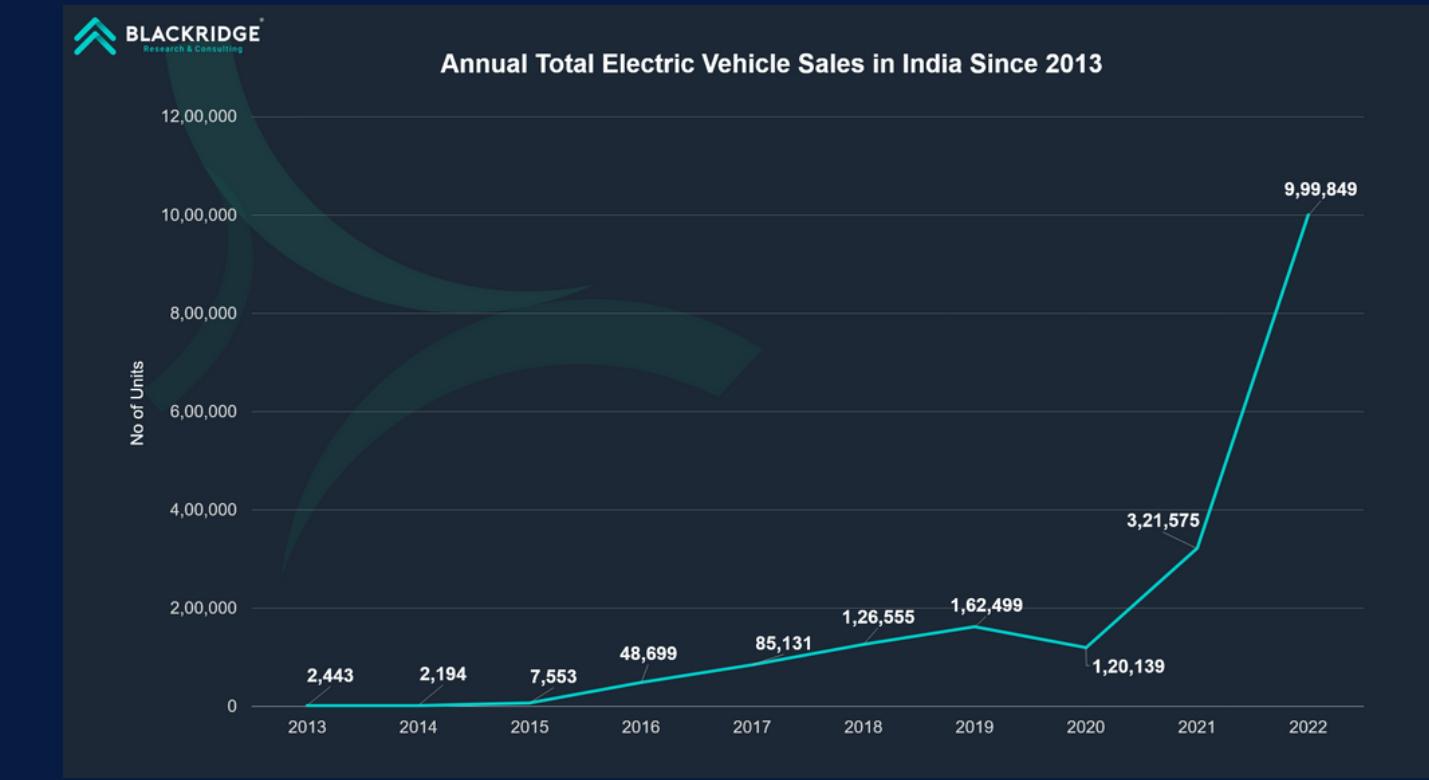


# ElectriGo.

Powering Your Electric Journey

# Introduction



## Electrifying India's Roads: The Surge of Electric Vehicles

Electric vehicles in India have surged in the last five years, but the sector still has a long way to go before reaching parity with ICE vehicles.

Though the Indian government's policies, rising fuel prices and greater awareness among the customers has given the nascent sector a much needed boost, EVs are still limited to affluent few or fleet owners.

# WHY?

There are various challenges in taking EVs to masses

**We want to bring confidence in using EVs in people, as their promoted use will greatly reduce the emissions by the vehicles which run on fuels, leading to a greener and more sustainable future.**

## Problem 1

### **Lack of sufficient charging infrastructure**

The lack of a robust and widespread charging network makes it inconvenient for EV owners, especially for those living in apartments or without dedicated parking spaces.

## Problem 3

### **Range anxiety, the fear of running out of electricity**

Although EV ranges have been improving, there is still a perception that EVs may not offer sufficient range for long-distance travel, particularly in a country with vast distances like India.

## Problem 2

### **Battery Technology and Supply Chain**

India currently relies heavily on imports for battery manufacturing, leading to supply chain challenges. The charging time of EVs is longer than the refuelling time of conventional vehicles, which affects their convenience & usability.

# What do we offer?



**Locating nearest public or private EV charging station which are fetched using API's provided by the government**

**Rigorous monitoring & maintaining the state of charging (soc), including battery health, replacement due date, etc. of EV using Machine Learning algorithm**

One stop platform for

**Battery Swap using Vehicle-2-Grid technology, powering bi-directional charging and hence EVs to manage renewable energy demand better & balance our energy system.**

(in case of emergencies)

