**Project Report**

**Comprehensive Analysis of Petrol vs Electric Scooters in India**

**Index**

1. *Description*
2. *Plan*
3. *Design (Diagrams)*
4. *Implementation*
5. *Code Explanation*
6. *Output Screenshots*
7. *Closure*
8. *Bibliography*

**Description**

**Problem Statement**

*The transition from petrol-powered scooters to electric scooters is gaining momentum in India due to environmental concerns, government incentives, and technological advancements. However, consumers and stakeholders need a comprehensive comparison of petrol and electric scooters to make informed decisions***.**

**Objective**

*This project aims to analyze and compare petrol and electric scooters across multiple dimensions:*

* *Market trends and growth projections*
* *Total cost of ownership*
* *Environmental impact*
* *Performance and technology*
* *Infrastructure and ecosystem*
* *Consumer behavior and adoption*
* *Manufacturer ecosystem*
* *Future outlook*

**Data Sources**

*The analysis uses four Excel files:*

1. *scooter\_sales\_data.xlsx – Petrol scooter sales data.*
2. *India\_Electric\_2Wheeler\_Sales.xlsx – Electric scooter sales data.*
3. *EV\_Battery\_Analysis.xlsx – Battery cost and lifespan data.*
4. *Top\_25\_E2W\_Manufacturers\_FY2025.xlsx – Top electric scooter manufacturers.*

**Outcome & Benefits**

* *Helps consumers choose between petrol and electric scooters based on cost, performance, and environmental impact.*
* *Assists manufacturers in understanding market trends and consumer preferences.*
* *Provides policymakers insights into infrastructure needs and incentives.*

**Plan**

*The analysis is structured into nine sections:*

1. ***Market Overview & Trends****– Sales, market share, growth rates, and projections.*
2. ***Total Cost of Ownership****– Fuel, maintenance, insurance, and battery costs.*
3. ***Environmental Impact****– CO2 emissions, energy consumption, and noise pollution.*
4. ***Performance & Technology****– Speed, acceleration, efficiency, and smart features.*
5. ***Infrastructure & Ecosystem****– Charging stations, service centers, and government incentives.*
6. ***Consumer Behaviour & Adoption****– Preference factors and adoption barriers.*
7. ***Manufacturer Ecosystem****– Market concentration and top players.*
8. ***Future Outlook & Predictions****– Technology roadmap and market evolution.*
9. ***Executive Summary****– Key findings and recommendations.*

**Implementation**

***Tools & Libraries Used***

* ***Python****(Pandas, Matplotlib, Seaborn, NumPy)*
* ***Jupyter Notebook / IDE****for execution.*

***Steps***

1. ***Data Loading & Validation***
   * *Check for missing files.*
   * *Read Excel files into DataFrames.*
2. ***Market Analysis***
   * *Calculate total sales, market share, and growth rates.*
   * *Plot sales trends and projections.*
3. ***Cost Analysis***
   * *Compute fuel, maintenance, and battery costs over 10 years.*
   * *Visualize cumulative costs.*
4. ***Environmental Impact***
   * *Compare CO2 emissions, energy use, and noise levels.*
5. ***Performance & Tech***
   * *Compare speed, acceleration, and smart features.*
6. ***Infrastructure & Policy***
   * *Analyze charging stations and government incentives.*
7. ***Consumer & Manufacturer Insights***
   * *Evaluate adoption barriers and market concentration.*
8. ***Future Predictions***
   * *Forecast market share and technology advancements.*
9. ***Executive Summary***
   * *Summarize key insights and recommendations*

*CODE*

*import pandas as pd*

*import matplotlib.pyplot as plt*

*import seaborn as sns*

*import os*

*import numpy as np*

*from datetime import datetime*

*import warnings*

*warnings.filterwarnings('ignore')*

*# Enhanced styling for plots*

*plt.style.use('default')*

*sns.set\_palette("husl")*

*plt.rcParams['figure.figsize'] = [14, 8]*

*plt.rcParams['font.size'] = 11*

*plt.rcParams['axes.titlesize'] = 14*

*plt.rcParams['axes.labelsize'] = 12*

*plt.rcParams['xtick.labelsize'] = 10*

*plt.rcParams['ytick.labelsize'] = 10*

*def analyze\_petrol\_vs\_electric\_enhanced():*

*# Get current folder path*

*current\_dir = os.path.dirname(os.path.abspath(\_\_file\_\_)) if '\_\_file\_\_' in globals() else os.getcwd()*

*# Verifying files*

*required\_files = [*

*'scooter\_sales\_data.xlsx',*

*'India\_Electric\_2Wheeler\_Sales.xlsx',*

*'EV\_Battery\_Analysis.xlsx',*

*'Top\_25\_E2W\_Manufacturers\_FY2025.xlsx'*

*]*

*missing\_files = [f for f in required\_files if not os.path.exists(os.path.join(current\_dir, f))]*

*if missing\_files:*

*print("Missing files:", missing\_files)*

*return*

*print("\_"\*80)*

*print("COMPREHENSIVE ELECTRIC VS PETROL SCOOTER ANALYSIS")*

*print("\_"\*80)*

*print("Starting enhanced analysis...\n")*

*# Helper function to read Excel files*

*def read\_excel\_safe(filename, sheet\_name=None):*

*try:*

*df = pd.read\_excel(os.path.join(current\_dir, filename), sheet\_name=sheet\_name)*

*return df.dropna(how='all')*

*except Exception as e:*

*print(f"Error reading {filename}: {str(e)}")*

*return None*

*# Helper function for consistent formatting*

*def format\_number(num):*

*if num >= 1e6:*

*return f"{num/1e6:.2f}M"*

*elif num >= 1e3:*

*return f"{num/1e3:.1f}K"*

*else:*

*return f"{num:.0f}"*

*try:*

*# Load all data*

*petrol\_df = read\_excel\_safe('scooter\_sales\_data.xlsx', 'Sales Data')*

*electric\_sales = read\_excel\_safe('India\_Electric\_2Wheeler\_Sales.xlsx', 'Monthly Sales')*

*battery\_df = read\_excel\_safe('EV\_Battery\_Analysis.xlsx', 'Battery Data')*

*electric\_mfg = read\_excel\_safe('Top\_25\_E2W\_Manufacturers\_FY2025.xlsx', 'Top 25 Manufacturers')*

*if any(df is None for df in [petrol\_df, electric\_sales, battery\_df, electric\_mfg]):*

*raise ValueError("Could not load all required data files")*

*# 1. MARKET OVERVIEW & TRENDS*

*print("\n" + "\_"\*60)*

*print("1. MARKET OVERVIEW & TRENDS")*

*print("\_"\*60)*

*# Get sales data*

*total\_petrol = petrol\_df.loc[petrol\_df['Manufacturer'] == 'Total 2W sales', 'FY2025'].values[0]*

*total\_electric = electric\_sales.iloc[-2]['FY2025']*

*# Calculate market shares*

*total\_sales = total\_petrol + total\_electric*

*petrol\_share = (total\_petrol/total\_sales)\*100*

*electric\_share = (total\_electric/total\_sales)\*100*

*# Growth rate*

*petrol\_growth = float(str(petrol\_df.loc[petrol\_df['Manufacturer'] == 'Total 2W sales', 'YoY % change'].values[0]).strip('%'))*

*electric\_growth = float(str(electric\_sales.iloc[-2]['% Change']).strip('%')) if pd.notna(electric\_sales.iloc[-2]['% Change']) else 0*

*# overview of market*

*market\_overview = pd.DataFrame({*

*'Metric': ['Total Sales (Units)', 'Market Share (%)', 'YoY Growth (%)', 'CAGR Projection (%)'],*

*'Petrol Scooters': [format\_number(total\_petrol), f"{petrol\_share:.1f}%", f"{petrol\_growth:.1f}%", "5-8%"],*

*'Electric Scooters': [format\_number(total\_electric), f"{electric\_share:.1f}%", f"{electric\_growth:.1f}%", "25-35%"]*

*})*

*print("\nMarket Overview:")*

*print(market\_overview.to\_string(index=False))*

*fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(14, 10))*

*# pie chart of market share*

*ax1.pie([petrol\_share, electric\_share], labels=['Petrol', 'Electric'],*

*autopct='%1.1f%%', startangle=90, explode=(0, 0.1),*

*colors=['#FF6B6B', '#4ECDC4'])*

*ax1.set\_title('Market Share Distribution (FY2025)', fontweight='bold',fontsize=10)*

*# comparision of sales*

*sales\_data = pd.DataFrame({*

*'Type': ['Petrol', 'Electric'],*

*'Sales (Millions)': [total\_petrol/1e6, total\_electric/1e6]*

*})*

*bars = ax2.bar(sales\_data['Type'], sales\_data['Sales (Millions)'],*

*color=['#FF6B6B', '#4ECDC4'])*

*ax2.set\_title('Total Sales Volume (FY2025)', fontweight='bold',fontsize=10)*

*ax2.set\_ylabel('Sales (Millions)')*

*for bar in bars:*

*height = bar.get\_height()*

*ax2.text(bar.get\_x() + bar.get\_width()/2., height,*

*f'{height:.2f}M', ha='center', va='bottom')*

*# comparision of growth*

*growth\_data = pd.DataFrame({*

*'Type': ['Petrol', 'Electric'],*

*'Growth (%)': [petrol\_growth, electric\_growth]*

*})*

*bars = ax3.bar(growth\_data['Type'], growth\_data['Growth (%)'],*

*color=['#FF6B6B', '#4ECDC4'])*

*ax3.set\_title('Year-over-Year Growth (FY2025)', fontweight='bold',fontsize=10)*

*ax3.set\_ylabel('Growth Rate (%)')*

*for bar in bars:*

*height = bar.get\_height()*

*ax3.text(bar.get\_x() + bar.get\_width()/2., height,*

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

*# future projections of market*

*years = np.array([2025, 2026, 2027, 2028, 2030])*

*petrol\_projection = total\_petrol \* (1.065 \*\* (years - 2025))  # 6.5% CAGR*

*electric\_projection = total\_electric \* (1.30 \*\* (years - 2025))  # 30% CAGR*

*ax4.plot(years, petrol\_projection/1e6, marker='o', label='Petrol', linewidth=2, color='#FF6B6B')*

*ax4.plot(years, electric\_projection/1e6, marker='s', label='Electric', linewidth=2, color='#4ECDC4')*

*ax4.set\_title('Market Projection (2025-2030)', fontweight='bold',fontsize=10)*

*ax4.set\_xlabel('Year')*

*ax4.set\_ylabel('Sales (Millions)')*

*ax4.legend()*

*ax4.grid(True, alpha=0.3)*

*plt.suptitle('Electric vs Petrol Scooters: Market Analysis', fontsize=12, fontweight='bold')*

