# Project Report : Analysis of Ev vs Non-Ev vehicles

# Contents :

* Index
* Description
* Problem and Relevant information
* Purpose, outcome and benefits.
* Plan
* Design (Diagrams)
* Implemnetation
* Code & explanation
* O/P screenshots
* Closure
* Bibliography

# 1. Description

This project aims to analyze and compare electric vehicles (EVs) and internal combustion engine (ICE) vehicles (non-EVs) in India on a **state-wise** basis. With the rising importance of sustainable transportation, this analysis provides insights into the **growth trends**, **environmental impact**, and **infrastructure readiness** across Indian states.

The focus is on three major aspects:

* **EV Growth Rate (State-wise)**
* **EV vs Non-EV CO₂ Emissions (State-wise)**
* **EV Charging Infrastructure (State-wise Charging Stations)**

# 2.Problem Statement

With the increase in EV usage, it is necessary to understand:

* Which states are leading in EV adoption?
* What impact EVs have on emissions compared to ICE vehicles?
* Which regions need infrastructure upgrades like charging stations?

**Datasets Used**

| **Dataset** | **Description** |
| --- | --- |
| EV Registration Data | State-wise number of EVs |
| ICE Vehicle Data | State-wise number of non-EVs |
| Emissions Data | CO₂ emissions by EV and ICE vehicles |

# 3.Purpose

**1. Understanding India's Transition to EVs**

India is witnessing a gradual shift from conventional internal combustion engine (ICE) vehicles to electric vehicles (EVs) due to rising fuel prices, climate concerns, and government policies. This project helps **quantify** and **visualize** this transition on a **state-wise level**, giving clarity on the speed and effectiveness of adoption.

**2. Environmental Impact Assessment**

One of the key motivations behind promoting EVs is **reducing vehicular CO₂ emissions**. By comparing state-wise emissions from EVs and ICE vehicles, this project helps identify:

* How much emissions can be reduced if more EVs are adopted
* Which states contribute the most to vehicular pollution
* Regions where EV adoption can have the most **climate-positive impact**

**3. Evaluating Infrastructure Readiness**

Adopting EVs at scale requires strong charging infrastructure. This project analyzes the **availability of public charging stations** across states, identifying:

* Infrastructure gaps
* States that are well-prepared for EV adoption
* Regions where government or private investments are urgently needed

**4. Tracking Growth Patterns**

Analyzing the **growth rate of EV registrations** over time provides insights into:

* Effectiveness of EV policies and subsidies
* Behavioural change among consumers
* Emerging EV hubs (e.g., Delhi, Karnataka, Maharashtra)

This helps policymakers and companies understand **where the EV mark**

**Outcome**

* Trends and rankings for EV growth
* Emissions savings comparison
* Charging infrastructure gaps

**Benefits**

* Identify regions requiring more EV infrastructure
* Guide policymakers and private investors
* Help understand environmental improvements

# 4. Plan

| **Step** | **Task** |
| --- | --- |
| 1 | Collect datasets |
| 2 | Clean and preprocess data |
| 3 | Merge datasets |
| 4 | Calculate growth rates, emission comparisons |
| 5 | Create visualizations |
| 6 | Generate report and conclusions |

# 5. Design (Diagrams)

**A. Project Workflow Diagram**

[Data Collection]

↓

[Data Cleaning]

↓

[Feature Engineering → EV Growth, Emissions, Charging Infra]

↓

[Visualization → Bar charts, Pie Charts, Heatmaps]

↓

[Insights & Reporting]

# 6. Implementation

**Tools & Technologies:**

* **Python**
* **Libraries:** pandas, matplotlib, seaborn, plotly, geopandas
* **Platform:** Jupyter Notebook / Google Colab

# **7. Code with Explanation**

# 1.Growth Rate in ev

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Create DataFrame from provided data

data = {

    'State/UT': ['Uttar Pradesh', 'Tamil Nadu', 'Madhya Pradesh', 'Maharashtra',

                 'Karnataka', 'Bihar', 'Rajasthan', 'Mizoram'],

    'EV Units FY23‑24': ['~78,744', '~96,000', '~78,744', '–', '–', '–', '–', '–'],

    'EV Units FY24‑25': ['~107,258', '~131,482', '~107,258', '241,941',

                         '179,037', '112,854', '109,393', '–'],

    'Growth Rate (%)': ['36.2%', '35.9%', '36.2%', '– (volume only)',

                        '– (volume only)', '– (volume only)', '– (volume only)', '139% Y-o-Y (highest)']

