

Time Series Forecasting of User Engagement in Python Software Applications Based on Usage Logs

Janarthan V. ^{1*}, Erandi J.D.T. ² and Tharaka Y.M.S. ¹

¹Department of Software Engineering, Faculty of Computing,
Sabaragamuwa University of Sri Lanka, Sri Lanka

²Department of Data Science, Faculty of Computing,
Sabaragamuwa University of Sri Lanka, Sri Lanka

*janarthanvijayakumar@gmail.com

Understanding users' engagement in Python-based software applications is important for improving usability, maintaining user satisfaction, and strategic future planning. Python usage logs contain important but rarely used variables like timestamps, functional interaction, session and download counts to analyze user engagement. However, existing traditional analysis mainly supports statistical analysis rather than capturing temporal dependencies, irregular usage patterns, and domain specific variables like feature updates or version releases and these studies mainly focused on single applications like Instagram, Facebook and TikTok but not considered as Python-based applications. Therefore, research addresses this gap by developing a time series forecasting framework with use of Python application usage logs. This research study analyzes and compares three types of forecasting model like traditional statistical models (ARIMA, SARIMA and Prophet), deep learning models (LSTM and GRU), and hybrid models (Prophet-LSTM, SARIMAX-GRU and Prophet-ElasticNet), guided by research questions assessing by evaluating their forecasting performance and testing under different use cases, such as sparse logs and bursty activity, to examine their robustness and practical suitability for Python application forecasting. The study started with literature reviewing of limits of existing forecasting models, limits of features and application types, and fewer studies on Python applications. Continuing with accurate methodology including data analysis preprocessing, feature engineering, models development, evaluation (MAE, RMSE, and MAPE) and testing and comparison of models. This study involves a Python native forecasting tool developed with Django, React, REST-API, TensorFlow, Prophet and Python libraries with developed model artifacts. Findings explain that hybrid models got high performance (MAPE are 6.35 than both statistical and deep learning models particularly under the sparse logs and bursty conditions. The research concludes that irregular features and hybrid residual prediction methods gave high forecast performance and offered valuable insights for Python developers and product teams.

Keywords: *Time Series, Engagement Forecasting, Prophet-LSTM, Python Software Applications, Usage Logs*