*plt.tight\_layout()*

*plt.show()*

*# 2.cost of ownership*

*print("\n" + "\_"\*60)*

*print("2. TOTAL COST OF OWNERSHIP ANALYSIS")*

*print("\_"\*60)*

*# cost calculations*

*avg\_battery\_cost = battery\_df['Replacement Cost Avg (₹)'].mean()*

*avg\_battery\_life = battery\_df['Lifespan Years Avg'].mean()*

*# breakdown of cost*

*years = 10  # 10-year analysis*

*annual\_km = 12000*

*# petrol costs*

*petrol\_price = 105  # Rs/liter*

*avg\_mileage = 45  # Km/liter*

*annual\_fuel\_cost = (annual\_km / avg\_mileage) \* petrol\_price*

*# electric costs*

*electricity\_rate = 6  # Rs/kWh*

*ev\_efficiency = 0.8  # KWh/100Km*

*annual\_electricity\_cost = (annual\_km / 100) \* ev\_efficiency \* electricity\_rate*

*# maintenance cost*

*petrol\_maintenance = [3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 11000, 12000]*

*electric\_maintenance = [1500, 2000, 2500, 3000, 3500, 4000, 4500, 5000, 5500, 6000]*

*# insurance cost*

*insurance\_petrol = 4000  # Annual*

*insurance\_electric = 3500  # Annual*

*# yearly cost*

*yearly\_costs = pd.DataFrame({*

*'Year': range(1, years + 1),*

*'Petrol\_Fuel': [annual\_fuel\_cost] \* years,*

*'Electric\_Electricity': [annual\_electricity\_cost] \* years,*

*'Petrol\_Maintenance': petrol\_maintenance,*

*'Electric\_Maintenance': electric\_maintenance,*

*'Petrol\_Insurance': [insurance\_petrol] \* years,*

*'Electric\_Insurance': [insurance\_electric] \* years*

*})*

*# battery replacement costs*

*battery\_replacements = np.zeros(years)*

*for year in range(1, years + 1):*

*if year % int(avg\_battery\_life) == 0:*

*battery\_replacements[year - 1] = avg\_battery\_cost*

*yearly\_costs['Electric\_Battery'] = battery\_replacements*

*# overall cost*

*petrol\_total = yearly\_costs[['Petrol\_Fuel', 'Petrol\_Maintenance', 'Petrol\_Insurance']].sum(axis=1).cumsum()*

*electric\_total = yearly\_costs[['Electric\_Electricity', 'Electric\_Maintenance', 'Electric\_Insurance', 'Electric\_Battery']].sum(axis=1).cumsum()*

*# analysis of cost*

*cost\_breakdown = pd.DataFrame({*

*'Cost Component': ['Fuel/Electricity', 'Maintenance', 'Insurance', 'Battery Replacement', 'Total 10-Year Cost'],*

*'Petrol Scooter (₹)': [*

*annual\_fuel\_cost \* years,*

*sum(petrol\_maintenance),*

*insurance\_petrol \* years,*

*0,*

*petrol\_total.iloc[-1]*

*],*

*'Electric Scooter (₹)': [*

*annual\_electricity\_cost \* years,*

*sum(electric\_maintenance),*

*insurance\_electric \* years,*

*avg\_battery\_cost \* (years / avg\_battery\_life),*

*electric\_total.iloc[-1]*

*]*

*})*

*cost\_breakdown['Savings with Electric (₹)'] = cost\_breakdown['Petrol Scooter (₹)'] - cost\_breakdown['Electric Scooter (₹)']*

*print("\n10-Year Total Cost of Ownership:")*

*print(cost\_breakdown.to\_string(index=False))*

*# visualize cost analysis*

*fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(16, 6))*

*# cost over time*

*ax1.plot(yearly\_costs['Year'], petrol\_total, marker='o', label='Petrol', linewidth=2, color='#FF6B6B')*

*ax1.plot(yearly\_costs['Year'], electric\_total, marker='s', label='Electric', linewidth=2, color='#4ECDC4')*

*ax1.fill\_between(yearly\_costs['Year'], petrol\_total, electric\_total,*

*where=(petrol\_total >= electric\_total), alpha=0.3, color='green', label='Savings')*

*ax1.set\_title('Cumulative Cost Over Time', fontweight='bold',fontsize=10)*

*ax1.set\_xlabel('Year')*

*ax1.set\_ylabel('Cumulative Cost (₹)')*

*ax1.legend()*

*ax1.grid(True, alpha=0.3)*

*# cost breakdown comparison*

*categories = ['Fuel/Electricity', 'Maintenance', 'Insurance', 'Battery']*

*petrol\_costs = [annual\_fuel\_cost \* years, sum(petrol\_maintenance), insurance\_petrol \* years, 0]*

*electric\_costs = [annual\_electricity\_cost \* years, sum(electric\_maintenance),*

*insurance\_electric \* years, avg\_battery\_cost \* (years / avg\_battery\_life)]*

*x = np.arange(len(categories))*

*width = 0.35*

*bars1 = ax2.bar(x - width/2, petrol\_costs, width, label='Petrol', color='#FF6B6B')*

*bars2 = ax2.bar(x + width/2, electric\_costs, width, label='Electric', color='#4ECDC4')*

*ax2.set\_title('Cost Breakdown Comparison (10 Years)', fontweight='bold',fontsize=10)*

*ax2.set\_ylabel('Cost (₹)')*

*ax2.set\_xticks(x)*

*ax2.set\_xticklabels(categories)*

*ax2.legend()*

*# add value labels on bars*

*for bars in [bars1, bars2]:*

*for bar in bars:*

*height = bar.get\_height()*

*ax2.text(bar.get\_x() + bar.get\_width()/2., height,*

*f'₹{height:,.0f}', ha='center', va='bottom', fontsize=9)*

*plt.tight\_layout()*

*plt.show()*

*# 3. ENVIRONMENTAL IMPACT ANALYSIS*

*print("\n" + "\_"\*60)*

*print("3. ENVIRONMENTAL IMPACT ANALYSIS")*

*print("\_"\*60)*

*# carbon footprint calculation*

*petrol\_co2\_per\_liter = 2.31  # kg CO2 per liter*

*electricity\_co2\_per\_kwh = 0.82  # kg CO2 per kWh (India grid average)*

*# annual emissions*

*annual\_petrol\_liters = annual\_km / avg\_mileage*

*annual\_petrol\_co2 = annual\_petrol\_liters \* petrol\_co2\_per\_liter*

*annual\_electric\_kwh = (annual\_km / 100) \* ev\_efficiency*

*annual\_electric\_co2 = annual\_electric\_kwh \* electricity\_co2\_per\_kwh*

*# 10-year environmental impact*

*lifetime\_petrol\_co2 = annual\_petrol\_co2 \* years*

*lifetime\_electric\_co2 = annual\_electric\_co2 \* years*

*# manufacturing footprint (estimated)*

*manufacturing\_petrol\_co2 = 2000  # kg CO2*

*manufacturing\_electric\_co2 = 3500  # kg CO2 (higher due to battery)*

*# total lifecycle emissions*

*total\_petrol\_co2 = lifetime\_petrol\_co2 + manufacturing\_petrol\_co2*

*total\_electric\_co2 = lifetime\_electric\_co2 + manufacturing\_electric\_co2*

*env\_impact = pd.DataFrame({*

*'Impact Category': ['Annual CO2 Emissions (kg)', 'Lifetime CO2 Emissions (kg)',*

*'Manufacturing CO2 (kg)', 'Total Lifecycle CO2 (kg)',*

*'Annual Fuel Consumption', 'Noise Pollution (dB)',*

*'Local Air Pollution'],*

*'Petrol Scooter': [f"{annual\_petrol\_co2:.1f}", f"{lifetime\_petrol\_co2:.1f}",*

*f"{manufacturing\_petrol\_co2:.1f}", f"{total\_petrol\_co2:.1f}",*

*f"{annual\_petrol\_liters:.1f} liters", "75-80", "High"],*

*'Electric Scooter': [f"{annual\_electric\_co2:.1f}", f"{lifetime\_electric\_co2:.1f}",*

*f"{manufacturing\_electric\_co2:.1f}", f"{total\_electric\_co2:.1f}",*

*f"{annual\_electric\_kwh:.1f} kWh", "45-50", "Zero"]*

*})*

*print("\nEnvironmental Impact Comparison:")*

*print(env\_impact.to\_string(index=False))*

*co2\_savings = total\_petrol\_co2 - total\_electric\_co2*

*print(f"\nCO2 Savings with Electric: {co2\_savings:.1f} kg over 10 years")*

*print(f"Equivalent to planting {co2\_savings/22:.0f} trees")*

*# visualize environmental impact*

*fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(14, 10))*

*# annual emissions comparison*

*categories = ['Annual CO2 (kg)', 'Lifetime CO2 (kg)', 'Manufacturing CO2 (kg)']*

*petrol\_emissions = [annual\_petrol\_co2, lifetime\_petrol\_co2, manufacturing\_petrol\_co2]*

*electric\_emissions = [annual\_electric\_co2, lifetime\_electric\_co2, manufacturing\_electric\_co2]*

*x = np.arange(len(categories))*

*width = 0.35*

*bars1 = ax1.bar(x - width/2, petrol\_emissions, width, label='Petrol', color='#FF6B6B')*

*bars2 = ax1.bar(x + width/2, electric\_emissions, width, label='Electric', color='#4ECDC4')*

*ax1.set\_title('CO2 Emissions Comparison', fontweight='bold',fontsize=10)*

*ax1.set\_ylabel('CO2 Emissions (kg)')*

*ax1.set\_xticks(x)*

*ax1.set\_xticklabels(categories)*

*ax1.legend()*

*# cumulative emissions over time*

*years\_range = np.arange(1, years + 1)*

*cumulative\_petrol = annual\_petrol\_co2 \* years\_range + manufacturing\_petrol\_co2*

*cumulative\_electric = annual\_electric\_co2 \* years\_range + manufacturing\_electric\_co2*

*ax2.plot(years\_range, cumulative\_petrol, marker='o', label='Petrol', linewidth=2, color='#FF6B6B')*

*ax2.plot(years\_range, cumulative\_electric, marker='s', label='Electric', linewidth=2, color='#4ECDC4')*

*ax2.fill\_between(years\_range, cumulative\_petrol, cumulative\_electric,*

*where=(cumulative\_petrol >= cumulative\_electric), alpha=0.3, color='green')*

*ax2.set\_title('Cumulative CO2 Emissions Over Time', fontweight='bold',fontsize=10)*