}

df = pd.DataFrame(data)

# Clean numerical columns

def clean\_num(x):

    if isinstance(x, str):

        x = x.replace('~', '').replace(',', '').replace('%', '')

        if any(char.isdigit() for char in x):

            return float(''.join(filter(lambda y: y.isdigit() or y == '.', x)))

    return np.nan

df['EV Units FY23‑24'] = df['EV Units FY23‑24'].apply(clean\_num)

df['EV Units FY24‑25'] = df['EV Units FY24‑25'].apply(clean\_num)

# Extract growth rate numbers

df['Growth Rate'] = df['Growth Rate (%)'].str.extract(r'(\d+\.?\d\*)').astype(float)

plt.figure(figsize=(12, 6))

sns.barplot(

    data=df.sort\_values('EV Units FY24‑25', ascending=False).head(7),

    x='State/UT',

    y='EV Units FY24‑25',

    palette='viridis'

)

plt.title('Top States by EV Registrations (FY 2024-25)', fontsize=14)

plt.ylabel('Number of EVs (Thousands)')

plt.ylim(0, 260000)

plt.gca().yaxis.set\_major\_formatter(plt.FuncFormatter(lambda x, \_: f'{int(x/1000)}'))

plt.xticks(rotation=45)

plt.tight\_layout()

plt.savefig('top\_states\_volume.png', dpi=300)

plt.show()

# Filter states with available growth rates

growth\_df = df.dropna(subset=['Growth Rate']).sort\_values('Growth Rate', ascending=False)

plt.figure(figsize=(10, 6))

bars = plt.bar(

    growth\_df['State/UT'],

    growth\_df['Growth Rate'],

    color=['#1f77b4' if state != 'Mizoram' else '#ff7f0e' for state in growth\_df['State/UT']]

)

plt.title('Y-o-Y EV Growth Rate Comparison', fontsize=14)

plt.ylabel('Growth Rate (%)')

plt.xticks(rotation=15)

# Add value labels

for bar in bars:

    height = bar.get\_height()

    plt.text(bar.get\_x() + bar.get\_width()/2., height + 3,

             f'{height:.1f}%', ha='center', va='bottom')

plt.tight\_layout()

plt.savefig('growth\_rate\_comparison.png', dpi=300)

plt.show()

# Select states with both years' data

compare\_df = df[df['EV Units FY23‑24'].notna()].head(3).melt(

    id\_vars='State/UT',

    value\_vars=['EV Units FY23‑24', 'EV Units FY24‑25'],

    var\_name='Year',

    value\_name='Units'

)

plt.figure(figsize=(10, 6))

sns.barplot(

    data=compare\_df,

    x='State/UT',

    y='Units',

    hue='Year',

    palette='Blues'

)

plt.title('EV Registration Comparison: FY23-24 vs FY24-25', fontsize=14)

plt.ylabel('Number of EVs (Thousands)')

plt.gca().yaxis.set\_major\_formatter(plt.FuncFormatter(lambda x, \_: f'{int(x/1000)}'))

plt.xticks(rotation=15)

plt.legend(title='Financial Year')

plt.tight\_layout()

plt.savefig('yoy\_comparison.png', dpi=300)

plt.show()

plt.figure(figsize=(10, 6))

sns.scatterplot(

    data=df,

    x='EV Units FY24‑25',

    y='Growth Rate',

    hue='State/UT',

    s=200,

    palette='tab10'

)

plt.title('EV Volume vs Growth Rate (FY 2024-25)', fontsize=14)

plt.xlabel('EV Registrations (Thousands)')

plt.ylabel('Growth Rate (%)')

plt.xlim(0, 260000)

plt.gca().xaxis.set\_major\_formatter(plt.FuncFormatter(lambda x, \_: f'{int(x/1000)}'))

plt.grid(alpha=0.3)

