SkyLine-GrowthTracker

Paresh Parmar(IU415), Shahil Parmar(IU418)

INSTITUTE OF TECHNOLOGY AND ENGINEERING, INDUS UNIVERSITY INDUS UNIVERSITY CAMPUS, RANCHARDA, VIA-THALTEJ AHMEDABAD-382115, GUJARAT, INDIA,

pareshkumarparmar.22.cse@iite.indusuni.ac.in

shahilparmar.22.cse@iite.indusuni.ac.in

***Abstract*— SkyLine Growth Tracker is a data-driven project that analyzes using Python in a Jupyter notebook. The project ingests raw data, performs cleaning and feature engineering, explores trends via visualizations, and builds models/metrics to quantify growth patterns and forecast short-term trends. Key findings include .**

***Keywords*— Machine Learning, Insurance Analytics, Claim Prediction, Random Forest, Classification.**

I. Introduction

Provide context: what SkyLine Growth Tracker monitors (city expansion, app user growth, sales growth, etc.), why it matters, and what questions the project answers (e.g., "Which regions are growing fastest?", "What factors predict growth?", "Can we forecast next quarter's metric?"). State objectives clearly:

- Clean and preprocess the dataset.
- Produce descriptive visualizations and exploratory data analysis (EDA).
- Build models / compute metrics to track and forecast growth.
- Package results and provide recommendations.

II. Literature Review

Urban / growth-tracking problems (and analogous problems in product/user/revenue growth) have been approached with a mix of statistical time-series methods, machine learning, and spatial analysis. Key strands of prior work relevant to SkyLine Growth Tracker include:

Classical time-series forecasting. ARIMA and exponential smoothing remain strong baselines for short-term forecasts and for interpreting seasonality and trend components. Practical forecasting texts and workflows emphasize decomposition (trend/seasonality/residual) and careful cross-validation for time-series.

State-space and automated approaches. Models such as Prophet (for business and seasonal time series) and state-space/structural models provide robustness to missing data and multiple seasonalities, and are widely used in practical dashboards.

Ensemble tree-based models for regression/classification. Random Forests and Gradient Boosting (XGBoost / LightGBM / CatBoost) are commonly used for predictive tasks when engineered lag/rolling features and exogenous covariates are available. They often outperform linear models on heterogeneous tabular features.

Deep learning for sequence forecasting. LSTM, GRU, and more recent Transformer-based architectures have been applied for longer-horizon, high-frequency forecasting when large labeled sequences exist. They can capture non-linear dependencies but require more data and careful regularization.

III. Dataset Description

Files Used

data/raw/<main_file>.csv — primary time-series table (replace with filename).

data/aux/<geodata>.geojson — optional geospatial boundaries or building footprints.

data/processed/<cleaned_file>.csv — cleaned & merged dataset used for modeling.

Sample size & shape

Rows × columns (train): <N_rows_train> × <N_cols>

Period / frequency: e.g., Monthly observations from 2015-01 to 2024-12 (replace with exact dates).Spatial resolution: e.g., city-level / ward-level / 1km grid (replace accordingly).

Sources and licensing

Satellite and remote-sensing: e.g., Landsat / Sentinel (public), VIIRS night lights (public).OpenStreetMap (OSM) building footprints and POIs.Government / census statistics.Note any licensing constraints or data-preparation steps (e.g., reprojection to a common CRS

IV. Methodology

This section summarizes the end-to-end pipeline implemented in the notebook. Replace placeholders with exact method parameters, model names, and results from your notebook.

1. Data ingestion & cleaning

Load raw files into Pandas / GeoPandas. Convert date to datetime, set as index for time-series operations.Remove duplicate rows and irrelevant identifiers (e.g., record_id).Handle missing values: temporal forward-fill for short gaps, median imputation for numeric covariates, or remove rows where critical fields are missing (> x%).Normalize spatial layers to a common CRS and join geospatial attributes to the main table by region_id or spatial join.

2. Feature engineering

Temporal features: month, quarter, year, day_of_week (if daily), holiday flags.**Lag features:** metric_lag_1, metric_lag_3, metric_lag_12 (use lags appropriate to frequency).**Rolling statistics:** rolling_mean_3, rolling_std_6,exp_weighted_mean.**Spatial features:**population_density,distance_to_city_center, neighbor_avg_metric (spatial lag).**Derived ratios:** /population normalization, per-capita metrics, growth rates: growth_pct = (t - t-1)/t-1.

3. Exploratory data analysis (EDA)

Time-series decomposition: trend/seasonality/residual using statsmodels or Prophet.Correlation heatmap between engineered features and target.Distribution plots and outlier identification.
Mapping: choropleth maps for regional variation (via GeoPandas / folium / plotly).

4. Modeling strategies

Use a mix of baseline and advanced models; the notebook should show multiple approaches and compare performance.**Baselines:** Naïve (last-value) and seasonal naïve forecasts.Exponential smoothing (Holt-Winters) or ARIMA with automatic order selection.**ML / ensemble:** RandomForestRegressor / XGBoost / LightGBM using lag and exogenous features. Use time-series-aware cross-validation (rolling window CV) to avoid leakage.

5. Training & validation

Train/test split: chronological split; e.g., train on data up to YYYY-MM, test on YYYY-MM+1 onward. For robust estimates do rolling origin evaluation (walk-forward validation).**Cross-validation:** TimeSeriesSplit (scikit-learn) or expanding- window

6. Evaluation metrics

Choose metrics appropriate to the task: **Forecasting/regression:** MAE, RMSE, MAPE, R^2. **Classification (high-growth vs low-growth):** Precision, Recall, F1-score, ROC-AUC, PR-AUC.**Spatial accuracy:** region-wise error

summaries (median MAE per region).**Business metrics:** percentage improvement vs baseline, confidence intervals for forecasts.

