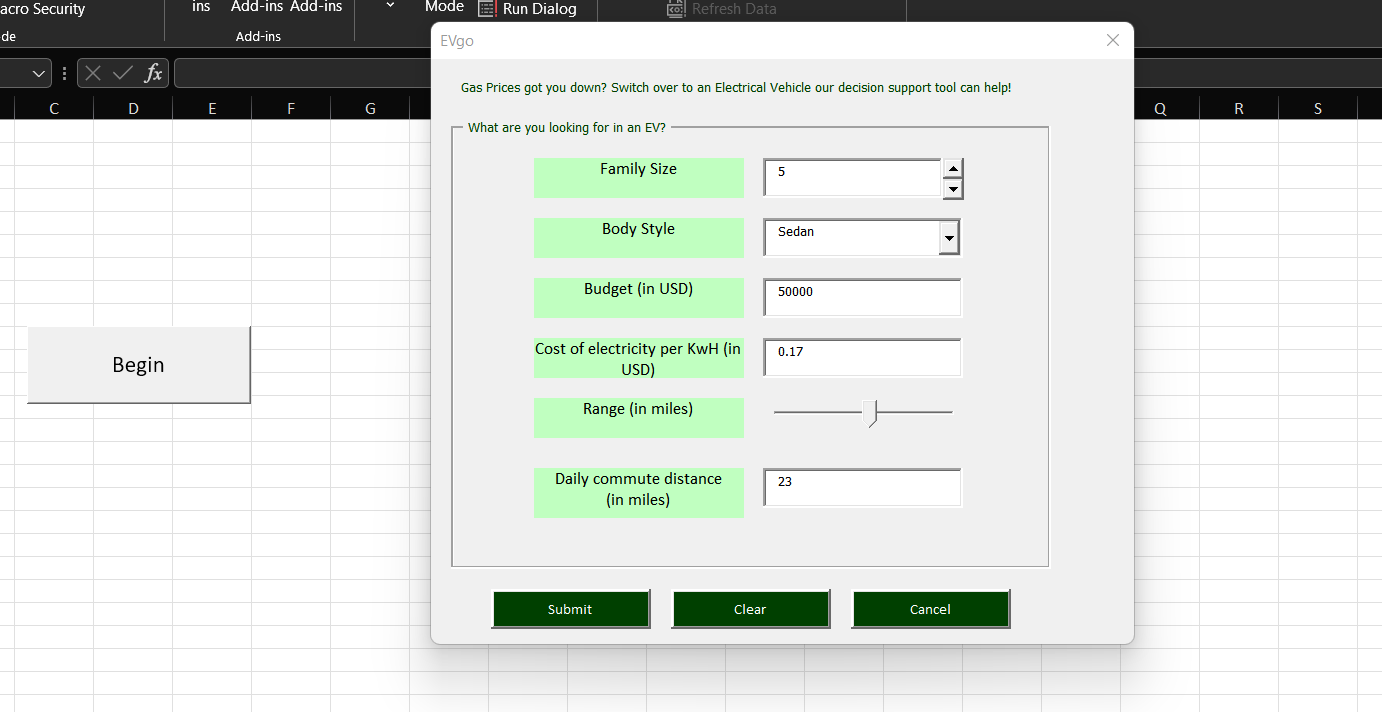


Figure User fills out values in user form

* If nothing matches the criteria, a validation message pops up informing the user that no EV meets his/her criteria

