edunet foundation

EV Charging Demand Prediction

By: Raghav Joshi

Learning Objectives

- To understand and apply time-series forecasting techniques to predict real-world trends.
- To perform feature engineering on time-based data, creating lag features and rolling statistics to improve model accuracy.
- To build and train a Gradient Boosting regression model for predictive analysis.
- To visualize historical data and model forecasts effectively using Matplotlib.
- To deploy a machine learning model as an interactive web application using Streamlit.
- To analyze the key drivers of Electric Vehicle (EV) adoption in a specific geographical region.

Source : www.freepik.com/

Tools and Technology used

- **Programming Language:** Python (Version 3.9+)

- **Core Libraries:**

1. **Pandas:** For data manipulation and analysis.
2. **NumPy:** For numerical operations.
3. **Scikit-learn:** For building the regression model and preprocessing.
4. **Matplotlib:** For creating static data visualizations.
5. **Joblib:** For saving and loading the trained model.

- **Web Framework:**

  Streamlit: For building and deploying the interactive dashboard.

Methodology

- **Data Collection:** Utilized historical EV registration data from the Washington State Department of Licensing (2017-2024).
- **Data Preprocessing:** Cleaned the dataset, handled date-time formats, and encoded categorical features like 'County'.
- **Feature Engineering:** Created crucial time-series features to feed the model, including:

1. Lag features (e.g., EV count from 1, 2, and 3 months prior).
2. Rolling mean of EV counts over a 3-month window.
3. Percentage change and growth slope calculations.

- **Model Training:** Trained a Gradient Boosting Regressor model on the engineered features to predict the 'Electric Vehicle (EV) Total'.
- **Model Evaluation:** Assessed the model's performance using metrics like R-squared (R^2) and Mean Absolute Error (MAE) to ensure forecast reliability.
- **Model Evaluation:** Assessed the model's performance using metrics like R-squared (R^2) and Mean Absolute Error (MAE) to ensure forecast reliability.

Problem Statement:

- The rapid increase in Electric Vehicle (EV) adoption creates an urgent need for adequate charging infrastructure.
- Without accurate demand forecasting, cities and utility providers risk under-planning, leading to charging station shortages, long wait times, and a poor user experience that could slow down the transition to sustainable transport.
- This project aims to solve this by providing a reliable forecast of EV growth at a county level.

Solution:

- Developed a predictive model that accurately forecasts the number of new EVs on the road for the next 3 years.
- Created an interactive Streamlit dashboard that allows users to select any county in Washington and instantly visualize the forecasted EV adoption trend.
- The application includes a comparison feature to analyze and contrast the growth patterns of up to three different counties simultaneously.
- This tool provides actionable data for urban planners and businesses to make informed decisions about charging station placement and resource allocation.
- **GitHub Repository:** https://github.com/RAgHavj12345/EV_Charging_Prediction
- **Try the Live Application:** https://evchargingprediction-tkfusxpyhmxkvkivseemmf.streamlit.app/

Code for Loading Data

Load Dataset

In [2]:

```
# Load data
df = pd.read_csv("Electric_Vehicle_Population_By_County.csv")
```

Explore and Understand the Data

In [3]:

```
# Check Dataset Dimensions
print("Dataset Shape:", df.shape)
```

```
Dataset Shape: (20819, 10)
```

Total 20819 data points and 10 features.

In [4]:

```
# Preview the Dataset
df.head()
```

Out[4]:

	Date	County	State	Vehicle Primary Use	Battery Electric Vehicles (BEVs)	Plug-In Hybrid Electric Vehicles (PHEVs)	Electric Vehicle (EV) Total	Non-Electric Vehicle Total	Total Vehicles	Percent Electric Vehicles
0	September 30 2022	Riverside	CA	Passenger	7	0	7	460	467	1.50
1	December 31 2022	Prince William	VA	Passenger	1	2	3	188	191	1.57
2	January 31 2020	Dakota	MN	Passenger	0	1	1	32	33	3.03
3	June 30 2022	Ferry	WA	Truck	0	0	0	3,575	3,575	0.00
4	July 31 2021	Douglas	CO	Passenger	0	1	1	83	84	1.19