# **What makes Electrigo unique?**

## **Unification of apps to provide a platform enforcing an entire Ecosystem**

**Offers feature to book  
the slot to charge EV in  
advance**

**Availability of both AC  
& DC batteries, hence  
making it ease**

**Use electric vehicles to  
manage renewable energy  
demand better and  
balance our energy system**

**Bilingual &  
Accessibility features**

**Integrated platform  
for the battery  
maintainace of EVs**

**Energy market parties  
can trade and optimise  
their balance**

# Utilizing AI in EVs for a Sustainable Future

## Our Machine Learning Models

### 1) Linear Regression

#### Linear Regression

```
#@title Linear Regression

# Separate features (X) and target (Y)
X = data.drop(columns=['SoC [%]'])
Y = data[['SoC [%]']]

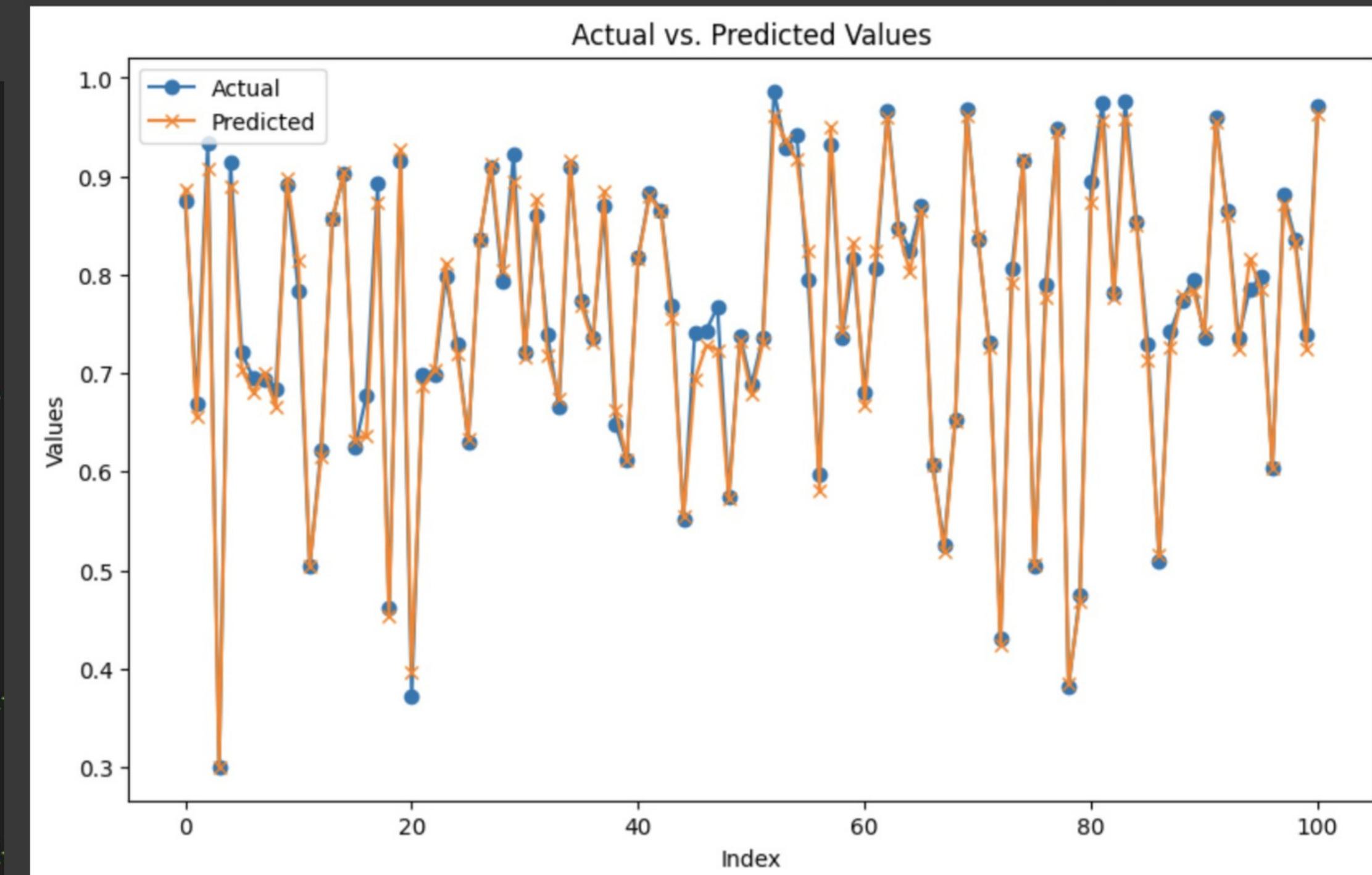
# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split

# Initializing the regression model
LR = LinearRegression()

# Training the model on the training data
LR.fit(X_train, y_train)

# Making predictions on the training and testing data
train_predictions = LR.predict(X_train)
test_predictions = LR.predict(X_test)

# Calculating RMSE for both training and testing data
train_rmse = mean_squared_error(y_train, train_predictions, squared=False)
test_rmse = mean_squared_error(y_test, test_predictions, squared=False)
```