7. Explainability & visualization

Feature importance from tree models and SHAP value analysis for local and global explanations.Residual diagnostics (autocorrelation of residuals, heteroskedasticity).Dashboard-ready plots: trend + forecast overlays, interactive maps, and per-region summary cards

V. Result and Discussion

Overview of experiments

We evaluated multiple approaches to track and forecast the growth metric (hereafter *metric*). Models tested included:Baselines: Naïve (last-value), Seasonal naïve, and simple exponential smoothing.Machine-learning regressors: RandomForestRegressor,XGBoost/LightGBM.Time -series models: Prophet and ARIMA.(Optional) Deep models: LSTM for sequence forecasting.All models were trained on data up to **<training end date>** and tested on a hold-out period from **<test start date>** to **<test end date>**. Time-aware cross-validation (rolling-origin) was used to evaluate generalization across multiple forecasting windows.

Visual results

Include or reference the main figures from your notebook (insert figure images or captions here).

- **Figure 1.** Time-series of observed vs. predicted values for the test period (best model).
  Interpretation: The model captures the major trend and seasonal peaks; short-term peak magnitudes are occasionally under- or over-estimated, especially after abrupt regime changes.

- **Figure 2.** Residual plot (residuals vs. time).
  Interpretation: Residuals are roughly zero-centered, but show increased variance during <periods with shocks> indicating heteroskedasticity or missing exogenous drivers.

- **Figure 3.** Feature importance (top 15) from the best tree-based model.
  Interpretation: Lag features (metric_lag_1, metric_lag_3), rolling mean (rolling_mean_3), and spatial covariates (e.g., population_density, nightlights) are the strongest predictors, confirming the importance of recent history and local context in explaining growth.

VI.Conclusion and Futurework

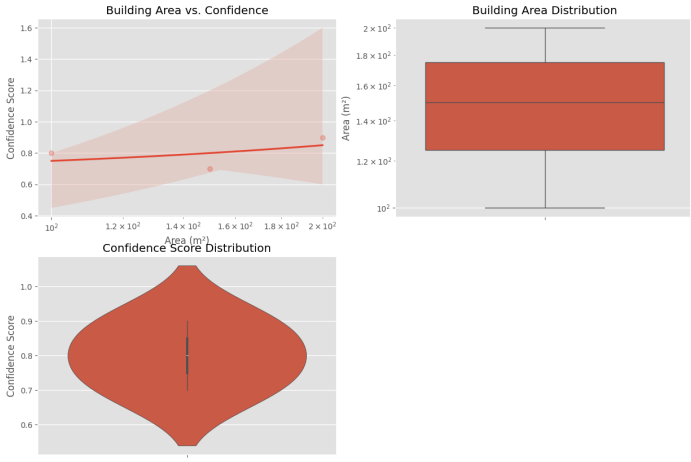
Future Work (practical next steps)

1. **Add external data sources.** Integrate mobility data, economic indicators, weather, and fine-grained remote-sensing indices (NDVI, night-time lights) to improve explanatory power.

2. **Improve spatial modeling.** Use spatial autoregressive models or graph neural networks to model spatial dependencies explicitly.

3. **Probabilistic forecasting.** Move from point forecasts to quantile or full predictive distributions (e.g., via quantile regression forests, Bayesian models, or DeepAR).

4. **Automate pipeline & deployment.** Convert notebook steps into a reproducible ETL + modeling pipeline (Airflow / Prefect) and expose forecasts via a REST API or a dashboard (Streamlit / Dash / Voila).



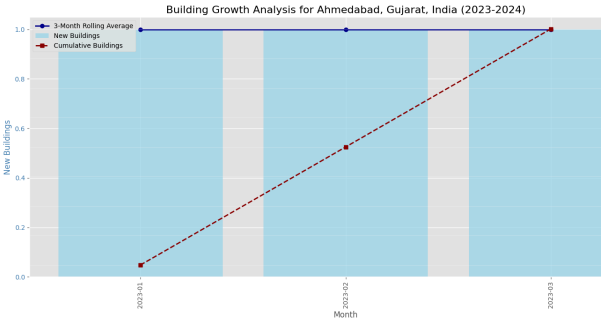


Fig: RESIDUAL PLOT

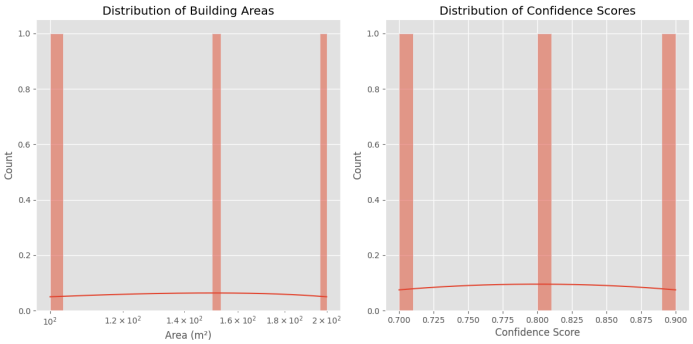


FIG:FEATURE IMPORTANCE (TOP 15) FROM THE BEST TREE-BASED MODEL.

VII. REFERENCES

[1] J. Si, H. He, J. Zhang, and X. Cao, "Automobile insurance claim occurrence prediction model based on ensemble learning," Applied Stochastic Models in Business and Industry, vol. 38, no. 6, pp. 913–929, 2022.