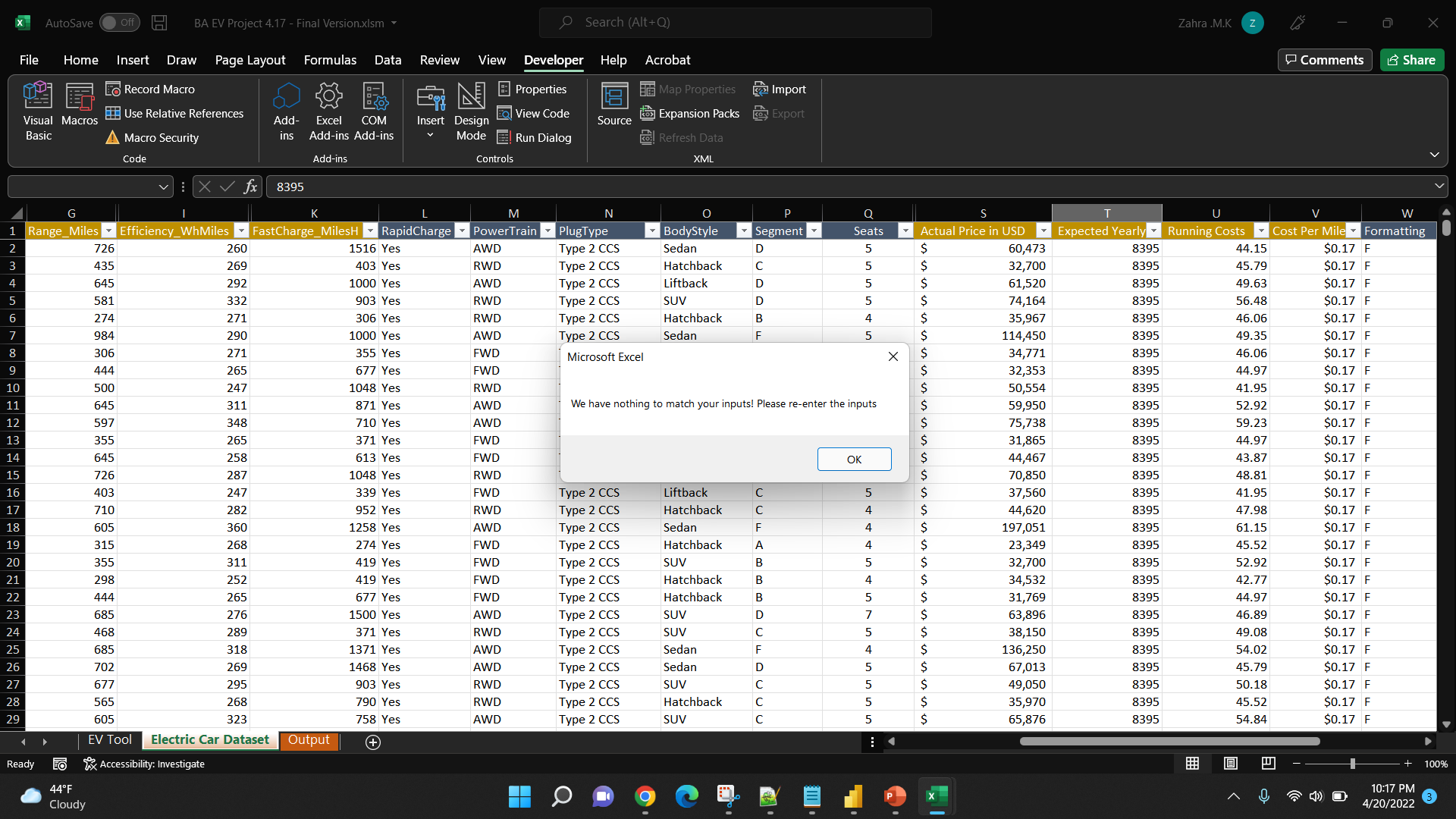


Figure If no values found, an error message comes up

* If there are values matching the user criteria, they will be highlighted in blue in the Electric Car Dataset tab
* Note that when the model is run and values are found user is directly navigated to the output screen, this was added for demonstration purposes only