Code for Choosing a Model

In [24]:

```
# Define param distribution
param_dist = {
    'n_estimators': [100, 150, 200, 250],
    'max_depth': [None, 5, 10, 15],
    'min_samples_split': [2, 4, 6, 8],
    'min_samples_leaf': [1, 2, 3],
    'max_features': ['sqrt', 'log2', None]
}

# Base model
rf = RandomForestRegressor(random_state=42)

# Randomized Search
random_search = RandomizedSearchCV(
    estimator=rf,
    param_distributions=param_dist,
    n_iter=30,  # 30 random combos
    scoring='r2',
    cv=3,
    n_jobs=-1,
    verbose=1,
    random_state=42
)

# Fit model
random_search.fit(X_train, y_train)

# Best model
model = random_search.best_estimator_
print("Best Parameters:", random_search.best_params_)
```

```
Fitting 3 folds for each of 30 candidates, totalling 90 fits
Best Parameters: {'n_estimators': 200, 'min_samples_split': 4, 'min_samples_leaf': 1, 'max_features': None, 'max_depth': 15}
```

Code for Evaluating a Model

In [25]:

```
# Predict and evaluate
y_pred = model.predict(X_test)
```

In [28]:

```
def evaluate(y_true, y_pred):
    mae = mean_absolute_error(y_true, y_pred)
    rmse = np.sqrt(mean_squared_error(y_true, y_pred))
    r2Score = r2_score(y_true, y_pred)
    print(f"MAE: {mae:.2f}, RMSE: {rmse:.2f}, R2 Score: {r2Score:.2f}")

evaluate(y_test, y_pred)
```

```
MAE: 0.01, RMSE: 0.06, R2 Score: 1.00
```

Code for Forecasting Logic

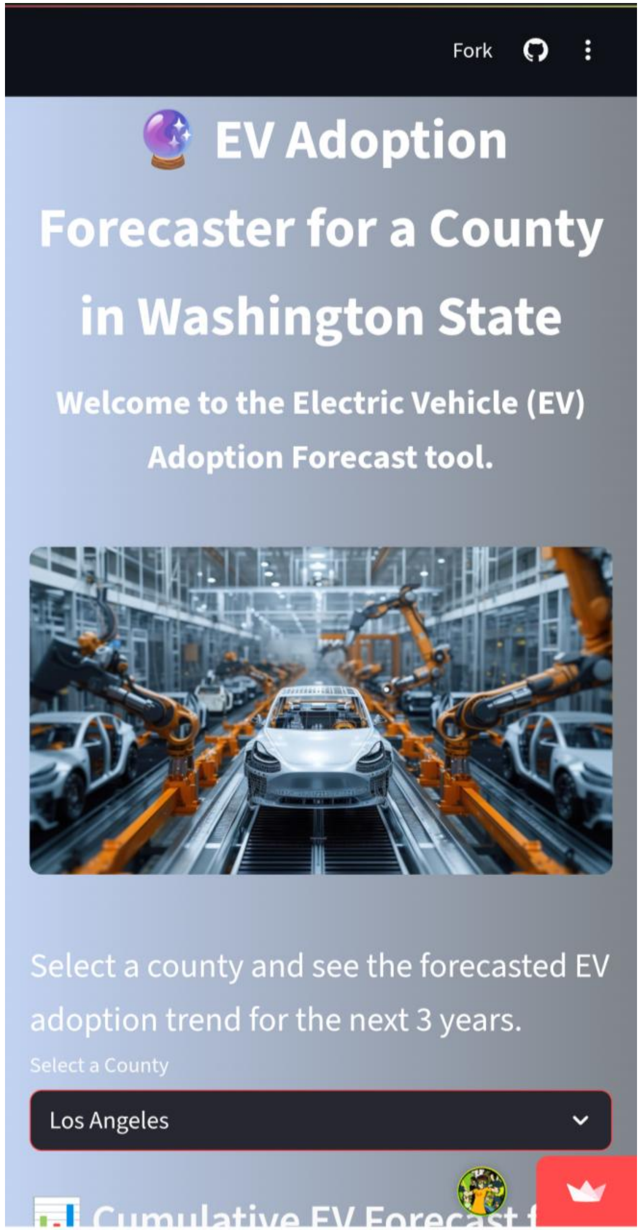
```python
# Forecasting loop from the Streamlit App
future_rows = []
forecast_horizon = 36

for i in range(1, forecast_horizon + 1):
    forecast_date = latest_date + pd.DateOffset(months=i)
    months_since_start += 1
    lag1, lag2, lag3 = historical_ev[-1], historical_ev[-2], historical_ev[-3]
    roll_mean = np.mean([lag1, lag2, lag3])
    pct_change_1 = (lag1 -  (variable) lag3: Any  = 0 else 0
    pct_change_3 = (lag1 - lag3) / lag3 if lag3 != 0 else 0
    recent_cumulative = cumulative_ev[-6:]
    ev_growth_slope = np.polyfit(range(len(recent_cumulative)), recent_cumulative, 1)[0] if len(recent_cumulative) == 6 else 0

    new_row = {
        'months_since_start': months_since_start,
        'county_encoded': county_code,
        'ev_total_lag1': lag1,
        'ev_total_lag2': lag2,
        'ev_total_lag3': lag3,
        'ev_total_roll_mean_3': roll_mean,
        'ev_total_pct_change_1': pct_change_1,
        'ev_total_pct_change_3': pct_change_3,
        'ev_growth_slope': ev_growth_slope
    }
    # Predict the next value
    pred = model.predict(pd.DataFrame([new_row]))[0]
    future_rows.append({"Date": forecast_date, "Predicted EV Total": round(pred)})
 # Update historical values for the next iteration
    historical_ev.append(pred)
    if len(historical_ev) > 6:
        historical_ev.pop(0)
```