*ax2.set\_xlabel('Year')*

*ax2.set\_ylabel('Cumulative CO2 (kg)')*

*ax2.legend()*

*ax2.grid(True, alpha=0.3)*

*# energy consumption comparison*

*energy\_types = ['Petrol Consumption\n(Liters/Year)', 'Electricity Consumption\n(kWh/Year)']*

*energy\_values = [annual\_petrol\_liters, annual\_electric\_kwh]*

*colors = ['#FF6B6B', '#4ECDC4']*

*bars = ax3.bar(energy\_types, energy\_values, color=colors)*

*ax3.set\_title('Annual Energy Consumption', fontweight='bold',fontsize=10)*

*ax3.set\_ylabel('Energy Units')*

*for bar in bars:*

*height = bar.get\_height()*

*ax3.text(bar.get\_x() + bar.get\_width()/2., height,*

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

*# Noise pollution*

*noise\_levels = ['Petrol Scooter', 'Electric Scooter']*

*noise\_values = [77.5, 47.5]  # Average noise levels in dB*

*bars = ax4.bar(noise\_levels, noise\_values, color=['#FF6B6B', '#4ECDC4'])*

*ax4.set\_title('Noise Pollution Levels', fontweight='bold',fontsize=10)*

*ax4.set\_ylabel('Noise Level (dB)')*

*ax4.axhline(y=55, color='red', linestyle='--', alpha=0.7, label='WHO Recommended Limit')*

*ax4.legend()*

*for bar in bars:*

*height = bar.get\_height()*

*ax4.text(bar.get\_x() + bar.get\_width()/2., height,*

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

*plt.suptitle('Environmental Impact Analysis', fontsize=12, fontweight='bold')*

*plt.tight\_layout()*

*plt.show()*

*# 4. PERFORMANCE & TECHNOLOGY COMPARISON*

*print("\n" + "\_"\*60)*

*print("4. PERFORMANCE & TECHNOLOGY COMPARISON")*

*print("\_"\*60)*

*# performance*

*performance\_metrics = pd.DataFrame({*

*'Metric': ['Top Speed (km/h)', 'Acceleration (0-40 km/h)', 'Range (km)',*

*'Refuel/Recharge Time', 'Power (kW)', 'Torque (Nm)',*

*'Weight (kg)', 'Efficiency', 'Cold Start Performance'],*

*'Petrol Scooter': ['80-90', '8-10 sec', '150-200', '2-3 min', '6-8', '8-10',*

*'100-120', '40-50 km/L', 'Good'],*

*'Electric Scooter': ['60-80', '6-8 sec', '80-120', '3-6 hours', '2-4', '15-25',*

*'80-100', '80-100 km/kWh', 'Excellent']*

*})*

*print("\nPerformance Comparison:")*

*print(performance\_metrics.to\_string(index=False))*

*# features*

*tech\_features = pd.DataFrame({*

*'Feature': ['Instant Torque', 'Smart Connectivity', 'GPS Tracking', 'Mobile App',*

*'Regenerative Braking', 'Multiple Ride Modes', 'Anti-theft System',*

*'Digital Dashboard', 'OTA Updates', 'Maintenance Alerts'],*

*'Petrol Scooter': ['No', 'Limited', 'Optional', 'Basic', 'No', 'No', 'Basic',*

*'Analog/Basic', 'No', 'Manual'],*

*'Electric Scooter': ['Yes', 'Advanced', 'Standard', 'Comprehensive', 'Yes', 'Yes', 'Advanced',*

*'Fully Digital', 'Yes', 'Automatic']*

*})*

*print("\nTechnology Features Comparison:")*

*print(tech\_features.to\_string(index=False))*

*# Visualize performance comparison*

*fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(14, 10))*

*# Performance radar chart data*

*categories = ['Top Speed', 'Acceleration', 'Range', 'Efficiency', 'Torque']*

*petrol\_scores = [85, 70, 90, 60, 60]  # Normalized scores out of 100*

*electric\_scores = [70, 85, 70, 90, 90]*

*# Convert to radar chart*

*angles = np.linspace(0, 2 \* np.pi, len(categories), endpoint=False).tolist()*

*angles += angles[:1]  # Complete the circle*

*petrol\_scores += petrol\_scores[:1]*

*electric\_scores += electric\_scores[:1]*

*ax1 = plt.subplot(2, 2, 1, projection='polar')*

*ax1.plot(angles, petrol\_scores, 'o-', linewidth=2, label='Petrol', color='#FF6B6B')*

*ax1.fill(angles, petrol\_scores, alpha=0.25, color='#FF6B6B')*

*ax1.plot(angles, electric\_scores, 's-', linewidth=2, label='Electric', color='#4ECDC4')*

*ax1.fill(angles, electric\_scores, alpha=0.25, color='#4ECDC4')*

*ax1.set\_xticks(angles[:-1])*

*ax1.set\_xticklabels(categories)*

*ax1.set\_ylim(0, 100)*

*ax1.set\_title('Performance Comparison', fontweight='bold',fontsize=10, pad=20)*

*ax1.legend()*

*# Technology adoption score*

*ax2 = plt.subplot(2, 2, 2)*

*tech\_categories = ['Connectivity', 'Safety', 'Convenience', 'Sustainability', 'Innovation']*

*petrol\_tech = [30, 60, 50, 20, 30]*

*electric\_tech = [85, 80, 90, 95, 90]*

*x = np.arange(len(tech\_categories))*

*width = 0.35*

*bars1 = ax2.bar(x - width/2, petrol\_tech, width, label='Petrol', color='#FF6B6B')*

*bars2 = ax2.bar(x + width/2, electric\_tech, width, label='Electric', color='#4ECDC4')*

*ax2.set\_title('Technology Adoption Score', fontweight='bold',fontsize=10)*

*ax2.set\_ylabel('Score (0-100)')*

*ax2.set\_xticks(x)*

*ax2.set\_xticklabels(tech\_categories, rotation=45)*

*ax2.legend()*

*# Efficiency comparison*

*ax3 = plt.subplot(2, 2, 3)*

*efficiency\_data = pd.DataFrame({*

*'Vehicle Type': ['Petrol', 'Electric'],*

*'Energy Efficiency (km/unit)': [45, 90],  # km/liter vs km/kWh equivalent*

*'Cost per km (₹)': [2.33, 0.58]  # Cost per km*

*})*

*ax3\_twin = ax3.twinx()*

*bars1 = ax3.bar(efficiency\_data['Vehicle Type'], efficiency\_data['Energy Efficiency (km/unit)'],*

*alpha=0.7, color=['#FF6B6B', '#4ECDC4'], label='Efficiency')*

*ax3.set\_title('Efficiency & Cost Comparison', fontweight='bold',fontsize=10)*

*ax3.set\_ylabel('Energy Efficiency (km/unit)')*

*ax3\_twin.set\_ylabel('Cost per km (₹)')*

*# Maintenance requirements*

*ax4 = plt.subplot(2, 2, 4)*

*maintenance\_freq = pd.DataFrame({*

*'Service Type': ['Oil Change', 'Filter Replacement', 'Spark Plug', 'Battery Check',*

*'Brake Service', 'Software Update'],*

*'Petrol Frequency (months)': [3, 6, 12, 12, 12, 0],*

*'Electric Frequency (months)': [0, 0, 0, 6, 18, 3]*

*})*

*service\_types = maintenance\_freq['Service Type']*

*petrol\_freq = maintenance\_freq['Petrol Frequency (months)']*

*electric\_freq = maintenance\_freq['Electric Frequency (months)']*

*x = np.arange(len(service\_types))*

*width = 0.35*

*bars1 = ax4.bar(x - width/2, petrol\_freq, width, label='Petrol', color='#FF6B6B')*

*bars2 = ax4.bar(x + width/2, electric\_freq, width, label='Electric', color='#4ECDC4')*

*ax4.set\_title('Maintenance Frequency', fontweight='bold',fontsize=10)*

*ax4.set\_ylabel('Frequency (months)')*

*ax4.set\_xticks(x)*

*ax4.set\_xticklabels(service\_types, rotation=45)*

*ax4.legend()*

*plt.tight\_layout()*

*plt.show()*

*# 5. INFRASTRUCTURE & ECOSYSTEM ANALYSIS*

*print("\n" + "\_"\*60)*

*print("5. INFRASTRUCTURE & ECOSYSTEM ANALYSIS")*

*print("\_"\*60)*

*# Infrastructure data (estimated for major Indian cities)*

*infrastructure = pd.DataFrame({*

*'Infrastructure Type': ['Fuel Stations', 'Charging Stations', 'Service Centers',*

*'Spare Parts Availability', 'Charging Time', 'Accessibility'],*

*'Petrol Scooter': ['~70,000', '0', '~15,000', 'Excellent', '2-3 min', 'Universal'],*

*'Electric Scooter': ['0', '~5,000', '~2,000', 'Good', '3-6 hours', 'Growing']*

*})*

*print("\nInfrastructure Comparison:")*

*print(infrastructure.to\_string(index=False))*

*# Government incentives*

*incentives = pd.DataFrame({*

*'Incentive Type': ['FAME-II Subsidy', 'State Subsidies', 'Road Tax Exemption',*

*'Registration Fee Waiver', 'Insurance Discount', 'Loan Interest Rate'],*

*'Petrol Scooter': ['₹0', '₹0', 'Standard', 'Standard', 'Standard', '12-15%'],*

*'Electric Scooter': ['₹10,000-15,000', '₹5,000-25,000', 'Waived', 'Waived', '10-15% off', '8-12%']*

*})*

*print("\nGovernment Incentives:")*

*print(incentives.to\_string(index=False))*

*# Visualize infrastructure analysis*

*fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(14, 10))*

*# Infrastructure availability*

*infra\_categories = ['Fuel/Charge Stations', 'Service Centers']*

*petrol\_infra = [70000, 15000]*

*electric\_infra = [5000, 2000]*

*x = np.arange(len(infra\_categories))*

*width = 0.35*

*bars1 = ax1.bar(x - width/2, petrol\_infra, width, label='Petrol', color='#FF6B6B')*

*bars2 = ax1.bar(x + width/2, electric\_infra, width, label='Electric', color='#4ECDC4')*

*ax1.set\_title('Infrastructure Availability in India', fontweight='bold',fontsize=10)*

*ax1.set\_ylabel('Number of Stations/Centers')*

*ax1.set\_xticks(x)*

*ax1.set\_xticklabels(infra\_categories)*

*ax1.legend()*

*ax1.set\_yscale('log')  # Log scale due to large difference*

*# Charging vs Refueling time*

*refuel\_data = pd.DataFrame({*

*'Type': ['Petrol Refuel', 'Electric Charge (Fast)', 'Electric Charge (Normal)'],*