# Add state labels

for i, row in df.iterrows():

    if not np.isnan(row['Growth Rate']):

        plt.annotate(row['State/UT'],

                    (row['EV Units FY24‑25'] + 5000,

                     row['Growth Rate'] + 2),

                    fontsize=9)

plt.tight\_layout()

plt.savefig('volume\_vs\_growth.png', dpi=300)

plt.show()

# 2.CO2 Emissions

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Create DataFrame from provided data

data = {

    'State/UT': ['Uttar Pradesh', 'Maharashtra', 'Karnataka', 'Tamil Nadu',

                 'Gujarat', 'Delhi', 'Rajasthan', 'Andhra Pradesh',

                 'West Bengal', 'Madhya Pradesh', 'India Total'],

    'EV Vehicles': [574967, 305006, 248747, 173152, 138410, 233212, 180670, 67905, 68376, 96151, 2086596],

    'ICE Vehicles': [43852548, 34323748, 29785247, 31618002, 22804558, 13994966, 18857703, 16553509, 15227219, 19616269, 246633769],

    'Total CO₂ by EVs (tonnes)': [287483.5, 152503, 124373.5, 86576, 69205, 116606, 90335, 33952.5, 34188, 48075.5, 1043298],

    'Total CO₂ by ICEs (tonnes)': [87705096, 68647496, 59570494, 63236004, 45609116, 27989932, 37715406, 33107018, 30454438, 39232538, 493267538]

}

df = pd.DataFrame(data)

# Remove India Total row for state-level analysis

df\_states = df[df['State/UT'] != 'India Total'].copy()

# Calculate total vehicles and total CO₂

df\_states['Total Vehicles'] = df\_states['EV Vehicles'] + df\_states['ICE Vehicles']

df\_states['Total CO₂ (tonnes)'] = df\_states['Total CO₂ by EVs (tonnes)'] + df\_states['Total CO₂ by ICEs (tonnes)']

# Calculate EV penetration percentage

df\_states['EV Penetration (%)'] = (df\_states['EV Vehicles'] / df\_states['Total Vehicles']) \* 100

# Calculate CO₂ savings potential

df\_states['CO₂ Savings Potential (tonnes)'] = df\_states['ICE Vehicles'] \* (2 - 0.5)  # If all ICE switched to EV

# India-level data

india\_row = df[df['State/UT'] == 'India Total'].iloc[0]

# Vehicle composition pie chart

plt.figure(figsize=(12, 6))

plt.subplot(1, 2, 1)

plt.pie(

    [india\_row['EV Vehicles'], india\_row['ICE Vehicles']],

    labels=['EV Vehicles', 'ICE Vehicles'],

    autopct='%1.1f%%',

    colors=['#4CAF50', '#F44336'],

    startangle=90

)

plt.title('India: Vehicle Composition', fontsize=14)

# CO₂ emissions composition pie chart

plt.subplot(1, 2, 2)

plt.pie(

    [india\_row['Total CO₂ by EVs (tonnes)'], india\_row['Total CO₂ by ICEs (tonnes)']],

    labels=['EV Emissions', 'ICE Emissions'],

    autopct='%1.1f%%',

    colors=['#66BB6A', '#EF5350'],

    startangle=90

)

plt.title('India: CO₂ Emissions Distribution', fontsize=14)

plt.tight\_layout()

plt.savefig('india\_emissions\_composition.png', dpi=300, bbox\_inches='tight')

plt.show()

# Select top 3 states by total CO₂ emissions

top\_states = df\_states.nlargest(3, 'Total CO₂ (tonnes)')['State/UT'].tolist()

# Create subplots

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

fig.suptitle('Vehicle Composition in Top 3 States by Emissions', fontsize=16)

for i, state in enumerate(top\_states):

    state\_data = df\_states[df\_states['State/UT'] == state].iloc[0]

    axes[i].pie(

        [state\_data['EV Vehicles'], state\_data['ICE Vehicles']],

        labels=['EV', 'ICE'],

        autopct='%1.1f%%',

        colors=['#4CAF50', '#F44336'],

        startangle=90

    )

    axes[i].set\_title(f"{state}\nTotal Vehicles: {state\_data['Total Vehicles']/1e6:.1f}M", fontsize=14)

plt.tight\_layout(rect=[0, 0, 1, 0.95])

plt.savefig('top\_states\_vehicle\_composition.png', dpi=300, bbox\_inches='tight')

plt.show()