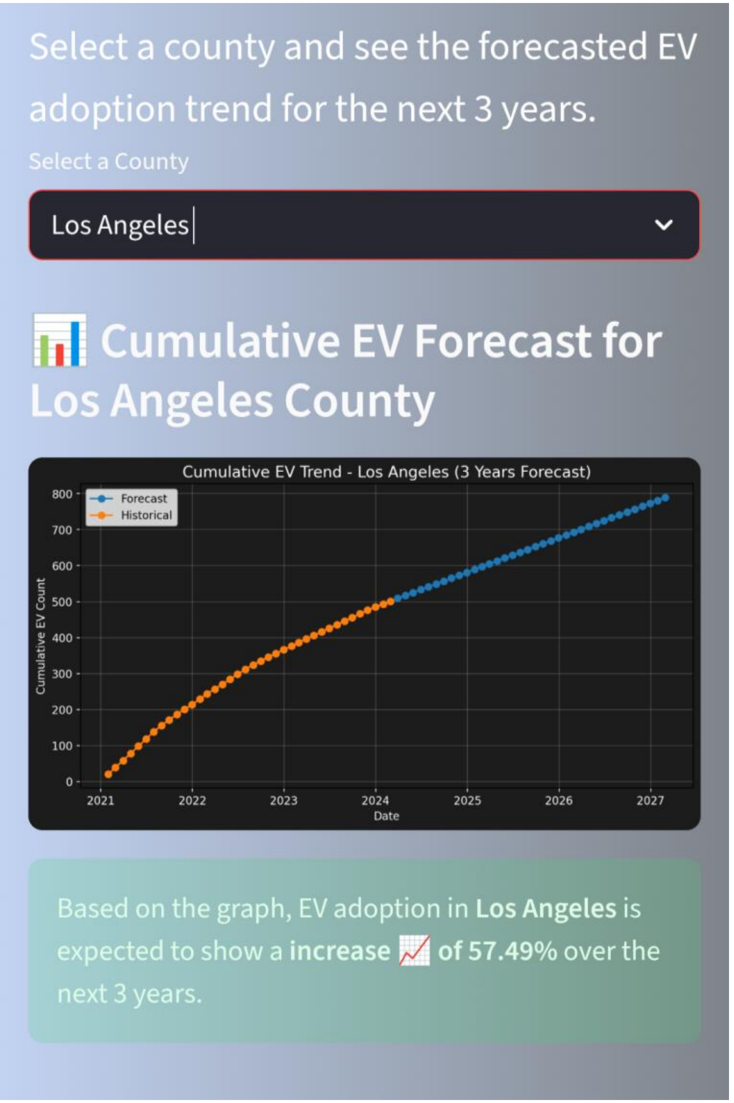
Interactive & User-Friendly Dashboard

- This is the main landing page of the deployed Streamlit application, providing a clean and professional user interface.
- The primary feature—the county selection dropdown—is immediately accessible, inviting users to interact with the model.
- This demonstrates the successful transformation of a complex model into an intuitive, user-facing product.
- Site Link: https://evchargingprediction-tkfusxpyhmxkvkivseemmf.streamlit.app/