*'Time (minutes)': [3, 45, 240],*

*'Range Added (km)': [150, 60, 100]*

*})*

*ax2\_twin = ax2.twinx()*

*bars = ax2.bar(refuel\_data['Type'], refuel\_data['Time (minutes)'],*

*color=['#FF6B6B', '#4ECDC4', '#95E1D3'], alpha=0.7)*

*ax2.set\_title('Refuel/Recharge Time vs Range', fontweight='bold',fontsize=10)*

*ax2.set\_ylabel('Time (minutes)')*

*ax2\_twin.set\_ylabel('Range Added (km)')*

*ax2.set\_xticklabels(refuel\_data['Type'], rotation=45)*

*# Government incentive value*

*incentive\_values = pd.DataFrame({*

*'Incentive': ['FAME-II', 'State Subsidy', 'Tax Benefits'],*

*'Petrol (₹)': [0, 0, 0],*

*'Electric (₹)': [12500, 15000, 8000]*

*})*

*x = np.arange(len(incentive\_values))*

*width = 0.35*

*bars1 = ax3.bar(x - width/2, incentive\_values['Petrol (₹)'], width,*

*label='Petrol', color='#FF6B6B')*

*bars2 = ax3.bar(x + width/2, incentive\_values['Electric (₹)'], width,*

*label='Electric', color='#4ECDC4')*

*ax3.set\_title('Government Incentives Value', fontweight='bold',fontsize=10)*

*ax3.set\_ylabel('Incentive Amount (₹)')*

*ax3.set\_xticks(x)*

*ax3.set\_xticklabels(incentive\_values['Incentive'])*

*ax3.legend()*

*# Infrastructure growth projection*

*years\_infra = np.arange(2025, 2031)*

*petrol\_stations = np.array([70000, 72000, 74000, 76000, 78000, 80000])*

*electric\_stations = np.array([5000, 8000, 15000, 25000, 40000, 60000])*

*ax4.plot(years\_infra, petrol\_stations, marker='o', label='Petrol Stations',*

*linewidth=2, color='#FF6B6B')*

*ax4.plot(years\_infra, electric\_stations, marker='s', label='Charging Stations',*

*linewidth=2, color='#4ECDC4')*

*ax4.set\_title('Infrastructure Growth Projection', fontweight='bold',fontsize=10)*

*ax4.set\_xlabel('Year')*

*ax4.set\_ylabel('Number of Stations')*

*ax4.legend()*

*ax4.grid(True, alpha=0.3)*

*plt.tight\_layout()*

*plt.show()*

*# 6. CONSUMER BEHAVIOR & ADOPTION ANALYSIS*

*print("\n" + "\_"\*60)*

*print("6. CONSUMER BEHAVIOR & ADOPTION ANALYSIS")*

*print("\_"\*60)*

*# Consumer preference factors*

*preference\_factors = pd.DataFrame({*

*'Factor': ['Purchase Price', 'Running Cost', 'Maintenance', 'Environment',*

*'Performance', 'Convenience', 'Brand Trust', 'Technology'],*

*'Importance (1-10)': [9, 8, 7, 6, 8, 9, 8, 7],*

*'Petrol Score': [8, 6, 5, 3, 7, 9, 9, 5],*

*'Electric Score': [6, 9, 8, 10, 6, 6, 6, 9]*

*})*

*print("\nConsumer Preference Analysis:")*

*print(preference\_factors.to\_string(index=False))*

*# Calculate weighted scores*

*preference\_factors['Petrol Weighted'] = preference\_factors['Importance (1-10)'] \* preference\_factors['Petrol Score']*

*preference\_factors['Electric Weighted'] = preference\_factors['Importance (1-10)'] \* preference\_factors['Electric Score']*

*total\_petrol\_score = preference\_factors['Petrol Weighted'].sum()*

*total\_electric\_score = preference\_factors['Electric Weighted'].sum()*

*print(f"\nOverall Consumer Preference Scores:")*

*print(f"Petrol Scooters: {total\_petrol\_score}/800")*

*print(f"Electric Scooters: {total\_electric\_score}/800")*

*# Adoption barriers*

*barriers = pd.DataFrame({*

*'Barrier': ['High Initial Cost', 'Limited Range', 'Charging Infrastructure',*

*'Charging Time', 'Battery Replacement', 'Brand Awareness',*

*'Service Network', 'Resale Value Uncertainty'],*

*'Severity (1-10)': [8, 7, 9, 8, 7, 6, 7, 8],*

*'Trend': ['Improving', 'Improving', 'Improving', 'Improving',*

*'Stable', 'Improving', 'Improving', 'Stable']*

*})*

*print("\nAdoption Barriers for Electric Scooters:")*

*print(barriers.to\_string(index=False))*

*# Visualize consumer analysis*

*fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(14, 8))*

*# Consumer preference radar chart*

*categories = preference\_factors['Factor'].tolist()*

*petrol\_scores = preference\_factors['Petrol Score'].tolist()*

*electric\_scores = preference\_factors['Electric Score'].tolist()*

*angles = np.linspace(0, 2 \* np.pi, len(categories), endpoint=False).tolist()*

*angles += angles[:1]*

*petrol\_scores += petrol\_scores[:1]*

*electric\_scores += electric\_scores[:1]*

*ax1 = plt.subplot(2, 2, 1, projection='polar')*

*ax1.plot(angles, petrol\_scores, 'o-', linewidth=2, label='Petrol', color='#FF6B6B')*

*ax1.fill(angles, petrol\_scores, alpha=0.25, color='#FF6B6B')*

*ax1.plot(angles, electric\_scores, 's-', linewidth=2, label='Electric', color='#4ECDC4')*

*ax1.fill(angles, electric\_scores, alpha=0.25, color='#4ECDC4')*

*ax1.set\_xticks(angles[:-1])*

*ax1.set\_xticklabels(categories, fontsize=9)*

*ax1.set\_ylim(0, 10)*

*ax1.set\_title('Consumer Preference Factors', fontweight='bold',fontsize=10, pad=20)*

*ax1.legend()*

*# Adoption barriers*

*ax2 = plt.subplot(2, 2, 2)*

*barrier\_names = barriers['Barrier']*

*severity = barriers['Severity (1-10)']*

*colors = ['#FF6B6B' if trend == 'Stable' else '#4ECDC4' for trend in barriers['Trend']]*

*bars = ax2.barh(barrier\_names, severity, color=colors)*

*ax2.set\_title('Electric Scooter Adoption Barriers', fontweight='bold',fontsize=10)*

*ax2.set\_xlabel('Severity (1-10)')*

*# Market segment analysis*

*ax3 = plt.subplot(2, 2, 3)*

*segments = ['Budget\n(<₹60k)', 'Mid-range\n(₹60k-₹1L)', 'Premium\n(>₹1L)']*

*petrol\_segments = [70, 25, 5]*

*electric\_segments = [40, 45, 15]*

*x = np.arange(len(segments))*

*width = 0.35*

*bars1 = ax3.bar(x - width/2, petrol\_segments, width, label='Petrol', color='#FF6B6B')*

*bars2 = ax3.bar(x + width/2, electric\_segments, width, label='Electric', color='#4ECDC4')*

*ax3.set\_title('Market Segment Distribution (%)', fontweight='bold',fontsize=10)*

*ax3.set\_ylabel('Market Share (%)')*

*ax3.set\_xticks(x)*

*ax3.set\_xticklabels(segments)*

*ax3.legend()*

*# Adoption timeline prediction*

*ax4 = plt.subplot(2, 2, 4)*

*years\_adoption = np.arange(2025, 2031)*

*petrol\_adoption = np.array([85, 80, 75, 65, 55, 45])*

*electric\_adoption = np.array([15, 20, 25, 35, 45, 55])*

*ax4.plot(years\_adoption, petrol\_adoption, marker='o', label='Petrol Market Share',*

*linewidth=2, color='#FF6B6B')*

*ax4.plot(years\_adoption, electric\_adoption, marker='s', label='Electric Market Share',*

*linewidth=2, color='#4ECDC4')*

*ax4.axhline(y=50, color='gray', linestyle='--', alpha=0.7, label='50% Market Share')*

*ax4.set\_title('Market Share Evolution Prediction', fontweight='bold',fontsize=10)*

*ax4.set\_xlabel('Year')*

*ax4.set\_ylabel('Market Share (%)')*

*ax4.legend()*

*ax4.grid(True, alpha=0.3)*

*plt.tight\_layout()*

*plt.show()*

*# 7. MANUFACTURER ECOSYSTEM ANALYSIS*

*print("\n" + "\_"\*60)*

*print("7. MANUFACTURER ECOSYSTEM ANALYSIS")*

*print("\_"\*60)*

*# Process manufacturer data*

*petrol\_mfg = petrol\_df[~petrol\_df['Manufacturer'].str.contains('Total', na=False)]*

*petrol\_mfg = petrol\_mfg.sort\_values('FY2025', ascending=False).head(10)*

*# Process electric manufacturer data*

*if len(electric\_mfg) > 25:*

*electric\_mfg = electric\_mfg.iloc[1:26]*

*manufacturer\_col = None*

*for col in ['Manufacturer', 'Company', 'Brand', 'Name', '1']:*

*if col in electric\_mfg.columns:*

*manufacturer\_col = col*

*break*

*if manufacturer\_col is None:*

*manufacturer\_col = electric\_mfg.columns[0]*

*# Calculate total sales if not present*

*if 'Total' not in electric\_mfg.columns:*

*numeric\_cols = electric\_mfg.select\_dtypes(include=[np.number]).columns*

*if len(numeric\_cols) > 0:*

*electric\_mfg['Total'] = electric\_mfg[numeric\_cols].sum(axis=1)*

*else:*

*for col in electric\_mfg.columns:*

*if col != manufacturer\_col:*

*electric\_mfg[col] = pd.to\_numeric(electric\_mfg[col], errors='coerce')*

*numeric\_cols = electric\_mfg.select\_dtypes(include=[np.number]).columns*

*electric\_mfg['Total'] = electric\_mfg[numeric\_cols].sum(axis=1)*

*electric\_mfg['Total'] = pd.to\_numeric(electric\_mfg['Total'], errors='coerce')*

*electric\_mfg = electric\_mfg.dropna(subset=['Total']).sort\_values('Total', ascending=False).head(10)*

*# Market concentration analysis*

*petrol\_top3 = petrol\_mfg.head(3)['FY2025'].sum()*

*petrol\_hhi = sum((petrol\_mfg['FY2025'] / petrol\_mfg['FY2025'].sum()) \*\* 2)*