## 2) Random Forest

### Random Forest Regressor

```
#@title Random Forest Regressor

import joblib
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error

data = pd.read_csv('/content/drive/MyDrive/Public/HackAI.csv')

X = data.drop(columns=['SoC [%]'])

Y = data[['SoC [%]']]
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

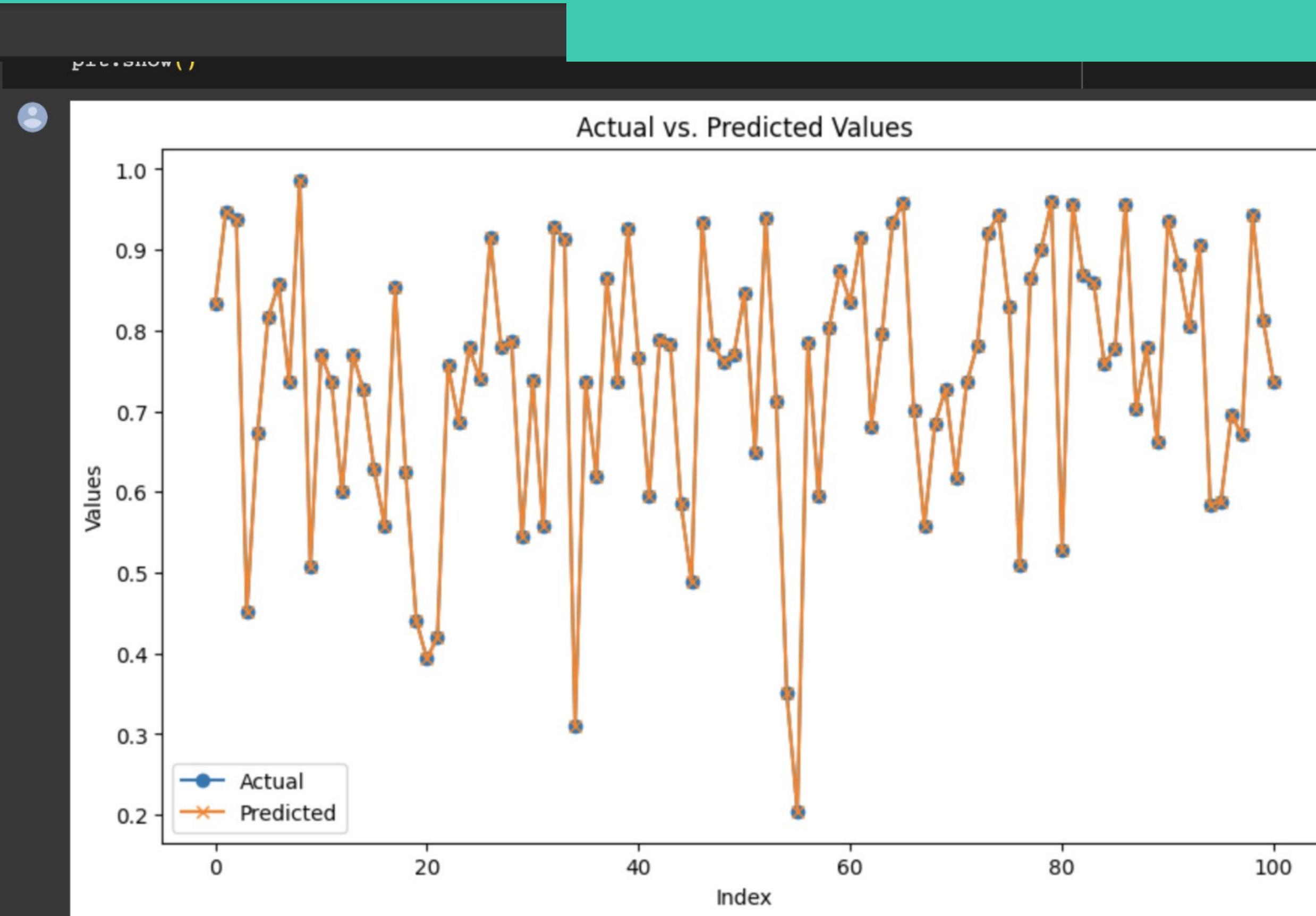
# Initializing the Random Forest Regressor
RF = RandomForestRegressor(n_estimators=100, random_state=42)

y_train = y_train.values.ravel()
# Now Training the model on the training data
RF.fit(X_train, y_train)

joblib.dump(RF, "/content/drive/MyDrive/Public/HackAI.pkl")

# Predicting on the test data
predictions = RF.predict(X_test)

# For performance measure calculating RMSE (Root Mean Square Error)
rmse = np.sqrt(mean_squared_error(y_test, predictions))
print(f'Root Mean Squared Error: {rmse:.3f}')
```