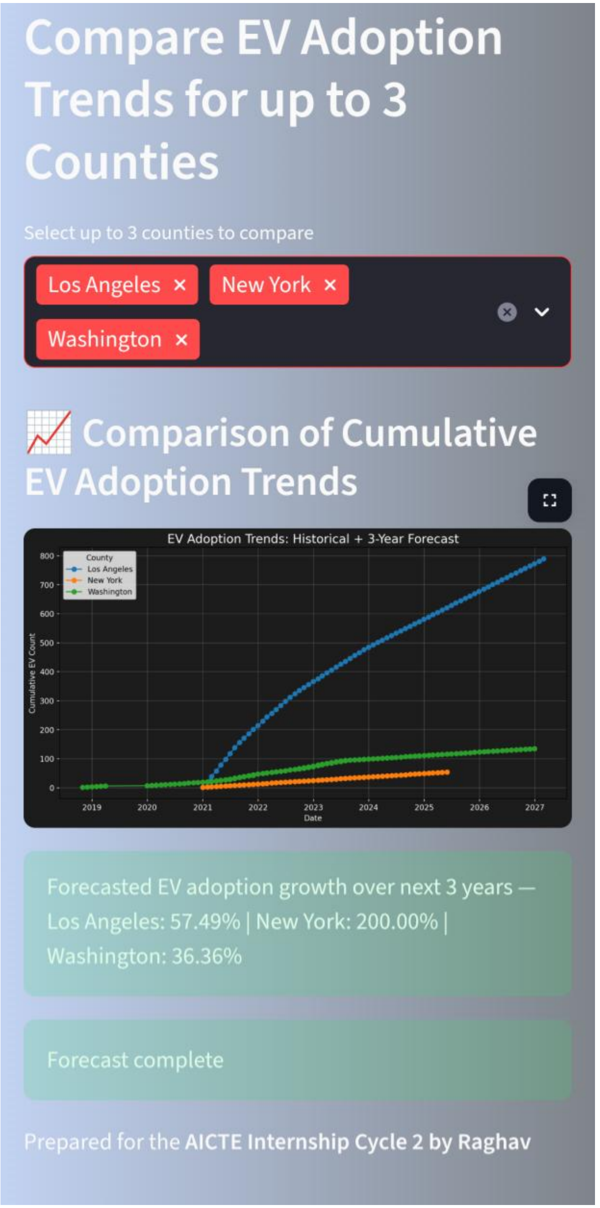
Detailed Forecast for a Single County

- After selecting a county, the application generates a 3-year forecast, visualizing the expected trend in EV adoption.
- The plot clearly distinguishes between historical data and the model's future predictions.
- A dynamic summary below the graph quantifies the forecast, providing an immediate, actionable insight (here, shows a increase of 57.49% over the next 3 years).