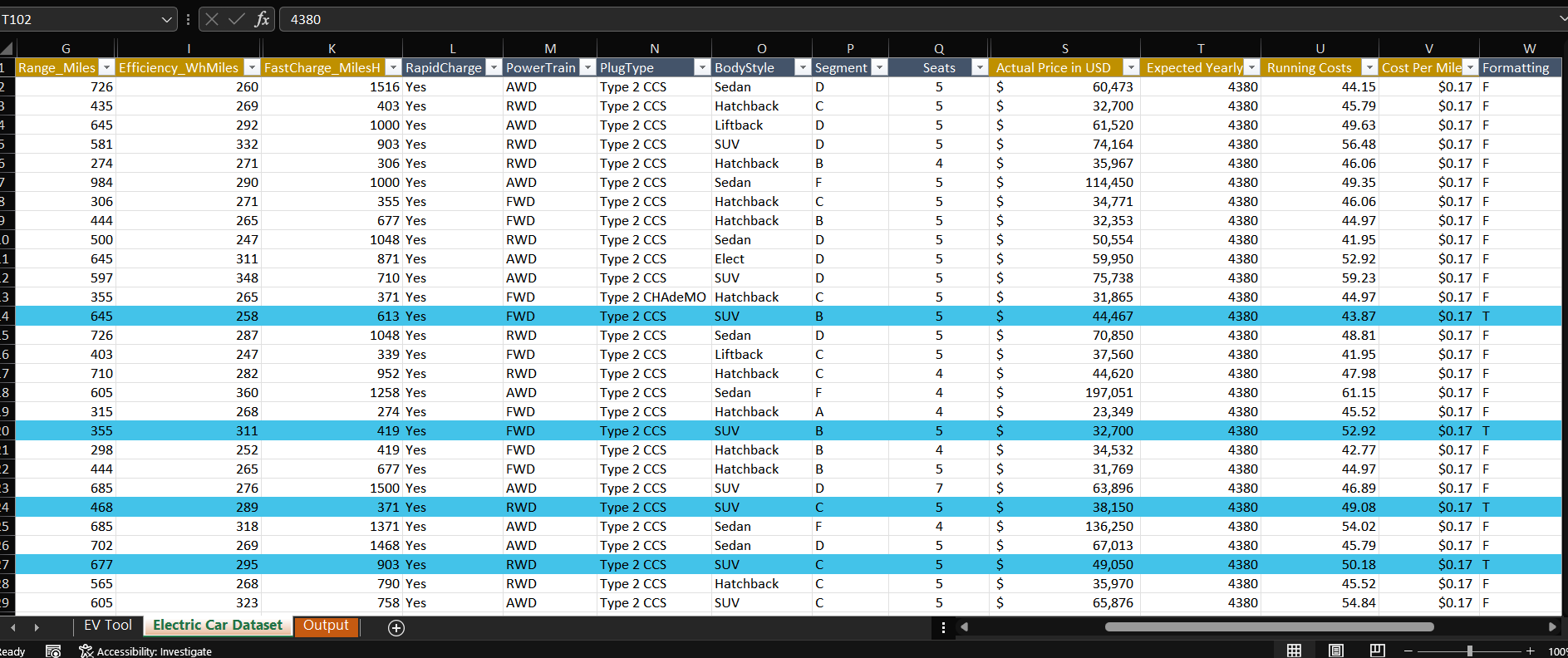


Figure If values found, they are highlighted in the Electric Car Dataset tab

* Output screen showcases the user entered criteria on the top and the possible matches below

