Predicting Monthly Total Revenue for Business

Business Problem

Accurate prediction of monthly billing amounts is crucial for financial planning and operational efficiency in large enterprises. This study aims to develop a predictive model using the monthly billed revenue dataset. There can be various Statistical model (ARIMA) and Machine Learning models (Random Forest, Gradient Boosting Machines, Support Vector Machines) that can fit to a financial dataset. Performance of each model should be measured and we will carefully determine which model performs best given the monthly billed revenue dataset. Since the size of the dataset is relatively small (Less than 1,000 observations), a cross-validation will be needed to avoid overfitting issues. The best model should be selected to forecast monthly billing amounts, thereby aiding in better resource allocation, budgeting, and strategic planning.

Background/History

Predicting billing amounts has traditionally involved simple statistical methods such as moving averages, linear regression, and exponential smoothing. While these methods are straightforward and easy to implement, they often fail to capture the complex patterns and interactions present in the data. Specifically, these traditional approaches struggle with:

- **Seasonality and Cyclic Patterns**: Traditional methods may not effectively capture seasonal variations and cyclic patterns in billing amounts.
- **Non-Linearity**: Billing data often exhibits non-linear relationships between different factors, which simple linear models cannot handle.
- **Interaction Effects**: Traditional models typically do not account for interactions between multiple variables, leading to oversimplified predictions.

Recent advancements in machine learning, particularly ensemble methods like GBM, offer more robust solutions by addressing these limitations. This study leverages these advancements to improve the accuracy of billing predictions.

Data Explanation

Datasets

The data is sourced from various billing systems within the company and stored in a Teradata database. They are all processed and interfaced to the ERP system. Later, processed billing

and adjustment data are replicated to the Teradata analytics solution DB. Dataengineers further process the data and produce one fact table. The scope of this project is limited to billing revenue data. So the dataset only contains billing information from February 2023 to July 2024.

Data Dictionary

- **report_month:** accounting period in date when the bills are reported
- **billing_year:** accounting year when the bills are reported
- **billing_month:** accounting month when the bills are reported
- **billing_cycle:** accounting day (1 - 31) when the bills are reported
- **billing_day:** accounting work day (excluding weekends) when the bills are reported
- **billing_amount :** billed amount booked in General Ledger

Data Preparation

The billing cycle runs from the 26th of the previous month to the 25th of the current month. Data preprocessing involved handling missing values, feature engineering, and splitting the data into training, validation, and test sets.

Feature Engineering

To enhance predictive power, the following features were engineered:

1. Moving Averages:
    - moving_avg_7 (7-day moving average): This feature helps to smooth out short-term fluctuations and highlight longer-term trends in the billing amounts. It provides the model with information about the average billing amount over the past week, which can be useful for capturing short-term trends and seasonality.
    - moving_avg_14 (14-day moving average): Similar to the 7-day moving average but over a longer period. This helps capture trends over a fortnight, which can be useful for identifying bi-weekly patterns in billing amounts.
    - moving_avg_30 (30-day moving average): This feature smooths the data over a month, helping to identify long-term trends and seasonal effects. It provides context on how the current billing amount compares to the monthly average.
2. Lag Features:
    - lag_1 (1-day lag): The billing amount from the previous day. This feature helps the model capture day-to-day dependencies and immediate past trends, which are often crucial in time series forecasting.
    - lag_7 (7-day lag): The billing amount from a week ago. This feature helps capture weekly seasonality and trends. Many businesses exhibit weekly patterns, making this a valuable feature.
    - lag_14 (14-day lag): The billing amount from two weeks ago. This feature helps identify bi-weekly patterns and dependencies.

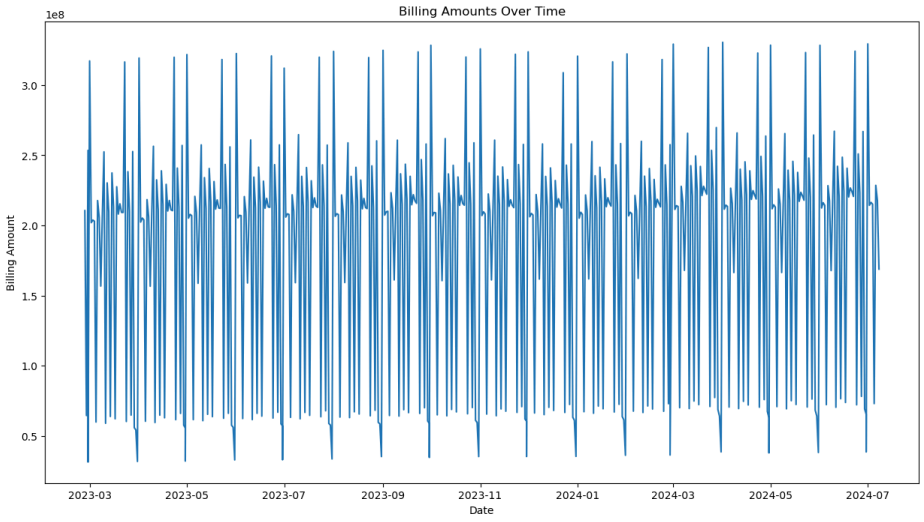
- lag_30 (30-day lag): The billing amount from a month ago. This feature helps capture monthly seasonality and long-term trends, which are particularly important in financial data where monthly cycles are common.