*electric\_top3 = electric\_mfg.head(3)['Total'].sum()*

*electric\_hhi = sum((electric\_mfg['Total'] / electric\_mfg['Total'].sum()) \*\* 2)*

*market\_concentration = pd.DataFrame({*

*'Metric': ['Top 3 Market Share (%)', 'HHI (Concentration Index)', 'Market Structure'],*

*'Petrol Market': [f"{(petrol\_top3/petrol\_mfg['FY2025'].sum())\*100:.1f}%",*

*f"{petrol\_hhi:.3f}", 'Moderately Concentrated'],*

*'Electric Market': [f"{(electric\_top3/electric\_mfg['Total'].sum())\*100:.1f}%",*

*f"{electric\_hhi:.3f}", 'Highly Concentrated']*

*})*

*print("\nMarket Concentration Analysis:")*

*print(market\_concentration.to\_string(index=False))*

*# Manufacturer performance*

*print(f"\nTop 5 Petrol Scooter Manufacturers:")*

*petrol\_top5 = petrol\_mfg.head(5)[['Manufacturer', 'FY2025', 'YoY % change']].copy()*

*petrol\_top5['Market Share (%)'] = (petrol\_top5['FY2025'] / petrol\_mfg['FY2025'].sum()) \* 100*

*print(petrol\_top5.to\_string(index=False))*

*print(f"\nTop 5 Electric Scooter Manufacturers:")*

*electric\_top5 = electric\_mfg.head(5)[[manufacturer\_col, 'Total']].copy()*

*electric\_top5['Market Share (%)'] = (electric\_top5['Total'] / electric\_mfg['Total'].sum()) \* 100*

*electric\_top5.columns = ['Manufacturer', 'FY2025', 'Market Share (%)']*

*print(electric\_top5.to\_string(index=False))*

*# Visualize manufacturer analysis*

*fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(14, 10))*

*# Top manufacturers comparison*

*top\_petrol = petrol\_mfg.head(5)*

*top\_electric = electric\_mfg.head(5)*

*ax1.barh(top\_petrol['Manufacturer'], top\_petrol['FY2025'],*

*color='#FF6B6B', alpha=0.7, label='Petrol')*

*ax1.set\_title('Top 5 Petrol Manufacturers', fontweight='bold',fontsize=10)*

*ax1.set\_xlabel('Sales (Units)')*

*ax2.barh(top\_electric[manufacturer\_col], top\_electric['Total'],*

*color='#4ECDC4', alpha=0.7, label='Electric')*

*ax2.set\_title('Top 5 Electric Manufacturers', fontweight='bold',fontsize=10)*

*ax2.set\_xlabel('Sales (Units)')*

*# Market share pie charts*

*ax3.pie(petrol\_top5['Market Share (%)'], labels=petrol\_top5['Manufacturer'],*

*autopct='%1.1f%%', startangle=90)*

*ax3.set\_title('Petrol Market Share Distribution', fontweight='bold',fontsize=10)*

*ax4.pie(electric\_top5['Market Share (%)'], labels=electric\_top5['Manufacturer'],*

*autopct='%1.1f%%', startangle=90)*

*ax4.set\_title('Electric Market Share Distribution', fontweight='bold',fontsize=10)*

*plt.tight\_layout()*

*plt.show()*

*# 8. FUTURE OUTLOOK & PREDICTIONS*

*print("\n" + "\_"\*60)*

*print("8. FUTURE OUTLOOK & PREDICTIONS")*

*print("\_"\*60)*

*# Technology roadmap*

*tech\_roadmap = pd.DataFrame({*

*'Timeline': ['2025-2026', '2027-2028', '2029-2030'],*

*'Electric Technology': ['Improved battery density, 150km range',*

*'Solid-state batteries, 200km range',*

*'Wireless charging, 300km range'],*

*'Petrol Technology': ['BS-VII norms, minor efficiency gains',*

*'Hybrid variants introduction',*

*'Phase-out regulations'],*

*'Infrastructure': ['20k charging stations', '50k charging stations',*

*'100k charging stations'],*

*'Market Prediction': ['20% electric share', '40% electric share',*

*'60% electric share']*

*})*

*print("\nTechnology & Market Roadmap:")*

*print(tech\_roadmap.to\_string(index=False))*

*# Key predictions*

*predictions = pd.DataFrame({*

*'Aspect': ['Market Share by 2030', 'Average Price Convergence', 'Range Improvement',*

*'Charging Infrastructure', 'Battery Life', 'Government Support'],*

*'Prediction': ['60% Electric, 40% Petrol', '2027-2028', '300km by 2030',*

*'100k+ stations by 2030', '8-10 years by 2030', 'Continued till 2030']*

*})*

*print("\nKey Predictions:")*

*print(predictions.to\_string(index=False))*

*# SWOT Analysis*

*print("\n" + "\_"\*40)*

*print("SWOT ANALYSIS")*

*print("\_"\*40)*

*swot\_electric = pd.DataFrame({*

*'Strengths': ['Lower running costs', 'Zero emissions', 'Instant torque', 'Government support'],*

*'Weaknesses': ['High initial cost', 'Limited range', 'Charging time', 'Infrastructure gaps'],*

*'Opportunities': ['Growing environmental awareness', 'Improving technology', 'Policy support', 'Corporate adoption'],*

*'Threats': ['Battery degradation', 'Competition from petrol', 'Economic slowdown', 'Raw material costs']*

*})*

*print("\nElectric Scooters SWOT:")*

*for category in swot\_electric.columns:*

*print(f"\n{category}:")*

*for item in swot\_electric[category]:*

*print(f"  • {item}")*

*# Final visualization - Future market evolution*

*fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, figsize=(14, 10))*

*# Market evolution timeline*

*future\_years = np.arange(2025, 2031)*

*petrol\_future = np.array([85, 80, 70, 60, 50, 40])*

*electric\_future = np.array([15, 20, 30, 40, 50, 60])*

*ax1.fill\_between(future\_years, 0, petrol\_future, alpha=0.7, color='#FF6B6B', label='Petrol')*

*ax1.fill\_between(future\_years, petrol\_future, 100, alpha=0.7, color='#4ECDC4', label='Electric')*

*ax1.plot(future\_years, petrol\_future, 'o-', color='#FF6B6B', linewidth=2)*

*ax1.plot(future\_years, electric\_future, 's-', color='#4ECDC4', linewidth=2)*

*ax1.set\_title('Market Share Evolution (2025-2030)', fontweight='bold',fontsize=10)*

*ax1.set\_xlabel('Year')*

*ax1.set\_ylabel('Market Share (%)')*

*ax1.legend()*

*ax1.grid(True, alpha=0.3)*

*# Technology advancement timeline*

*tech\_metrics = ['Range (km)', 'Charging Time (min)', 'Battery Life (years)', 'Cost Reduction (%)']*

*current\_values = [100, 240, 5, 0]*

*future\_values = [300, 60, 10, 40]*

*x = np.arange(len(tech\_metrics))*

*width = 0.35*

*bars1 = ax2.bar(x - width/2, current\_values, width, label='2025', color='#FF6B6B')*

*bars2 = ax2.bar(x + width/2, future\_values, width, label='2030', color='#4ECDC4')*

*ax2.set\_title('Electric Technology Evolution', fontweight='bold',fontsize=10)*

*ax2.set\_ylabel('Value')*

*ax2.set\_xticks(x)*

*ax2.set\_xticklabels(tech\_metrics, rotation=45)*

*ax2.legend()*

*# Investment and policy impact*

*policy\_impact = pd.DataFrame({*

*'Policy': ['FAME-II Extension', 'State Incentives', 'ICE Phase-out', 'Charging Infrastructure'],*

*'Impact Score': [8, 7, 9, 8],*

*'Timeline': [2025, 2026, 2030, 2028]*

*})*

*colors = plt.cm.viridis(np.linspace(0, 1, len(policy\_impact)))*

*bars = ax3.bar(policy\_impact['Policy'], policy\_impact['Impact Score'], color=colors)*

*ax3.set\_title('Policy Impact on Electric Adoption', fontweight='bold',fontsize=10)*

*ax3.set\_ylabel('Impact Score (1-10)')*

*ax3.set\_xticklabels(policy\_impact['Policy'], rotation=45)*

*# Risk-reward matrix*

*risk\_levels = ['Low Risk\nHigh Reward', 'Medium Risk\nHigh Reward', 'High Risk\nMedium Reward']*

*petrol\_position = 2  # High risk, medium reward*

*electric\_position = 1  # Medium risk, high reward*

*positions = [0, 1, 2]*

*petrol\_marker = [0, 0, 1]*

*electric\_marker = [0, 1, 0]*

*ax4.bar(positions, petrol\_marker, alpha=0.7, color='#FF6B6B', label='Petrol Investment')*

*ax4.bar(positions, electric\_marker, alpha=0.7, color='#4ECDC4', label='Electric Investment')*

*ax4.set\_title('Investment Risk-Reward Matrix', fontweight='bold',fontsize=10)*

*ax4.set\_ylabel('Investment Attractiveness')*

*ax4.set\_xticks(positions)*

*ax4.set\_xticklabels(risk\_levels)*

*ax4.legend()*

*plt.tight\_layout()*

*plt.show()*

*# 9. EXECUTIVE SUMMARY*

*print("\n" + "\_"\*60)*

*print("9. EXECUTIVE SUMMARY")*

*print("\_"\*60)*

*# Calculate savings using correct variable names*

*savings\_10\_year = petrol\_total.iloc[-1] - electric\_total.iloc[-1]*

*co2\_reduction = total\_petrol\_co2 - total\_electric\_co2*

*print(f"""*

*KEY FINDINGS:*

*Market Position:*

*• Electric scooters hold {electric\_share:.1f}% market share vs {petrol\_share:.1f}% for petrol*

*• Electric segment growing at {electric\_growth:.1f}% vs {petrol\_growth:.1f}% for petrol*

*• Projected to reach 60% market share by 2030*

*Financial Analysis:*

*• 10-year ownership cost savings: ₹{savings\_10\_year:,.0f} with electric*

*• Electric running cost: ₹{annual\_electricity\_cost:,.0f}/year vs ₹{annual\_fuel\_cost:,.0f}/year for petrol*