↳ Root Mean Squared Error: 0.001

# 3) Train- Test - Predict

```
# Calculating RMSE and R^2 for all three parts
train_rmse = mean_squared_error(y_train, train_predictions, squared=False)
test_rmse = mean_squared_error(y_test, test_predictions, squared=False)
pred_rmse = mean_squared_error(y_pred, pred_predictions, squared=False)

train_r2 = r2_score(y_train, train_predictions)
test_r2 = r2_score(y_test, test_predictions)
pred_r2 = r2_score(y_pred, pred_predictions)

# Print RMSE and R^2 values
print(f'Training RMSE: {train_rmse:.3f}, R^2: {train_r2:.3f}')
print(f'Testing RMSE: {test_rmse:.3f}, R^2: {test_r2:.3f}')
print(f'Prediction RMSE: {pred_rmse:.3f}, R^2: {pred_r2:.3f}')

# Plot actual vs. predicted values for testing data
plt.figure(figsize=(10, 6))
plt.plot(y_test, test_predictions, 'o')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values (Testing Data)')
plt.grid(True)
plt.show()

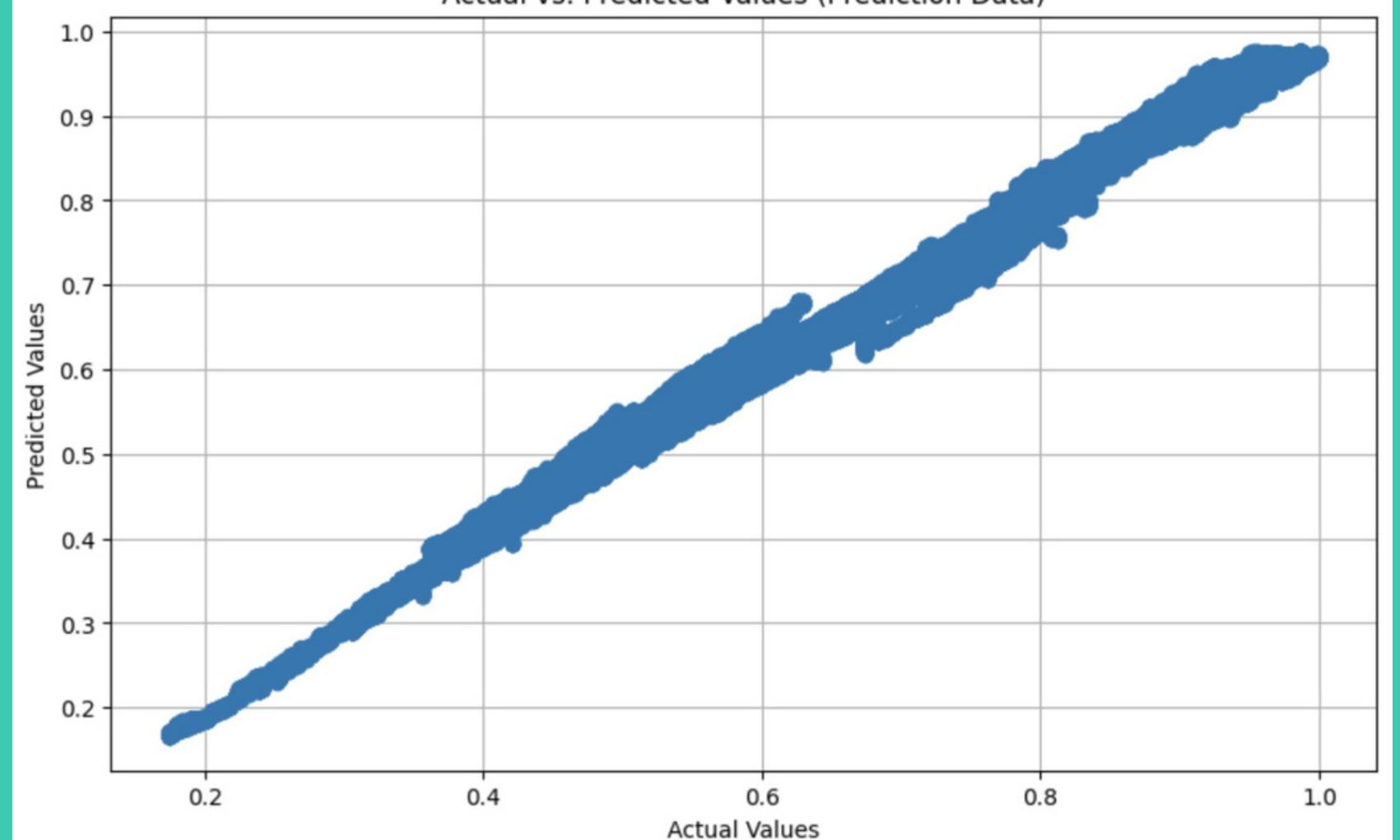
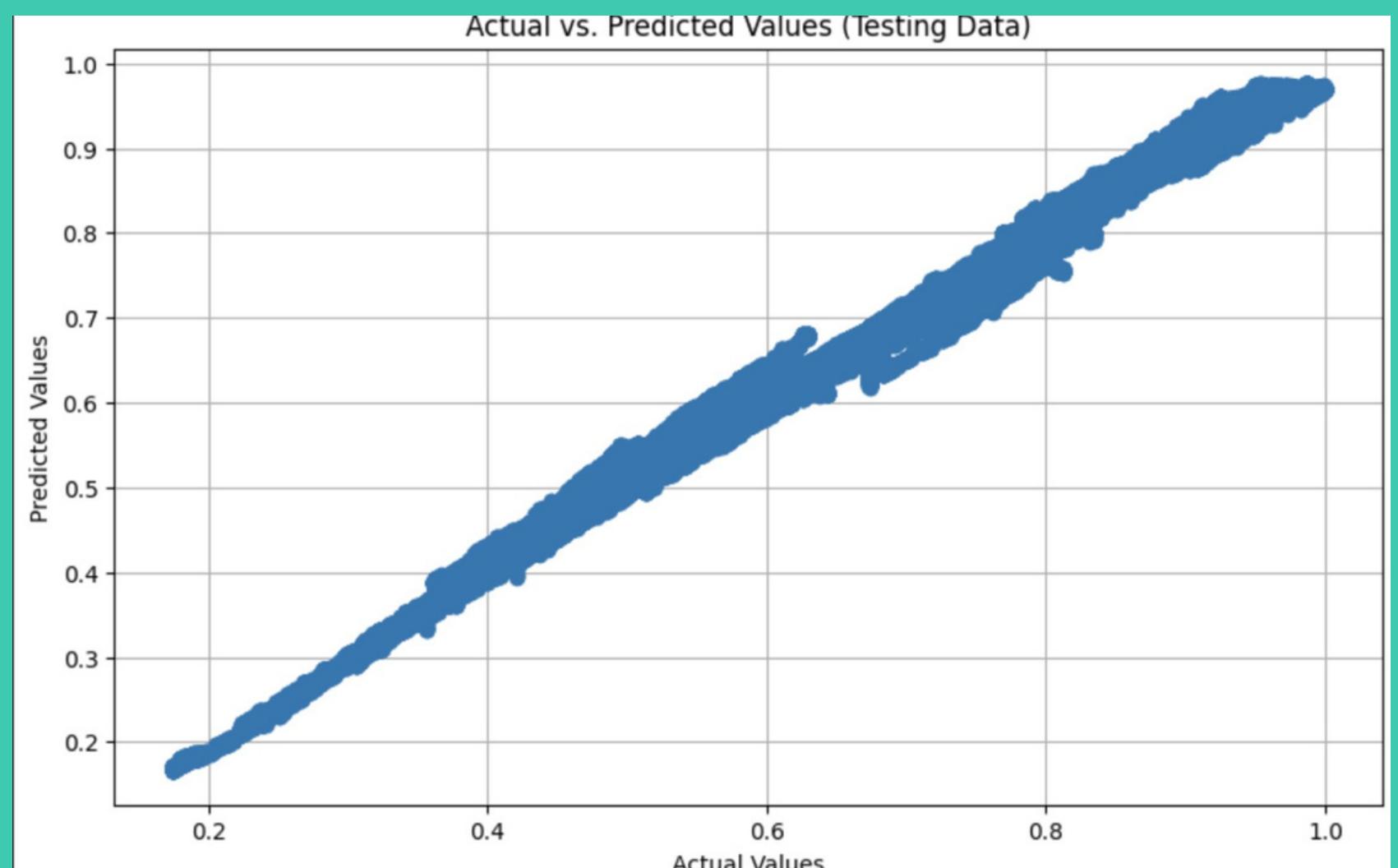
# Plot actual vs. predicted values for prediction data
plt.figure(figsize=(10, 6))
plt.plot(y_pred, pred_predictions, 'o')
plt.xlabel('Actual Values')
plt.ylabel('Predicted Values')
plt.title('Actual vs. Predicted Values (Prediction Data)')
plt.grid(True)
plt.show()

Training RMSE: 0.013, R^2: 0.994
Testing RMSE: 0.013, R^2: 0.994
Prediction RMSE: 0.013, R^2: 0.994
```