# Prepare data for stacked bar chart

df\_states\_sorted = df\_states.sort\_values('Total CO₂ (tonnes)', ascending=False)

# Create stacked bar chart

plt.figure(figsize=(14, 8))

bar\_width = 0.8

# Plot ICE emissions (bottom)

ice\_bars = plt.bar(

    df\_states\_sorted['State/UT'],

    df\_states\_sorted['Total CO₂ by ICEs (tonnes)'],

    color='#EF5350',

    label='ICE Emissions',

    width=bar\_width

)

# Plot EV emissions on top of ICE

ev\_bars = plt.bar(

    df\_states\_sorted['State/UT'],

    df\_states\_sorted['Total CO₂ by EVs (tonnes)'],

    bottom=df\_states\_sorted['Total CO₂ by ICEs (tonnes)'],

    color='#66BB6A',

    label='EV Emissions',

    width=bar\_width

)

plt.title('State-wise CO₂ Emissions from Vehicles', fontsize=16)

plt.ylabel('CO₂ Emissions (Million Tonnes)')

plt.yscale('log')

plt.xticks(rotation=45, ha='right')

plt.legend()

plt.grid(axis='y', alpha=0.3)

# Format y-axis labels

plt.gca().yaxis.set\_major\_formatter(plt.FuncFormatter(lambda x, \_: f'{x/1e6:.0f}M'))

plt.tight\_layout()

plt.savefig('state\_emissions\_comparison.png', dpi=300, bbox\_inches='tight')

plt.show()

plt.figure(figsize=(12, 8))

scatter = sns.scatterplot(

    data=df\_states,

    x='EV Penetration (%)',

    y='Total CO₂ (tonnes)',

    size='Total Vehicles',

    sizes=(100, 1000),

    hue='State/UT',

    palette='tab10',

    legend='brief'

)

plt.title('EV Penetration vs Total CO₂ Emissions', fontsize=16)

plt.xlabel('EV Penetration (%)')

plt.ylabel('Total CO₂ Emissions (Million Tonnes)')

plt.ylim(0, 1e8)

plt.gca().yaxis.set\_major\_formatter(plt.FuncFormatter(lambda x, \_: f'{x/1e6:.0f}M'))

plt.grid(alpha=0.3)

# Add state labels

for i, row in df\_states.iterrows():

    plt.annotate(row['State/UT'],

                (row['EV Penetration (%)'] + 0.05,

                 row['Total CO₂ (tonnes)'] + 5e5),

                fontsize=9)

plt.tight\_layout()

plt.savefig('ev\_penetration\_vs\_emissions.png', dpi=300, bbox\_inches='tight')

plt.show()

# Calculate potential savings

df\_states['CO₂ Savings Potential (M tonnes)'] = df\_states['CO₂ Savings Potential (tonnes)'] / 1e6

df\_sorted\_savings = df\_states.sort\_values('CO₂ Savings Potential (M tonnes)', ascending=False)

# Create horizontal bar chart

plt.figure(figsize=(12, 8))

bars = plt.barh(

    df\_sorted\_savings['State/UT'],

    df\_sorted\_savings['CO₂ Savings Potential (M tonnes)'],

    color='#29B6F6'

)

plt.title('Potential CO₂ Savings from Full EV Transition', fontsize=16)

plt.xlabel('Potential CO₂ Savings (Million Tonnes)')

plt.ylabel('State/UT')

plt.xlim(0, 70)

# Add value labels

for bar in bars:

    width = bar.get\_width()

    plt.text(width + 1, bar.get\_y() + bar.get\_height()/2,

             f'{width:.1f}M',

             va='center', ha='left')

plt.tight\_layout()

plt.savefig('co2\_savings\_potential.png', dpi=300, bbox\_inches='tight')

plt.show()