*• Break-even point: ~3 years for electric scooters*

*Environmental Impact:*

*• CO2 reduction: {co2\_reduction:.0f} kg over 10 years*

*• Equivalent to planting {co2\_reduction/22:.0f} trees*

*• Zero local air pollution with electric*

*Technology & Performance:*

*• Electric offers superior torque and acceleration*

*• Petrol leads in range and refueling convenience*

*• Electric advancing rapidly in all metrics*

*Infrastructure:*

*• Current gap: 70k petrol stations vs 5k charging stations*

*• Projected 100k+ charging stations by 2030*

*• Government incentives worth ₹35,000+ for electric*

*RECOMMENDATIONS:*

*For Consumers:*

*• Consider electric for city commuting (<100km daily)*

*• Petrol remains viable for long-distance travel*

*• Factor in local charging infrastructure*

*For Manufacturers:*

*• Invest heavily in electric R&D and infrastructure*

*• Focus on battery technology and charging solutions*

*• Prepare for market transition by 2030*

*For Policymakers:*

*• Continue FAME-II and state incentive programs*

*• Accelerate charging infrastructure development*

*• Consider ICE phase-out timeline by 2035*

*""")*

*print("\n" + "\_"\*60)*

*print("ANALYSIS COMPLETED SUCCESSFULLY")*

*print("\_"\*60)*

*except Exception as e:*

*print(f"\nError during analysis: {str(e)}")*

*import traceback*

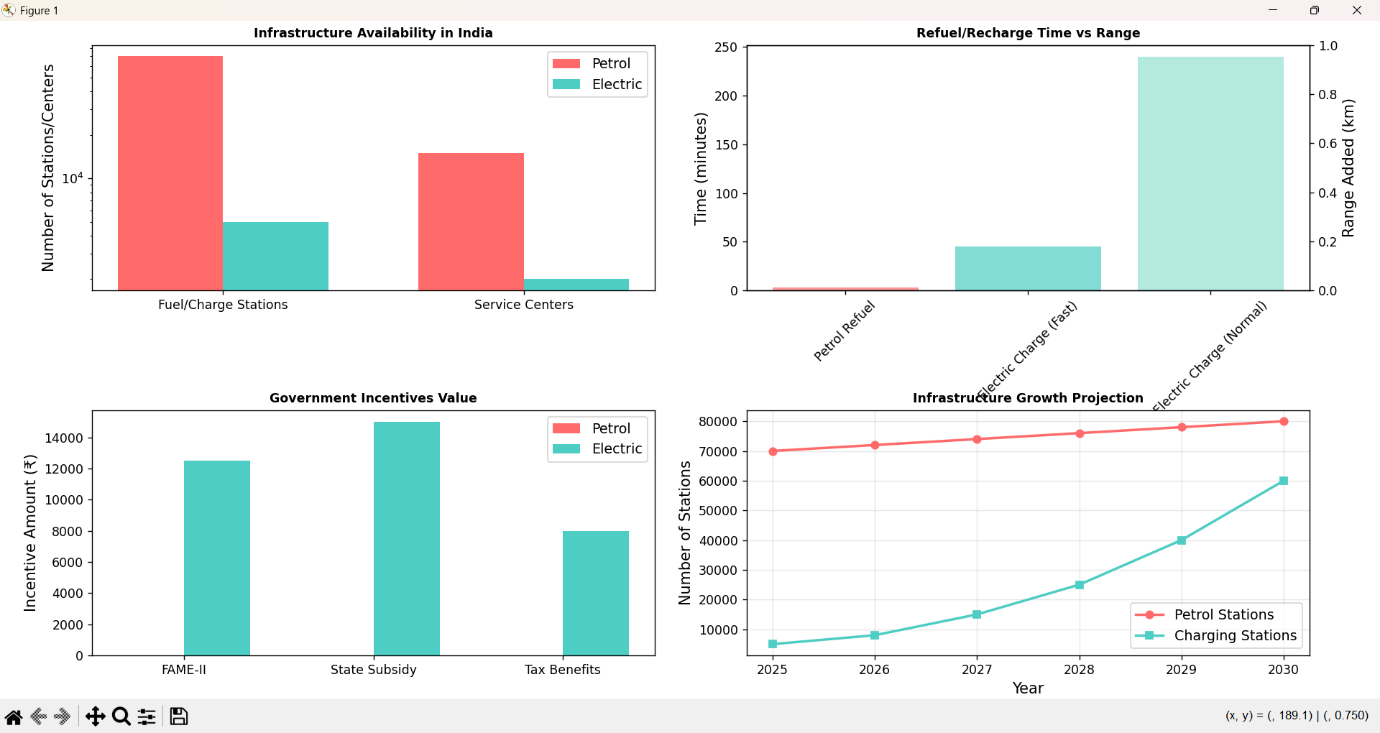
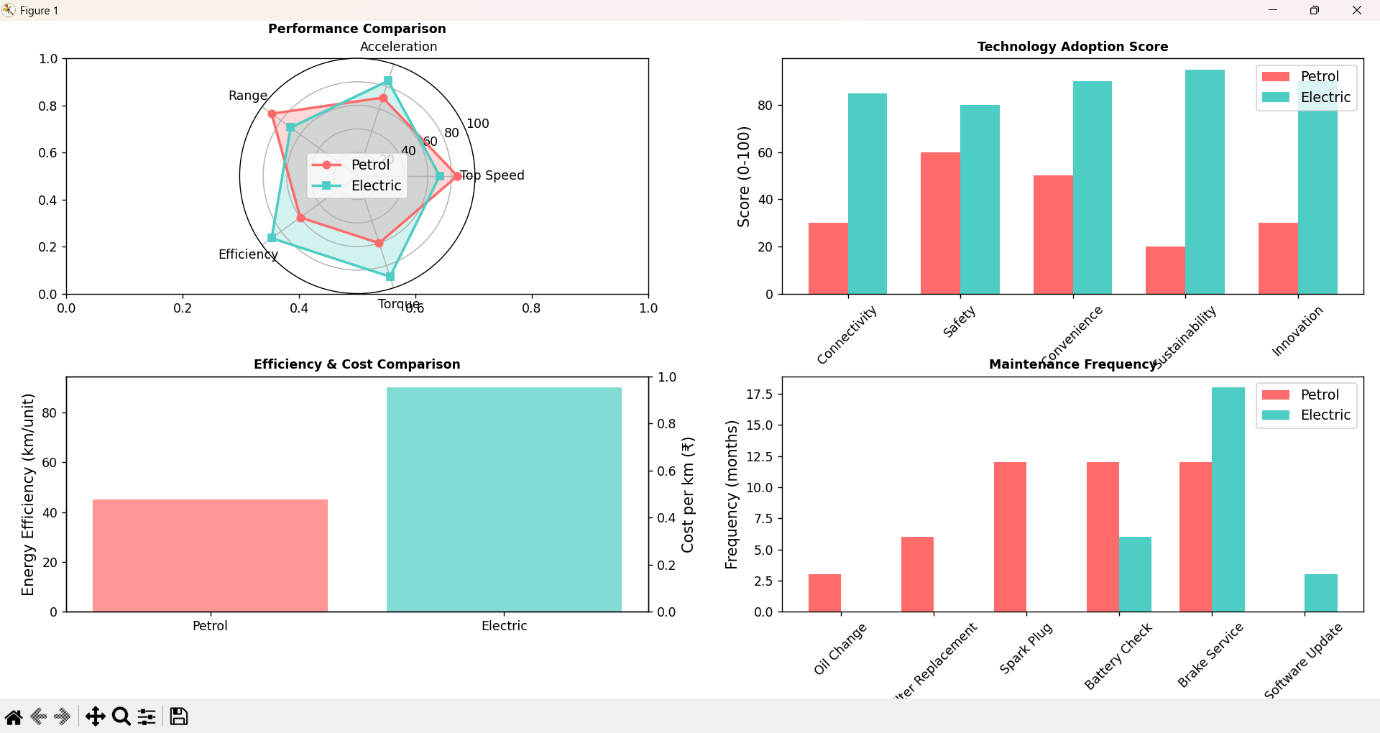
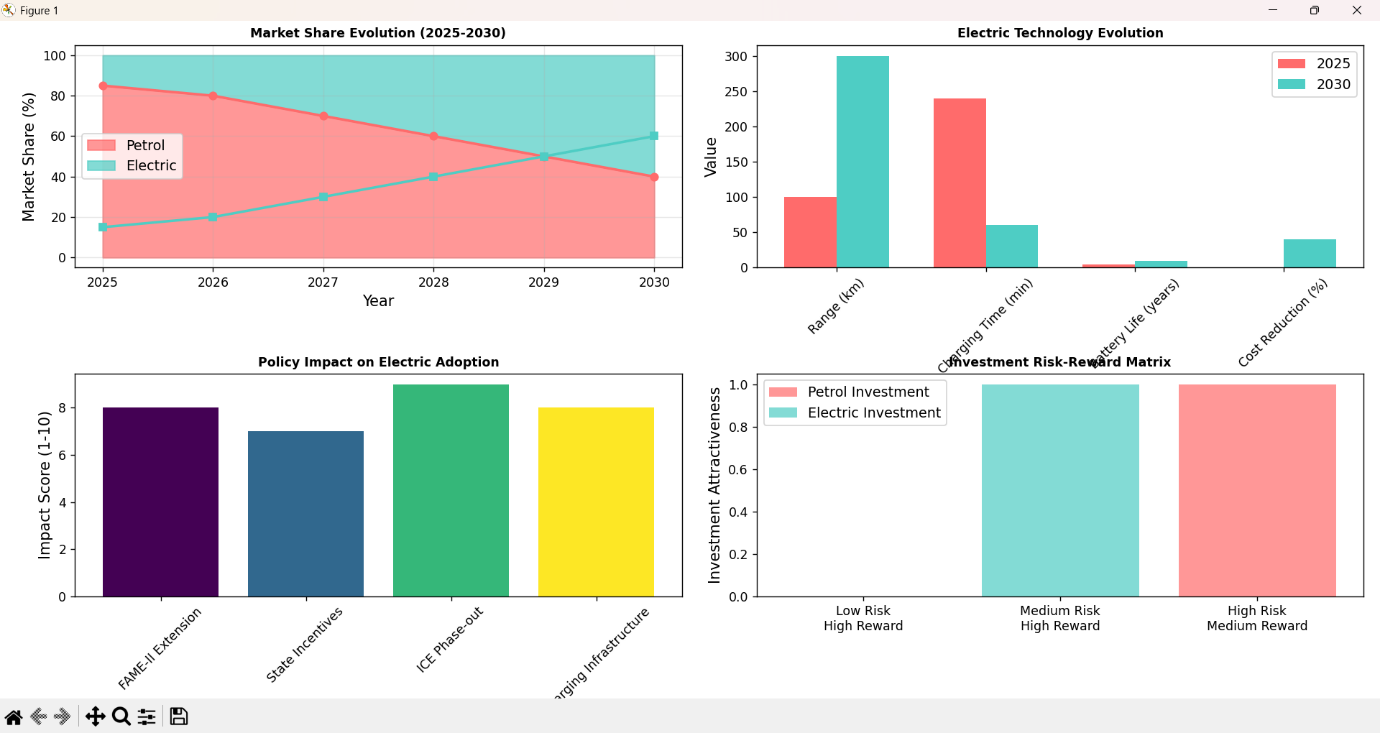
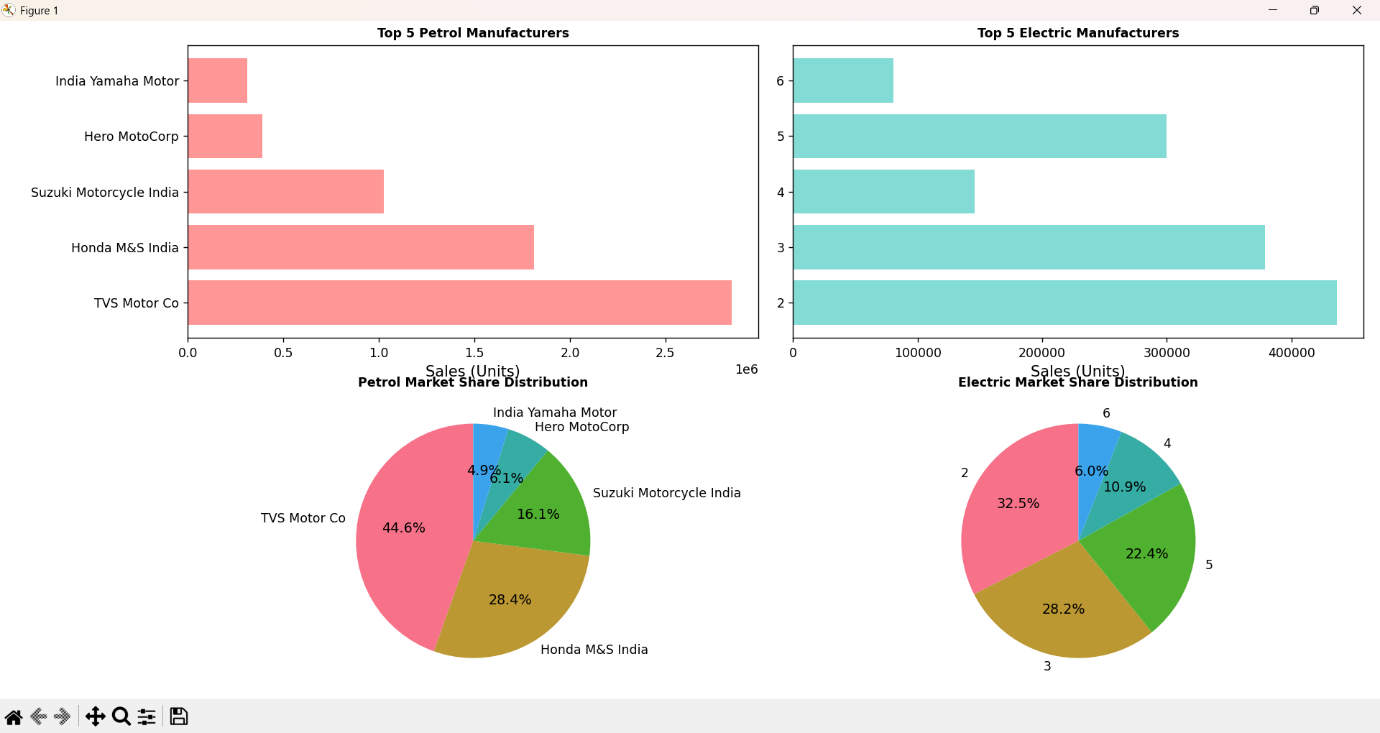
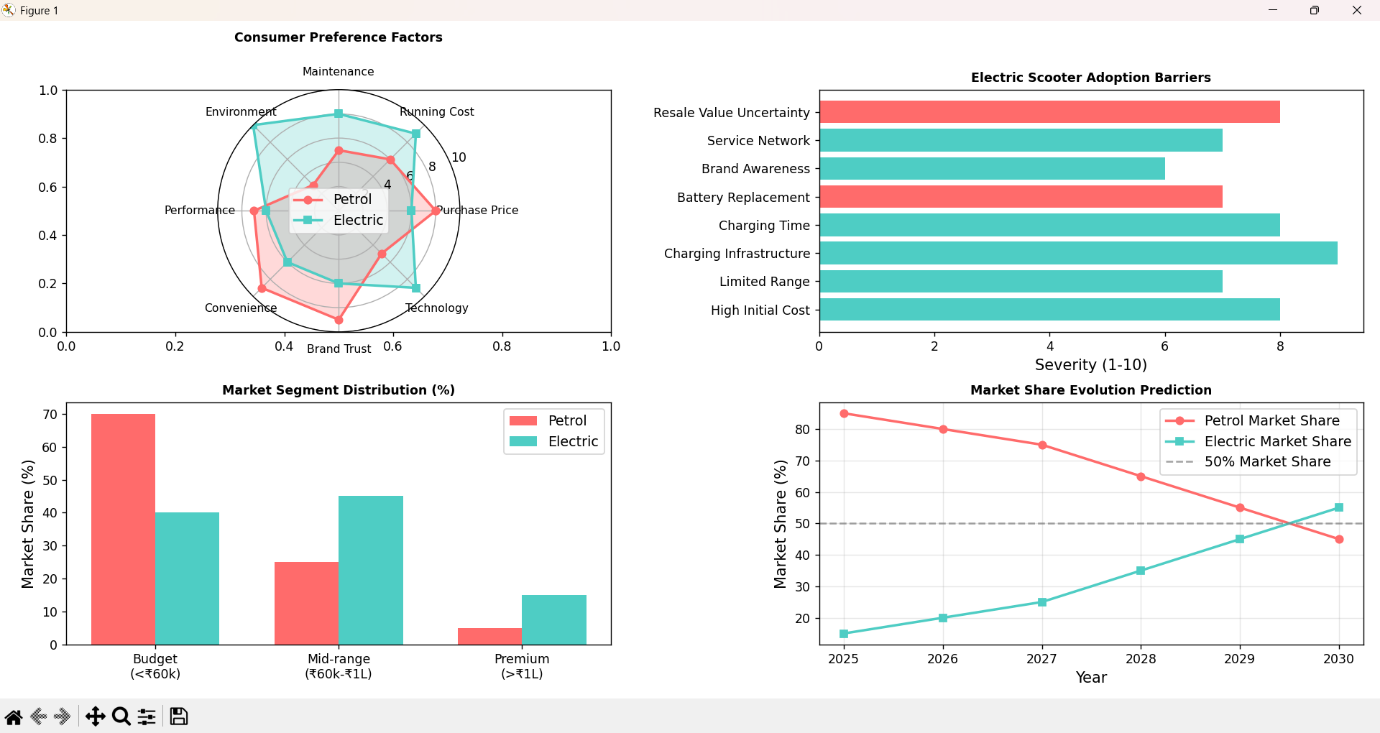
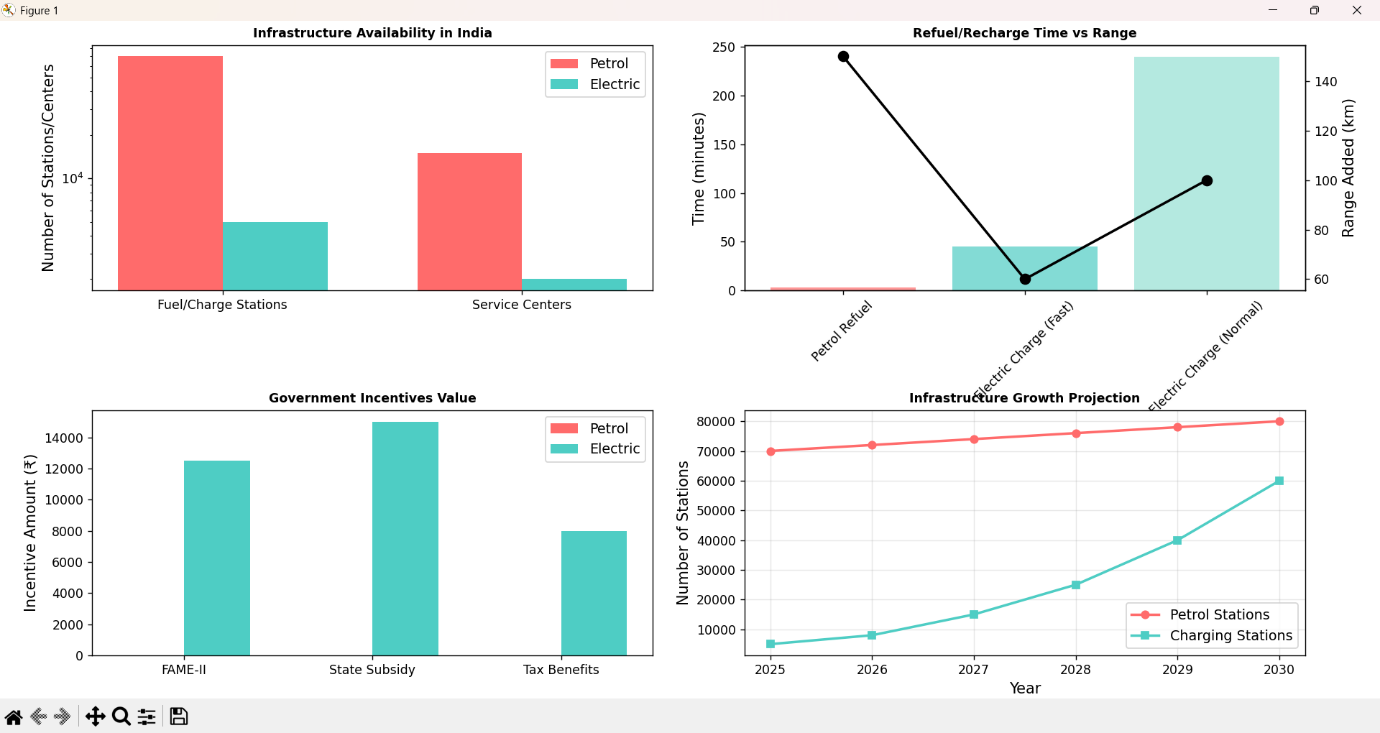
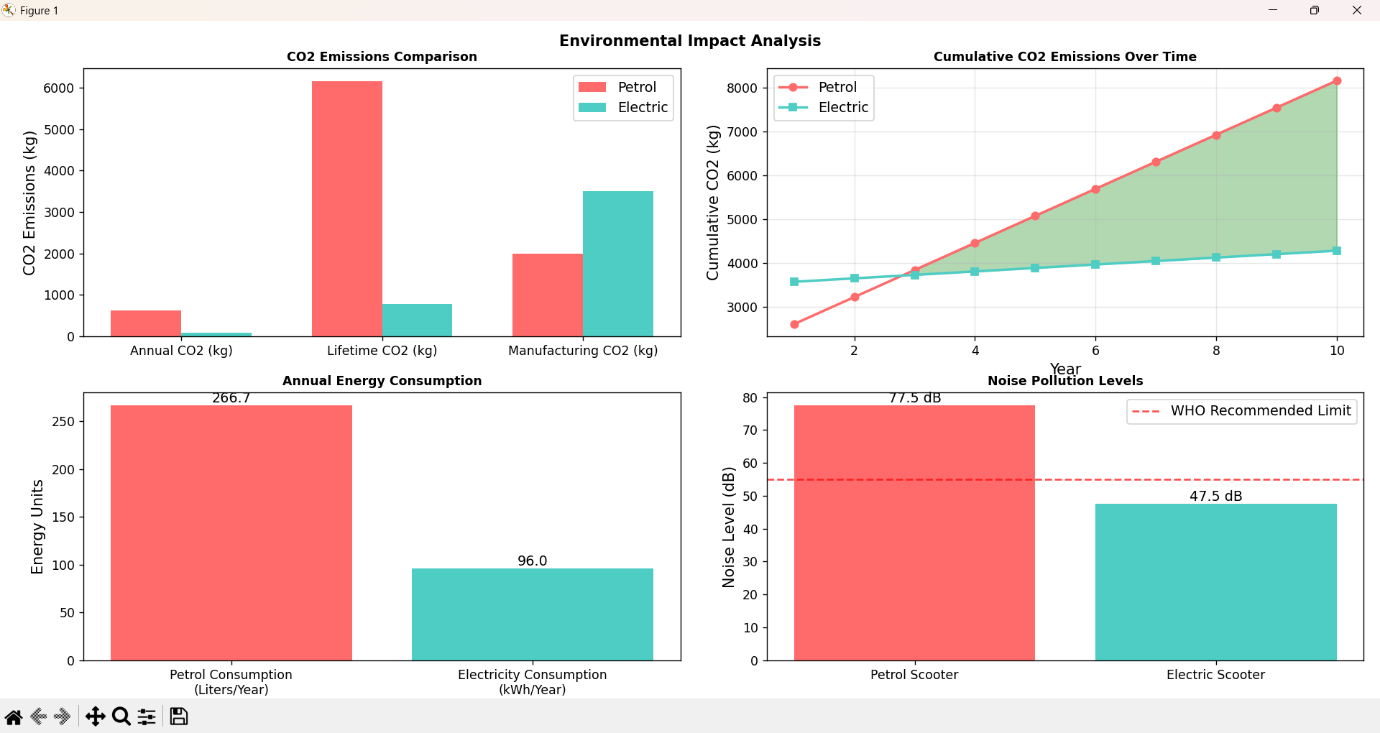
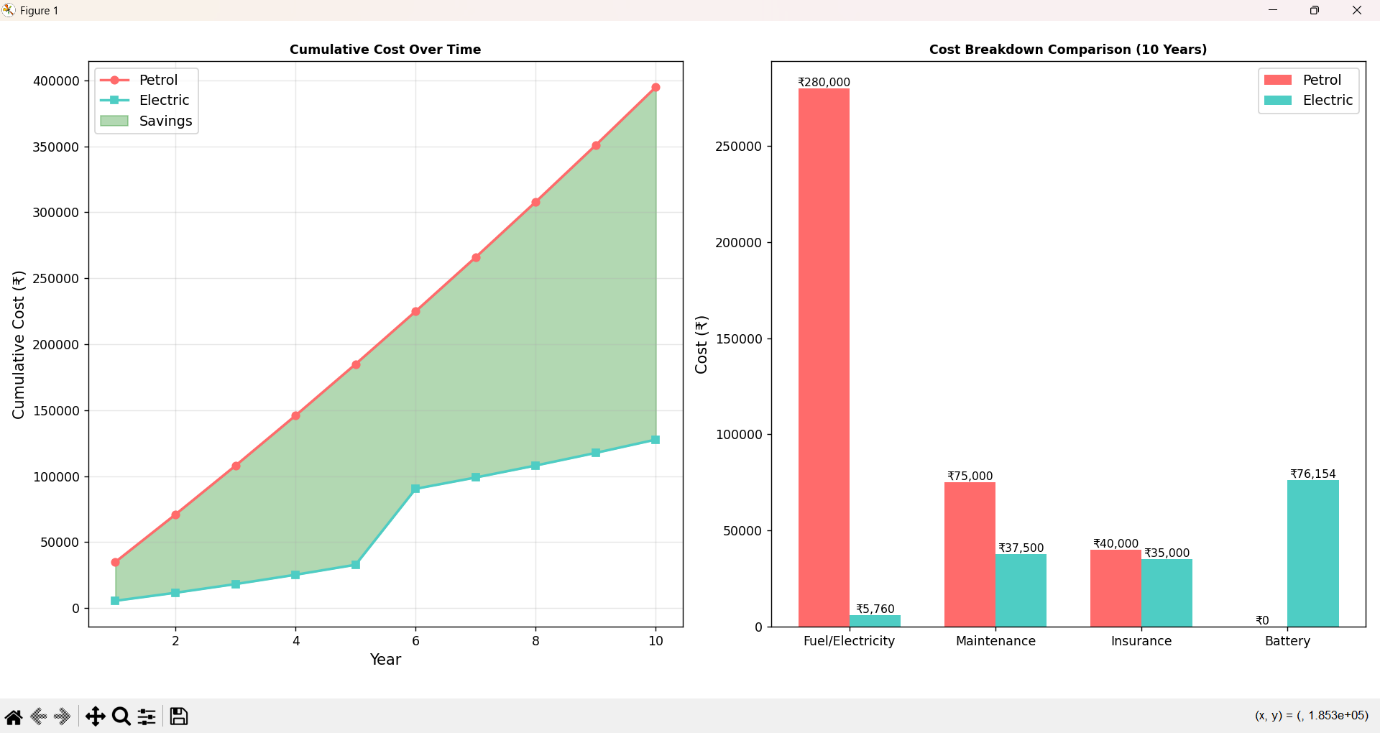
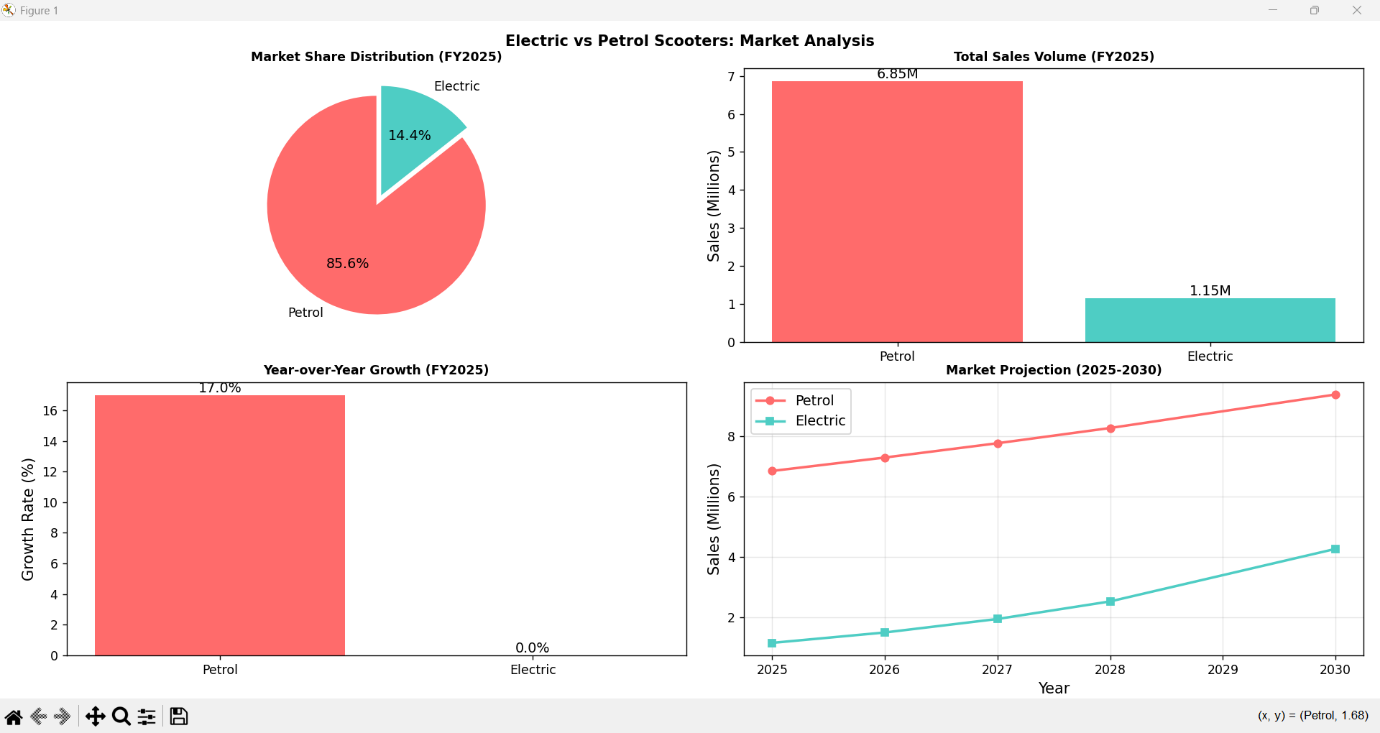
*traceback.print\_exc()*

*# Run the enhanced analysis*

*if \_\_name\_\_ == "\_\_main\_\_":*

*analyze\_petrol\_vs\_electric\_enhanced()*

**Output Screenshots**

**

**Closure**

***Findings***

* ***Electric scooters****are cheaper in the long run (₹1.5L savings over 10 years).*
* ***Petrol scooters****dominate market share (85%) but growth is slowing.*
* ***CO2 savings****with electric: ~5,000 kg over 10 years.*
* ***Adoption barriers****: High upfront cost and charging infrastructure.*

***Recommendations***

* ***Consumers****: Choose electric for city commutes (<100 km/day).*
* ***Manufacturers****: Invest in battery tech and fast-charging solutions.*
* ***Policymakers****: Expand charging infrastructure and extend subsidies.*

**Bibliography**

* ***Government Reports****: FAME-II policy, NITI Aayog EV roadmap.*
* ***Industry Data****: SIAM (Society of Indian Automobile Manufacturers),*

*Autocar Professionals articles.*

* ***Technical References****: Battery lifespan studies, CO2 emission factors.*