**Training RMSE - 0.013**

**Testing RMSE - 0.013**

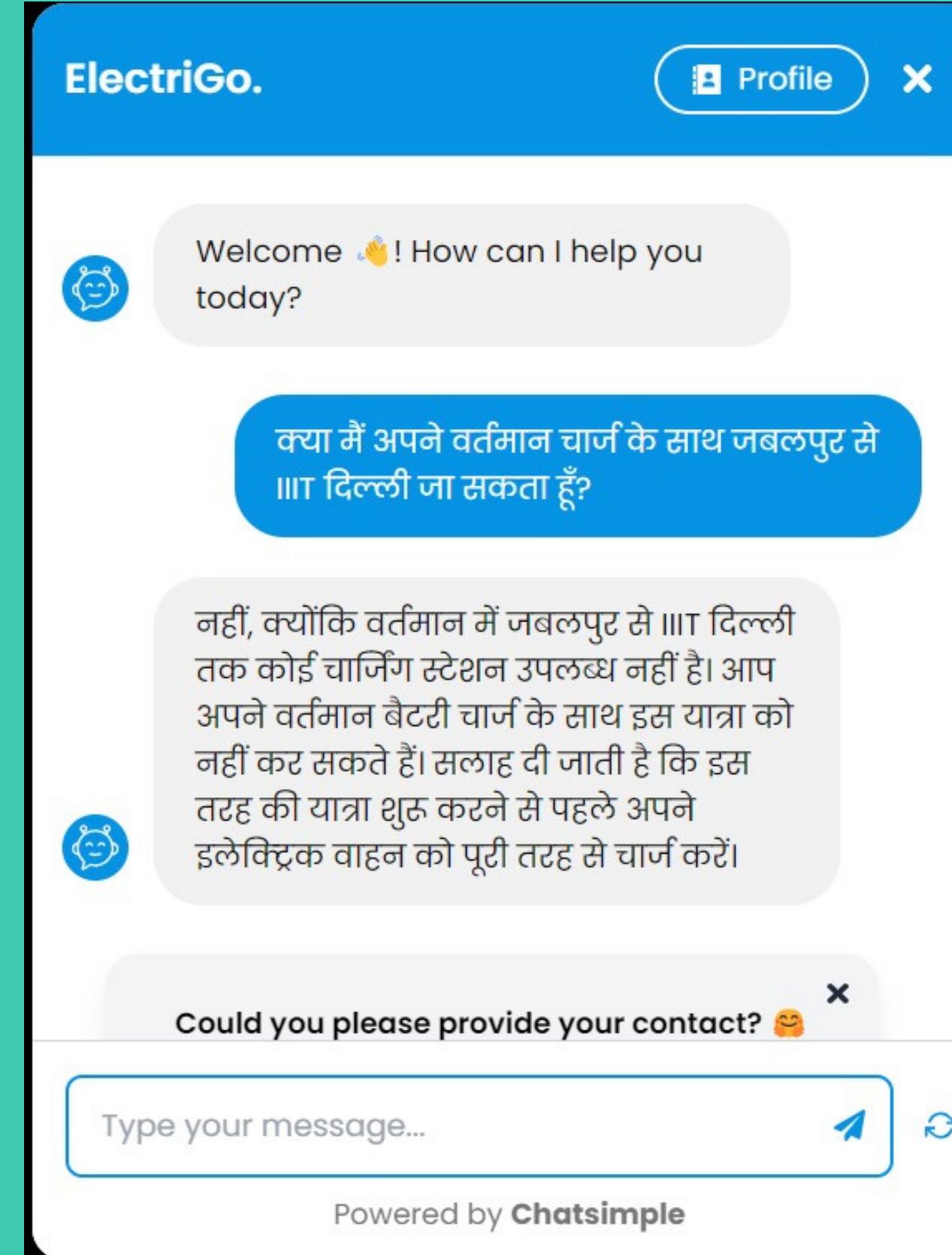
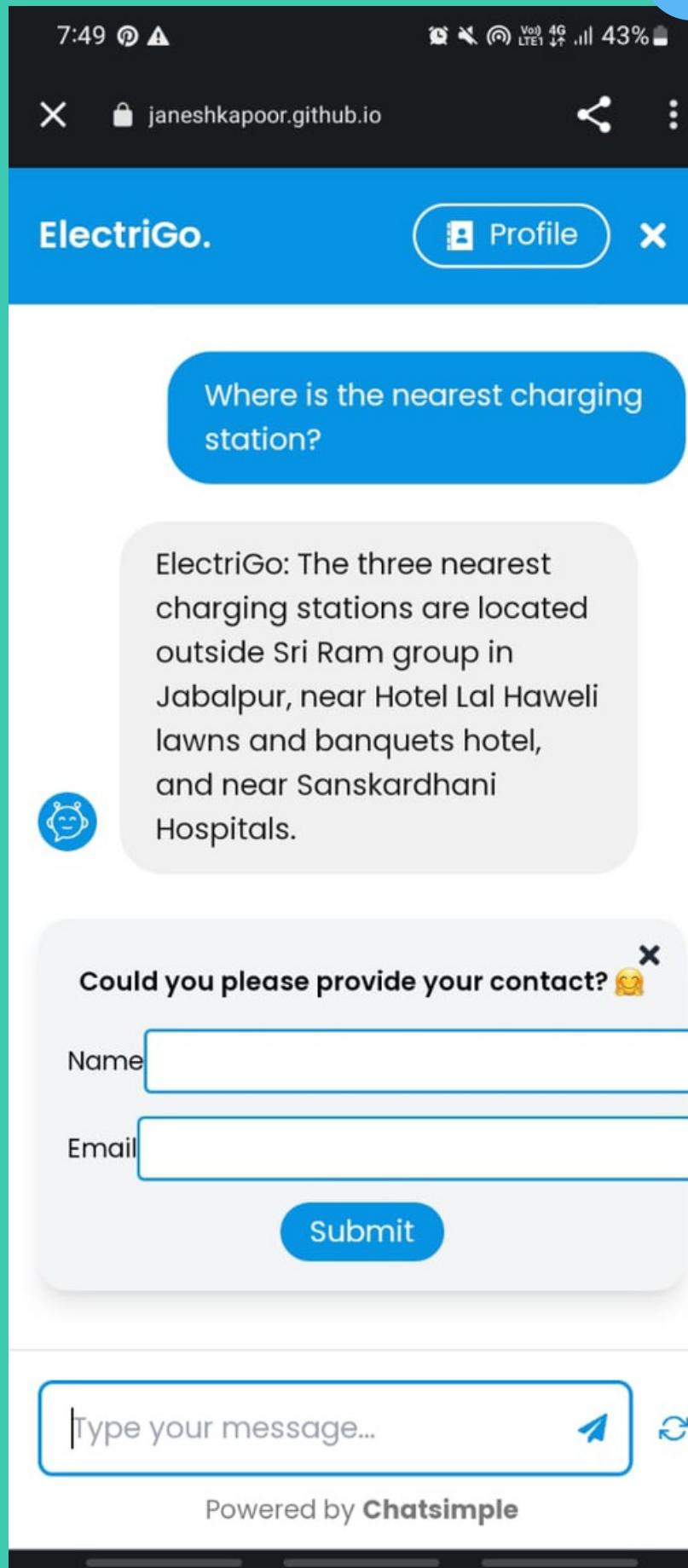
**Prediction RMSE - 0.013**