# 3.Charging Stations State wise

**import pandas as pd**

import matplotlib.pyplot as plt

import seaborn as sns

# Set style

sns.set\_style("whitegrid")

plt.rcParams['figure.figsize'] = (12, 6)

# ====== 1. CHARGING STATIONS ANALYSIS ======

charging\_data = {

    'State/UT': ['Karnataka', 'Maharashtra', 'Uttar Pradesh', 'Delhi', 'Tamil Nadu',

                'Kerala', 'Rajasthan', 'Gujarat', 'Haryana', 'Madhya Pradesh', 'Chhattisgarh'],

    'Public Charging Stations (PCS)': [5765, 3728, 1989, 1941, 1413, 958, 500, 476, 377, 341, 271]

}

charging\_df = pd.DataFrame(charging\_data).sort\_values('Public Charging Stations (PCS)', ascending=False)

# Plot charging stations distribution

plt.figure(figsize=(14, 7))

ax = sns.barplot(x='State/UT', y='Public Charging Stations (PCS)', data=charging\_df, palette='viridis')

plt.title('Public Charging Stations Distribution by State', fontsize=16)

plt.xticks(rotation=45, ha='right')

plt.ylabel('Number of Stations')

plt.xlabel('State/UT')

# Add value labels

for p in ax.patches:

    ax.annotate(f'{int(p.get\_height()):,}',

                (p.get\_x() + p.get\_width() / 2., p.get\_height()),

                ha='center', va='center', xytext=(0, 10), textcoords='offset points')

plt.tight\_layout()

plt.show()

# 4.Vehicles type emissions

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# Create DataFrame from provided data

data = {

    'Vehicle Type': ['Standard Gasoline Vehicle', 'Non-ev Vehicle', 'Ev Vehicle'],

    'Lifecycle Emissions': [24, 21, 19],

    'Production Emissions': [5.6, 6.5, 8.8],

    'Production Proportion': [23, 31, 46]

}

df = pd.DataFrame(data)

# Calculate use phase emissions

df['Use Phase Emissions'] = df['Lifecycle Emissions'] - df['Production Emissions']

# Calculate emission savings compared to gasoline vehicle

df['Emission Savings'] = df['Lifecycle Emissions'].iloc[0] - df['Lifecycle Emissions']

# Calculate percentage breakdown

df['Production %'] = (df['Production Emissions'] / df['Lifecycle Emissions']) \* 100

df['Use Phase %'] = (df['Use Phase Emissions'] / df['Lifecycle Emissions']) \* 100

plt.figure(figsize=(10, 6))

barplot = sns.barplot(

    data=df,

    x='Vehicle Type',

    y='Lifecycle Emissions',

    palette=['#E74C3C', '#F39C12', '#2ECC71']

)

plt.title('Total Lifecycle CO₂ Emissions by Vehicle Type', fontsize=15, pad=15)

plt.ylabel('Tonnes CO₂e')

plt.xlabel('')

plt.ylim(0, 26)

# Add value labels

for p in barplot.patches:

    barplot.annotate(

        f'{p.get\_height():.1f}',

        (p.get\_x() + p.get\_width()/2., p.get\_height()),

        ha='center', va='center', xytext=(0, 7), textcoords='offset points'

    )

# Add emission savings annotation

plt.text(0.5, 24.5, 'Baseline', ha='center', fontsize=9)

plt.text(1.5, 21.5, '3.0 tonnes saved', ha='center', fontsize=9, color='green')

plt.text(2.5, 19.5, '5.0 tonnes saved', ha='center', fontsize=9, color='green')

plt.tight\_layout()

plt.savefig('lifecycle\_emissions.png', dpi=300)

plt.show()

plt.figure(figsize=(10, 6))

# Plot use phase emissions (bottom)

use\_bars = plt.bar(

    df['Vehicle Type'],

    df['Use Phase Emissions'],

    color=['#3498DB', '#3498DB', '#3498DB'],

    label='Use Phase'

)

# Plot production emissions on top

prod\_bars = plt.bar(

    df['Vehicle Type'],

    df['Production Emissions'],

    bottom=df['Use Phase Emissions'],

    color=['#E67E22', '#E67E22', '#E67E22'],

    label='Production'