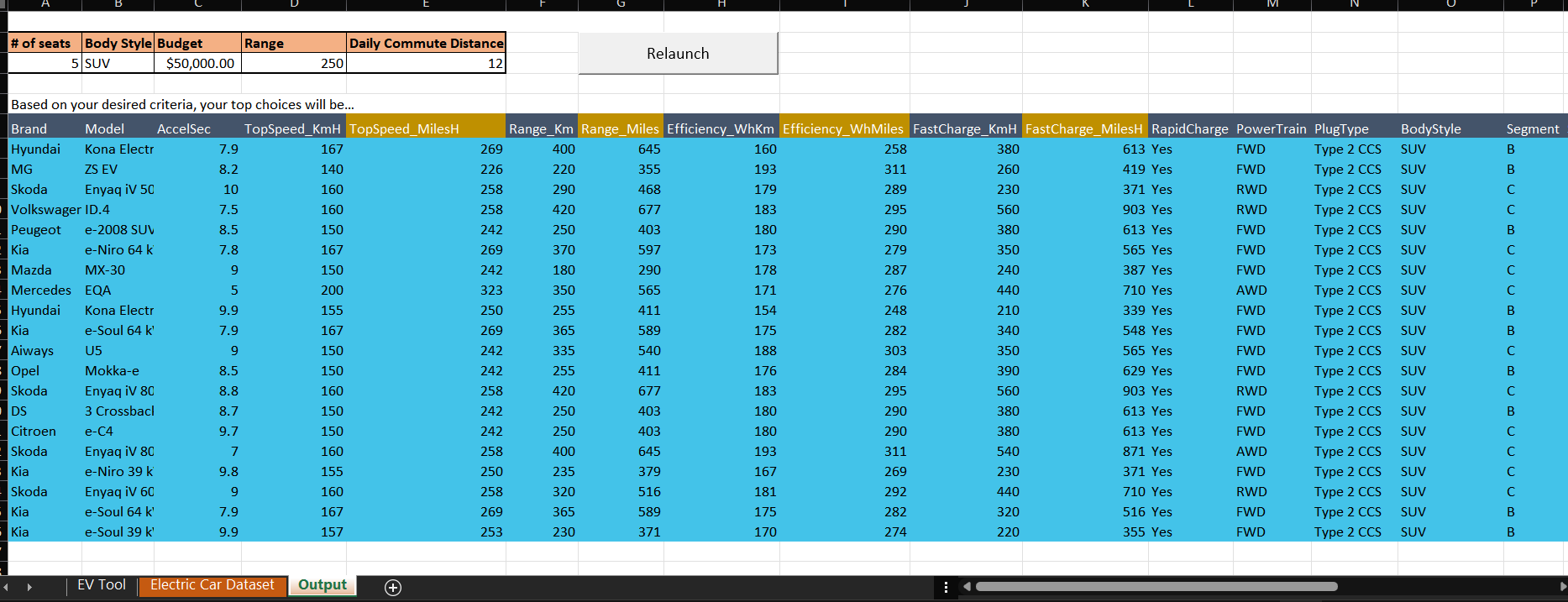


Figure Output tab shows the EV recommendations based on user input criteria

**Excel Techniques used:**

Since the model we developed was Primary VBA focused we utilized a lot of different coding techniques to implement our solution.

1. Each textbox on the user form has a **validation function** associated with it that checks the data the user has input. If the user input does not pass the validation check the function throws an error message prompting the user to correct the entered value
2. The main subroutine in our model is triggered when the user clicks the Submit button on the user form. Logic was added in this function to calculate additional metrics using **ranges** and **For..each..in loops** to traverse the data and populate the calculated columns
3. The main subroutine also calls the other subroutines in a sequential fashion using the **call** function
4. There is a column called formatting which contains a key formula that flags a row in the electric car data set if it meets the user criteria, this was written using **absolute referencing** and **IF AND functions**
5. The subroutine to format/highlight the rows matching user criteria also leverages **for loops** that run based on the **count** of rows in a range. Using the **interior color property,** we were able to change the background color for rows with matching user criteria
6. There is also an **additional validation check** here to show a error message if no cells are colored by the function
7. The subroutine to copy the highlighted rows utilizes **range object variables** and **offset property** to pass into a range and then identifies any colored rows using **interior color property** and **copy function** it copies its findings to a **different output worksheet**
8. Relaunch subroutine clears out the output sheet to allow users to start over, using the **delete** range feature. Additionally wiping out the row background color by setting **color property to None**
9. Note that code switches between the electric car data set sheet and the output sheet depending on what stage of the model it is in. We used **Activate function** and ranges to ensure the right sheet was referenced at different stages
10. There are also other buttons on the user form like clear, which resets the values on the form and cancel which closes out the user form

# Benefits to Target Audience

Our Electric Vehicle purchasing decision support system will give customers greater insights into the specifications which would lead to more informed decision making in the Electric vehicle purchasing process.

# Challenges

* One of the challenges from the start of the project was data relevancy. Based on the plug type and cost of EV in Euros we noticed that the EV data set was more focused on European EVs. There was no data on US EV’s in our analysis, we had to convert the existing data so that it would make sense to American customers.
* Another challenge we incurred was trying to use conditional formatting when copy and pasting the cells over to another worksheet. After some research we found out conditional formatting does not hold a color index within the cell so they cannot be copied based on a color index. Our workaround for this was to move away from conventional conditional formatting and instead use a formula to identify matching rows and then highlight them using interior color property.
* The offset function was another challenge for us, we had to use it multiple times throughout our code and kept having to refer to notes and articles to make sure we were using it correctly. We overcame this by using msgboxes to display cell addresses while writing VBA to utilize the offset function correctly.
* Switching over between worksheets was initially confusing, as we had to explicitly activate certain sheets in our code to ensure the code would run for the correct sheet
* *Open Issue we were unable to resolve:* The activate function used to make a sheet active throws a run time error the first time the application is run, after several searches and stack overflow quests we understand it is throwing an error due to us having renamed the workbook at one point. Note that the activate function references the codename property of the workbook (which is the same as the first saved name of the workbook) and renaming the workbook interferes with this setup thus generating a run time error.

# Personal Learning

Throughout this project we expanded our knowledge extensively on VBA. We were able to use and apply more advanced techniques than those we learned in class. Some of the VBA techniques that we were able to become proficient in were using ranges, the offset function, conditional formatting, user forms, for each next loop, and copying values from one worksheet to another.

With all the challenges that were listed above one way that we were able to debug our code was by using msgboxes. Adding these message boxes in certain spots of our code was helpful to make sure the code was doing what we intended it to do and that the correct variables and ranges were being selected and used.

# Closing Thoughts

Overall, this project was completed with great success. As a team we were able to apply our knowledge of Excel and VBA systems to a practical application. We took some of the concepts we learned in class and explored them further to create our Electric Vehicle purchasing decision support system. Through the course of our model building, we solved a lot of problems through trial and error and persistently worked through all the issues and bugs that we came across.