- 1.5 million+ EV rickshaw being already on the road.
- 15,000 EV rickshaw adding up every month.
- **While we were developing this solution we noticed that whenever we talk about EVs, what comes to our mind is the two or four wheelers owned by richer sections of the society. While we left the major portion of the EV industry in India and that is the EV rickshaw owners. Finally, we came up with the idea of developing a multilingual chatbot that provides information about the nearest charging stations, charging time etc to the rickshaw owners. The chatbot has a simple interface of just being used by numbers like 1 for locating the charging stations and so on, which provides better accessibility to these rickshaw owners.**

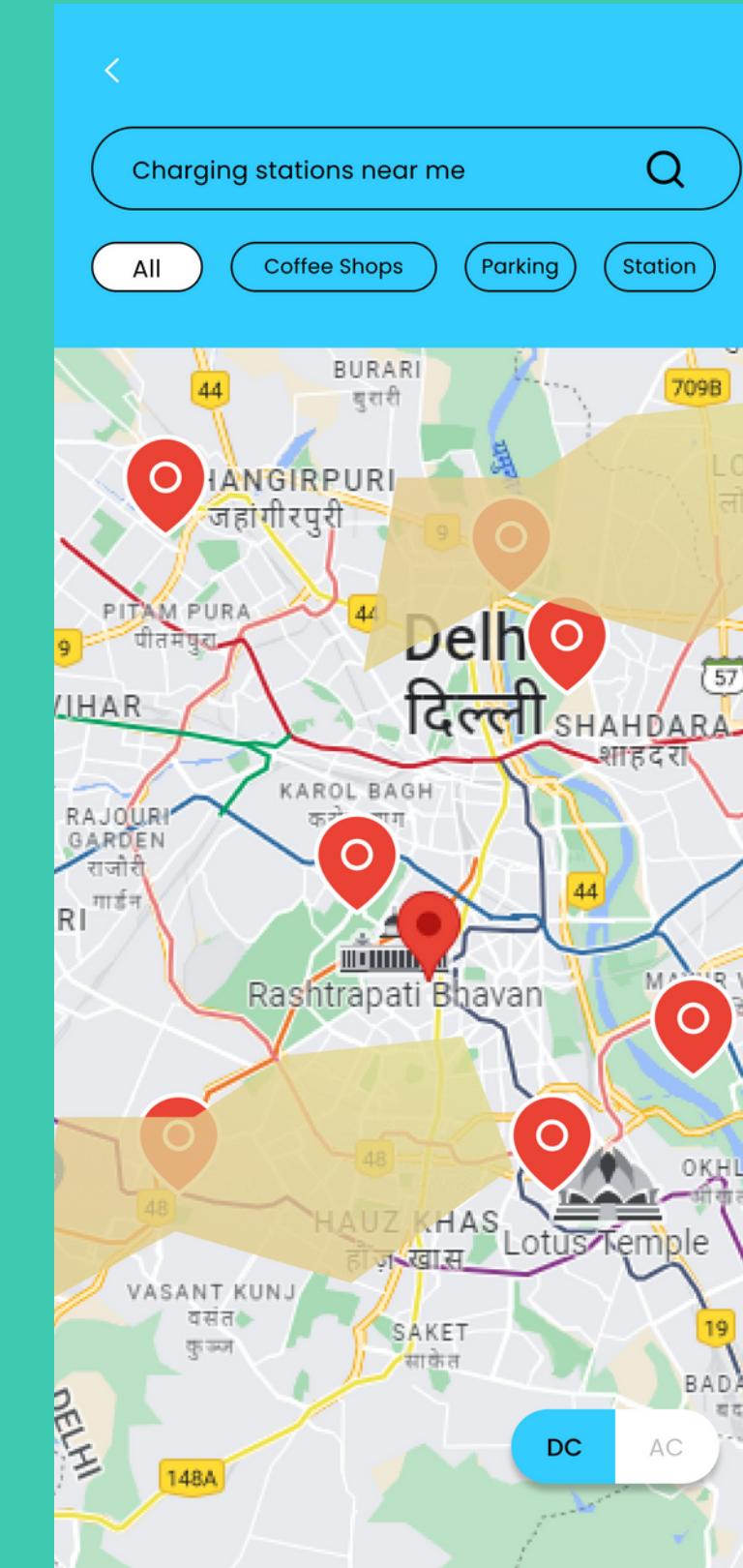
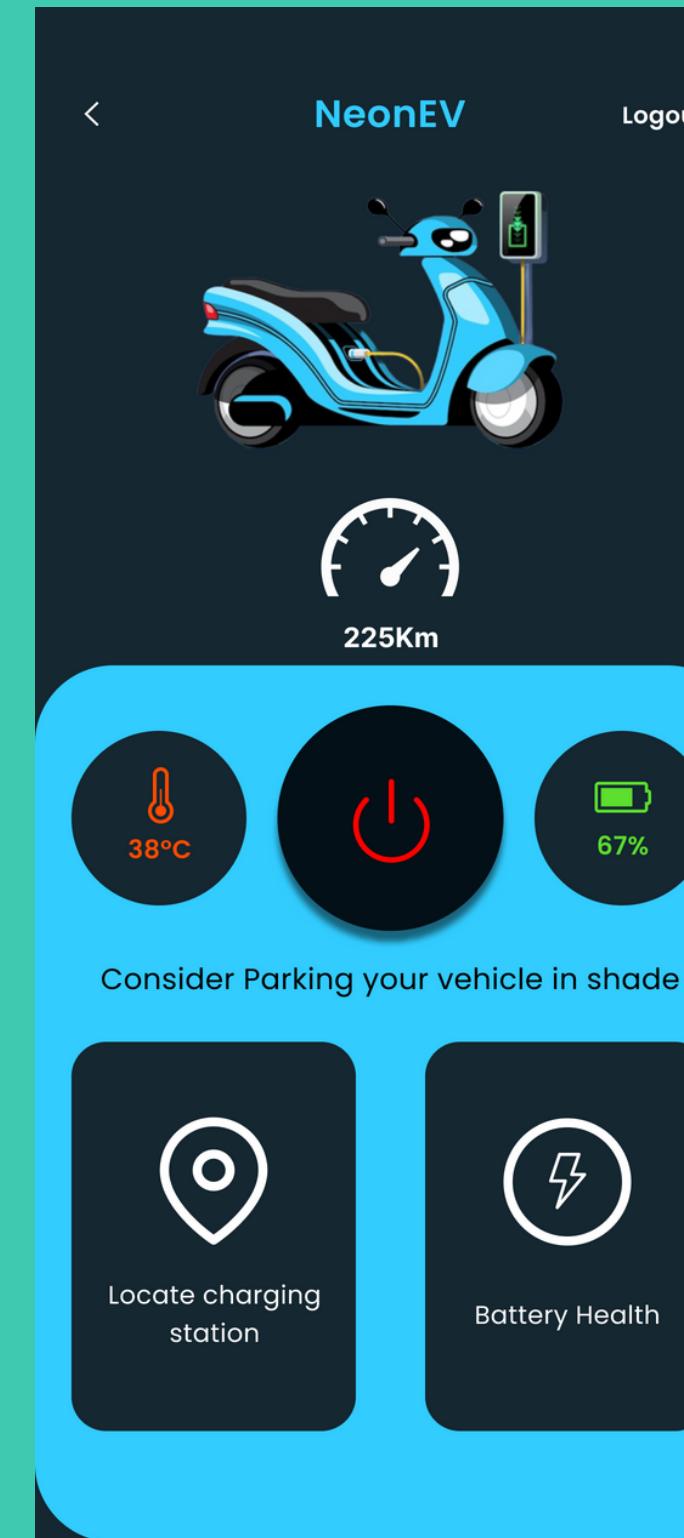
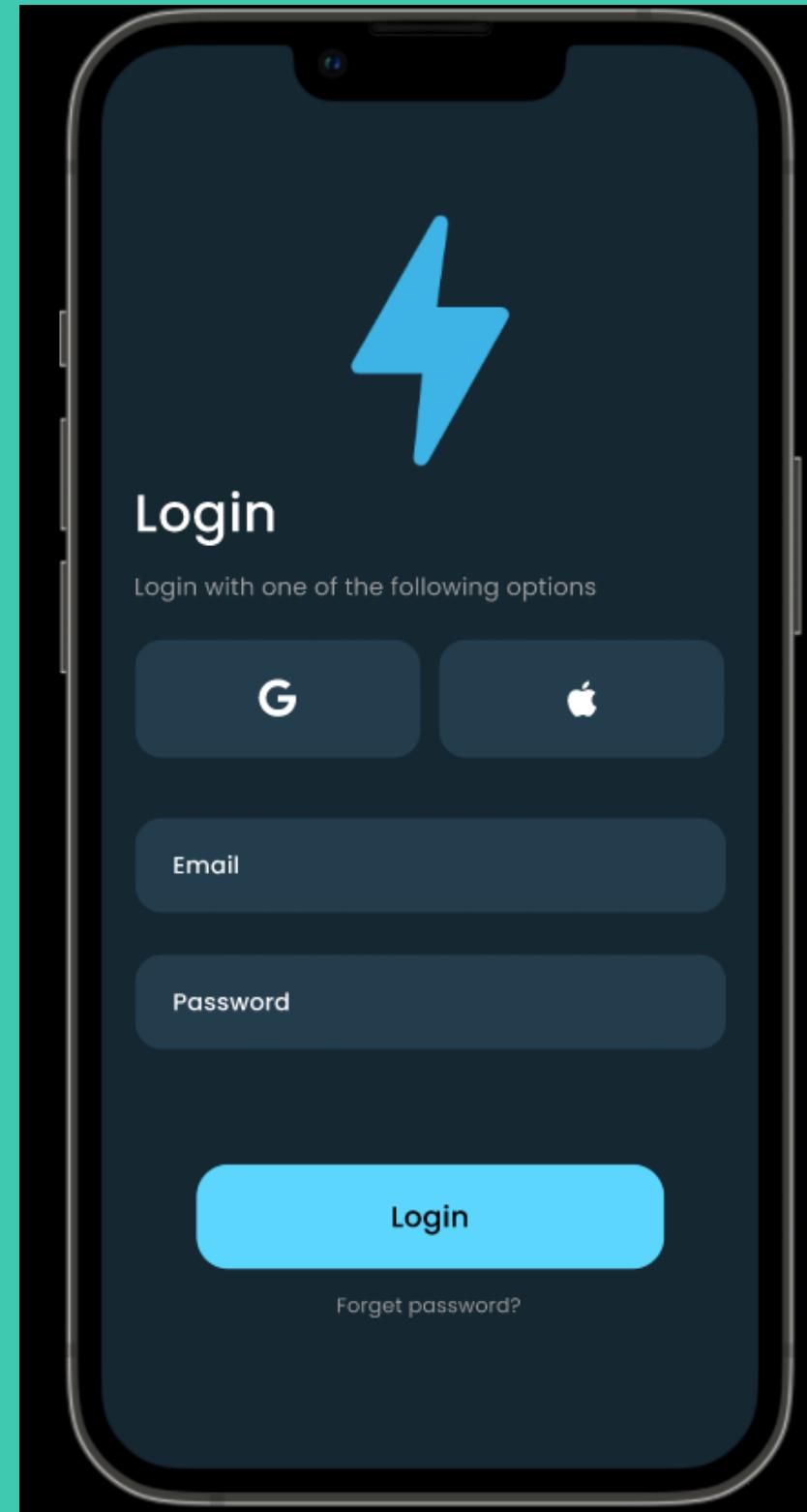
# AI-based Multilingual Chatbot