)

plt.title('Emissions Breakdown by Lifecycle Phase', fontsize=15, pad=15)

plt.ylabel('Tonnes CO₂e')

plt.legend(loc='upper right')

# Add percentage labels

for i, vehicle in enumerate(df['Vehicle Type']):

    total = df.loc[i, 'Lifecycle Emissions']

    prod = df.loc[i, 'Production Emissions']

    use = df.loc[i, 'Use Phase Emissions']

    plt.text(i, use/2, f'Use: {use:.1f}t\n({df.loc[i, "Use Phase %"]:.0f}%)',

             ha='center', va='center', color='white', fontsize=9)

    plt.text(i, use + prod/2, f'Prod: {prod:.1f}t\n({df.loc[i, "Production %"]:.0f}%)',

             ha='center', va='center', color='white', fontsize=9)

plt.tight\_layout()

plt.savefig('emissions\_breakdown.png', dpi=300)

plt.show()

plt.figure(figsize=(10, 6))

barplot = sns.barplot(

    data=df,

    x='Vehicle Type',

    y='Production Emissions',

    palette=['#E74C3C', '#F39C12', '#2ECC71']

)

plt.title('Manufacturing Emissions Comparison', fontsize=15, pad=15)

plt.ylabel('Tonnes CO₂e')

plt.xlabel('')

plt.ylim(0, 10)

# Add value labels and change annotations

for i, p in enumerate(barplot.patches):

    height = p.get\_height()

    barplot.annotate(

        f'{height:.1f}',

        (p.get\_x() + p.get\_width()/2., height),

        ha='center', va='center', xytext=(0, 7), textcoords='offset points'

    )

    # Add comparative annotations

    if i == 0:

        barplot.annotate(

            'Baseline',

            (p.get\_x() + p.get\_width()/2., height),

            ha='center', va='center', xytext=(0, -20), textcoords='offset points'

        )

    else:

        diff = height - df.loc[0, 'Production Emissions']

        barplot.annotate(

            f'+{diff:.1f}t',

            (p.get\_x() + p.get\_width()/2., height),

            ha='center', va='center', xytext=(0, -20), textcoords='offset points',

            color='red' if diff > 0 else 'green'

        )

plt.tight\_layout()

plt.savefig('production\_emissions.png', dpi=300)

plt.show()

fig, axes = plt.subplots(1, 3, figsize=(18, 6))

fig.suptitle('Emissions Distribution Across Lifecycle Phases', fontsize=16)

colors = ['#E67E22', '#3498DB']  # Production, Use Phase

for i, row in df.iterrows():

    sizes = [row['Production Emissions'], row['Use Phase Emissions']]

    labels = ['Production', 'Use Phase']

    axes[i].pie(

        sizes,

        labels=labels,

        autopct='%1.1f%%',

        startangle=90,

        colors=colors,

        wedgeprops={'edgecolor': 'w', 'linewidth': 2}

    )

    axes[i].set\_title(f"{row['Vehicle Type']}\nTotal: {row['Lifecycle Emissions']}t CO₂e", fontsize=13)

plt.tight\_layout(rect=[0, 0, 1, 0.95])

plt.savefig('emissions\_proportion.png', dpi=300)

plt.show()

# 8.output screenshot

# 

# 9. Closure

This data analysis project explored the transition to electric vehicles in India by comparing EV and ICE vehicles on key metrics. It identified:

* Fast-growing states in EV adoption (e.g., Delhi, Maharashtra, Karnataka)
* Areas with low emissions due to EV adoption
* States needing charging infrastructure investments

**Conclusion:**  
India is progressing toward EVs, but the transition is uneven. This project helps visualize disparities and opportunities in the EV ecosystem.

# 10. Bibliography

* Ministry of Road Transport & Highways (https://morth.nic.in)
* NITI Aayog EV Reports
* Central Electricity Authority (https://cea.nic.in
* EV-Ready India 2030 – CEEW
* State-wise EV Policy Documents
* Open Government Data Platform (<https://data.gov.in>)