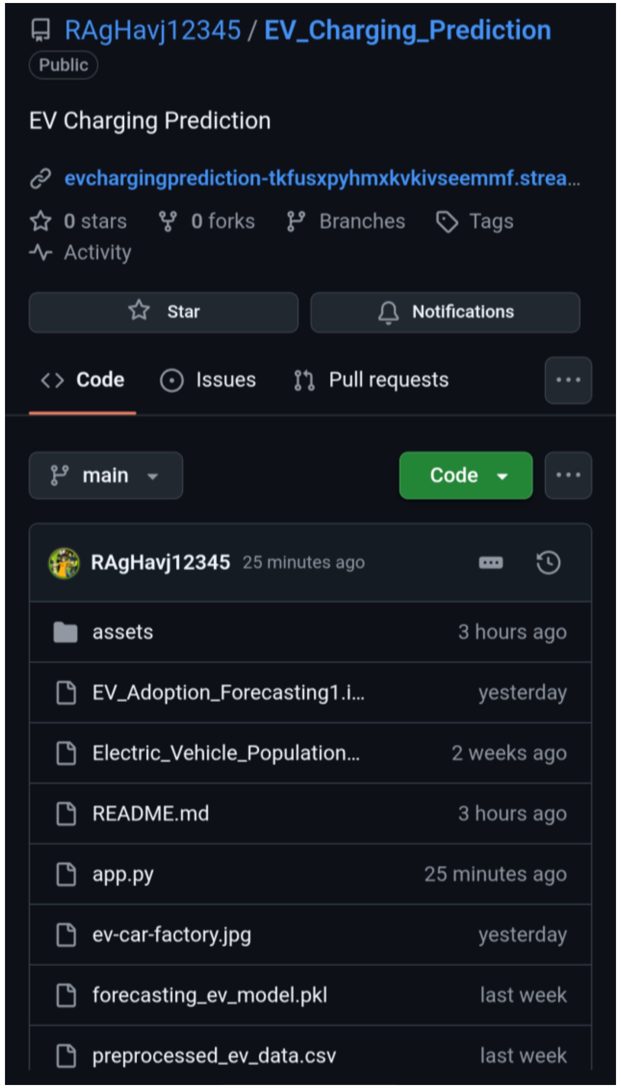


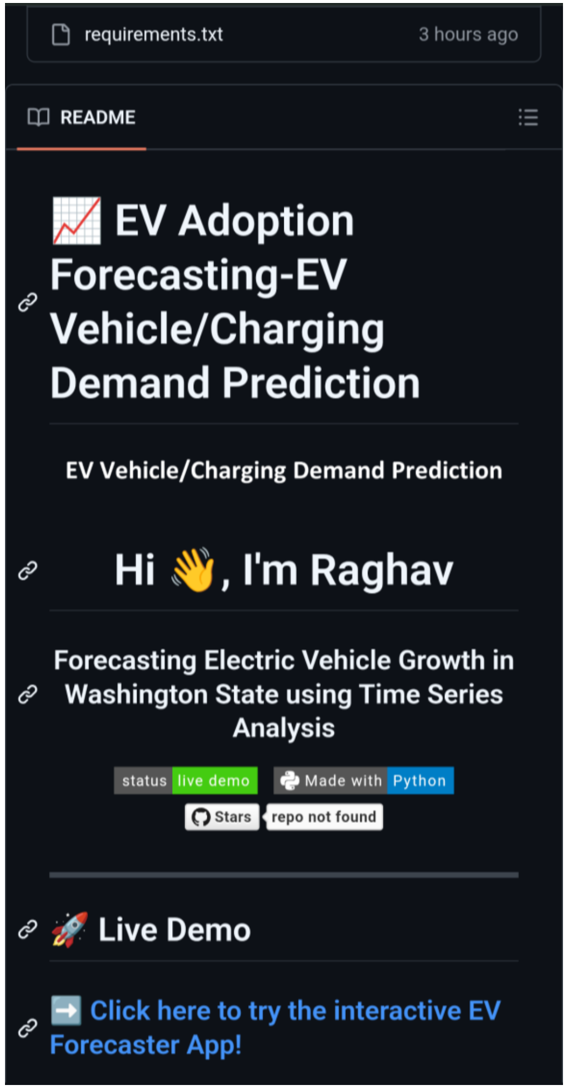
Comparative Analysis Across Regions

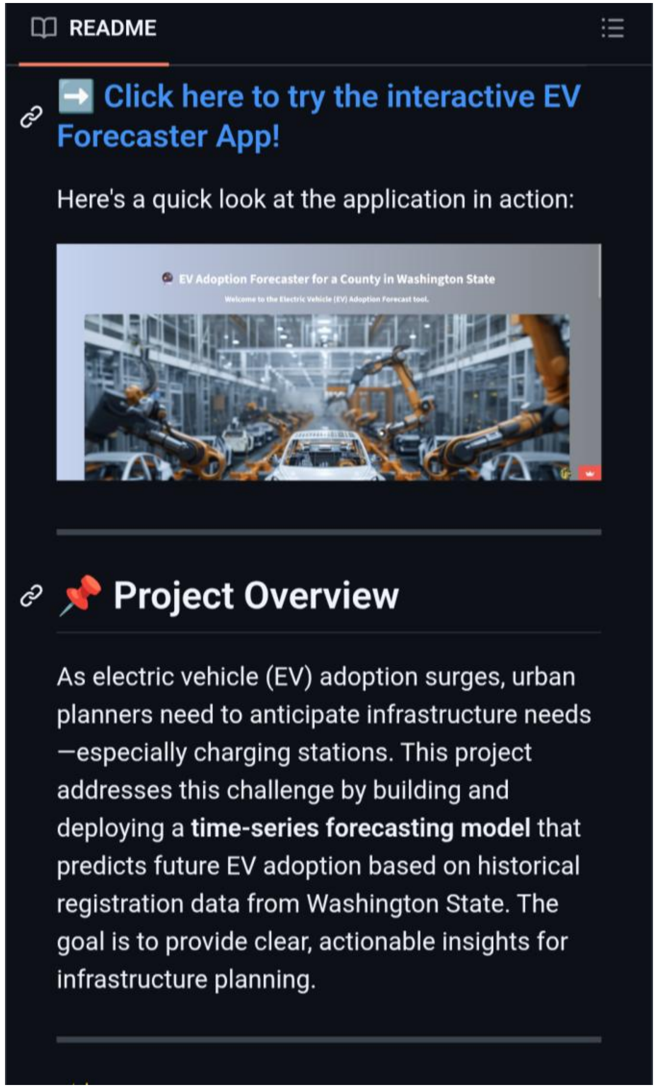
- The application includes an advanced feature to compare the EV growth trajectories of up to three different counties simultaneously.
- This allows for a high-level analysis of regional trends, highlighting which areas are leading or lagging in EV adoption.
- The summary provides a direct percentage growth comparison, which is crucial for stakeholders making decisions about resource allocation.

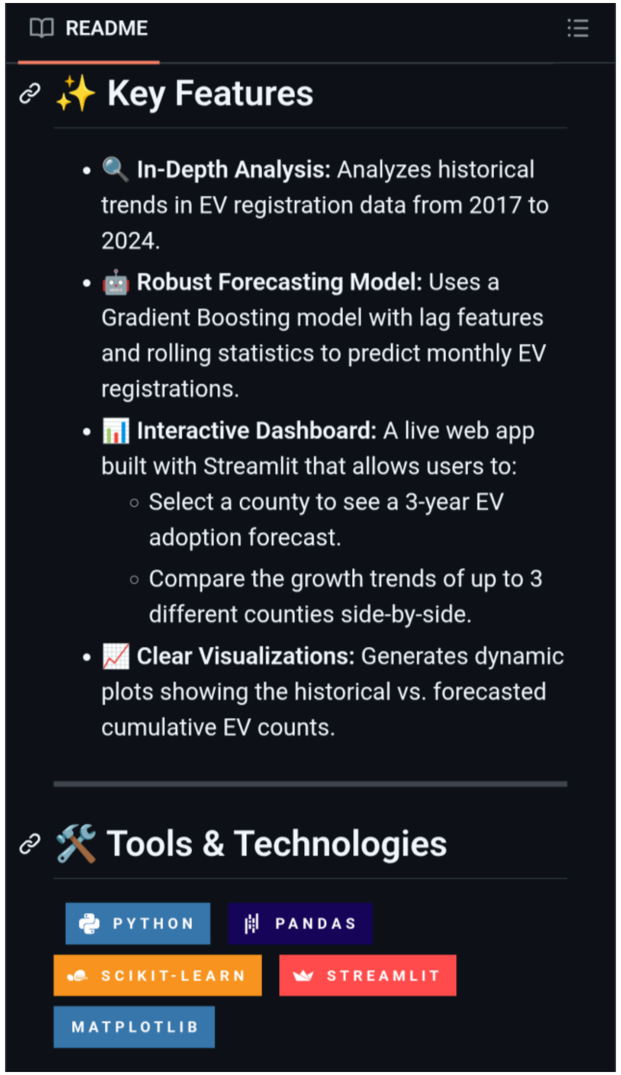


GitHub Repository Page: https://github.com/RAgHavj12345/EV_Charging_Prediction









Conclusion:

- This project successfully demonstrated that a time-series forecasting model can effectively predict EV adoption trends.
- The interactive Streamlit application makes these complex forecasts accessible and easy to understand for non-technical stakeholders.
- The model provides valuable insights that can help Washington State prepare for its electric future.
- Through this project, I gained practical experience in feature engineering, model deployment, and building end-to-end data science solutions.

Future Scope:

- Enhance model accuracy by incorporating external data sources, such as gas prices, government incentives, or local population growth.
- Experiment with more advanced forecasting models like LSTMs (a type of neural network) to potentially capture more complex, non-linear patterns in the data.