**When we talk about EVs  
E-Rickshaws are usually ignored, but  
as a matter of fact, they are the biggest  
consumer of EVs in India.**

-----  
**Our Chatbot makes their lives Easy :)**

# Application UI !!

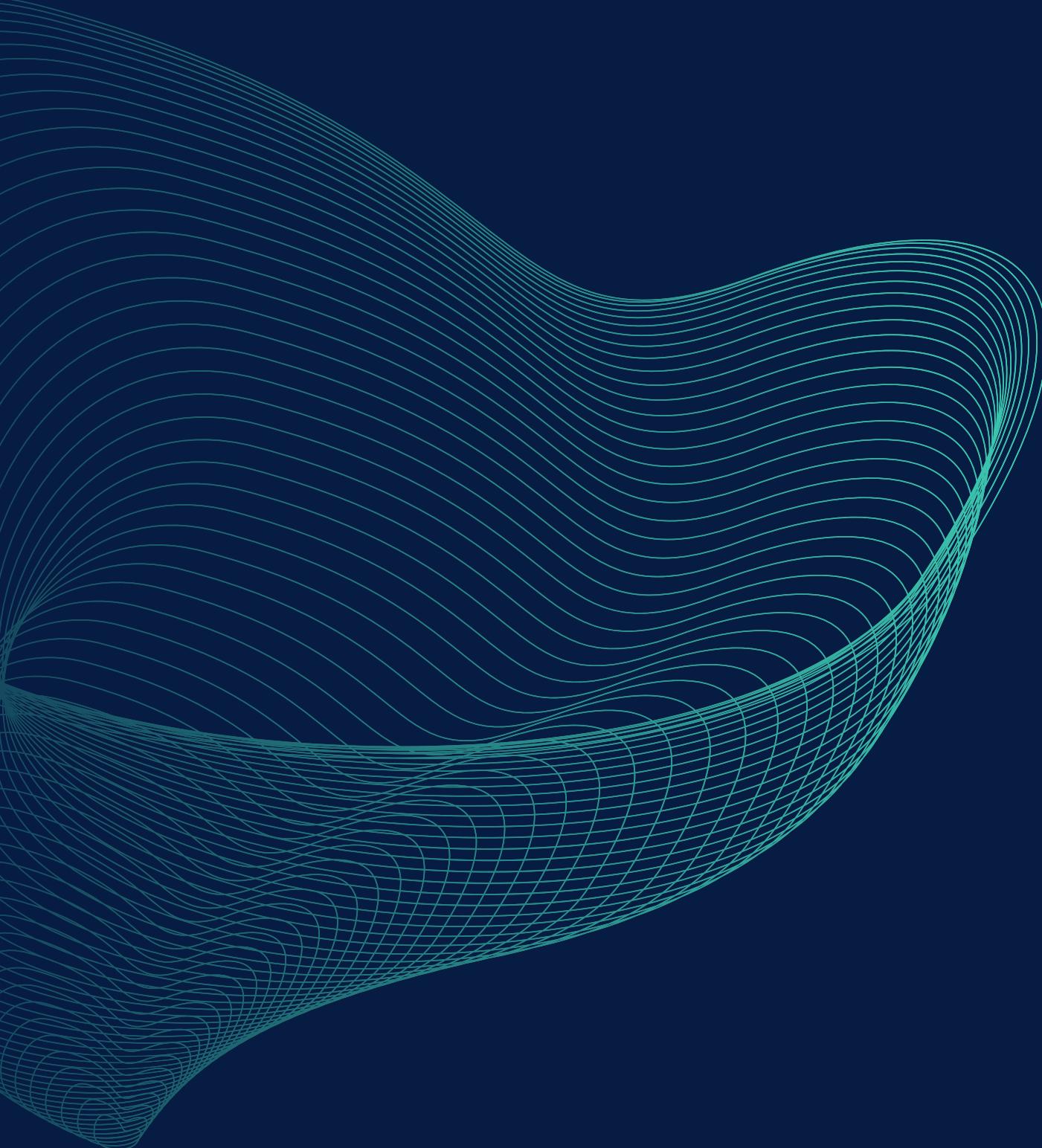


# OUR GOAL!

## A Greener Future

**Bringing confidence in using EVs in people, as their promoted use will greatly reduce the emissions of vehicles that run on fuels, leading to a greener and more sustainable future.**





# THANKING YOU

**Janesh Kappor**

**Divyam Khorwal**

**Tushar Chandra**

**Varsha Bhaskar**