3. Date-based Features:
    - day_of_week: The day of the week (0 = Monday, 6 = Sunday). This feature captures weekly patterns in billing amounts. For example, billing activity might be higher on weekdays and lower on weekends.
    - is_weekend: An indicator if the day is a weekend (1 if Saturday or Sunday, 0 otherwise). This binary feature simplifies the model's ability to identify weekends, which can significantly impact billing patterns.
    - day_of_year: The day of the year. This feature captures seasonal effects and trends throughout the year, such as holidays, end-of-quarter effects, or annual business cycles.
    - week_of_year: The week of the year. Similar to day_of_year, this feature captures weekly seasonal effects and trends, helping the model understand the position within the annual cycle.

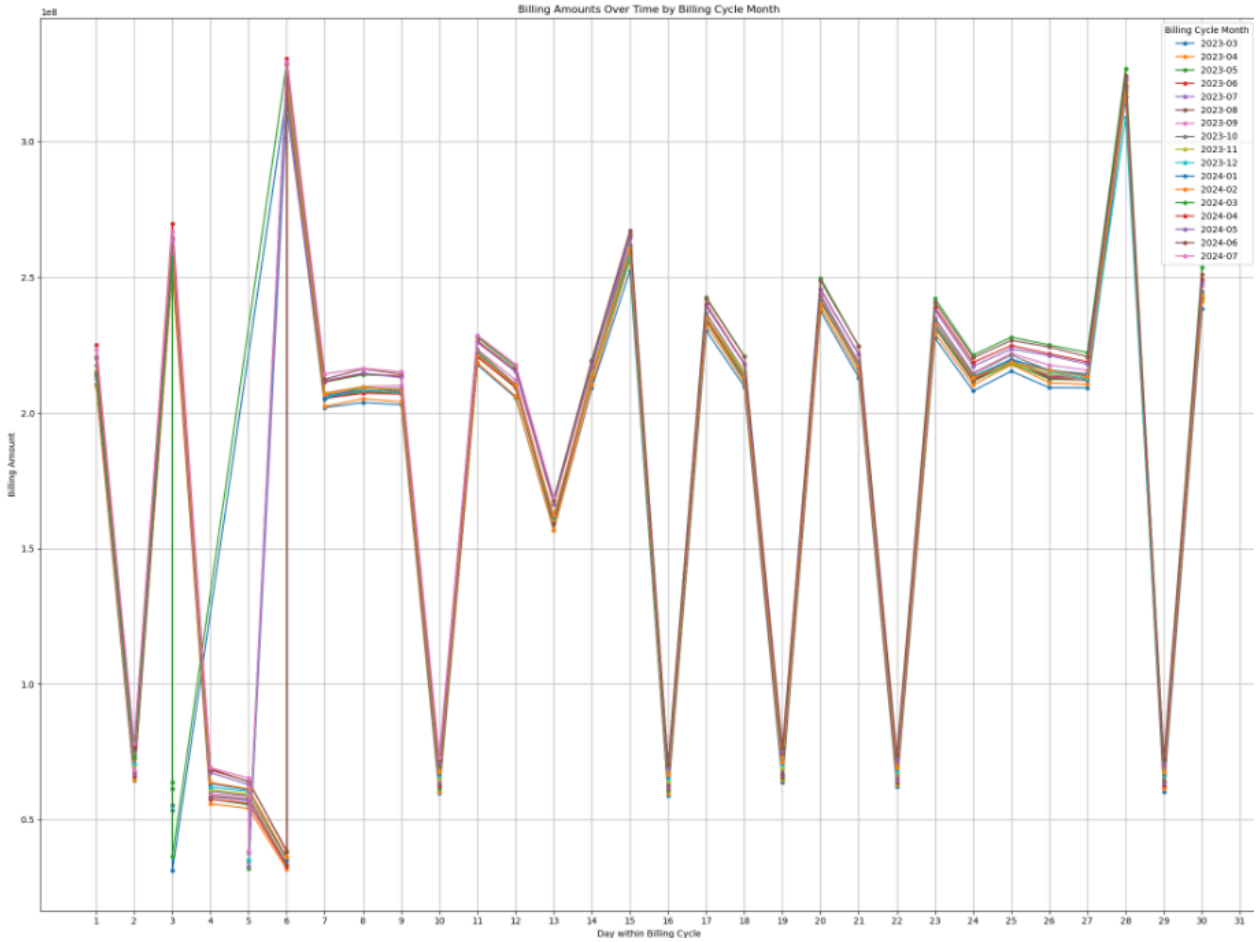
Moving averages and lag features help the model understand underlying trends and patterns in the billing data. This is crucial for making accurate predictions, especially in the presence of noise and short-term fluctuations. Date-based features like day_of_week, is_weekend, day_of_year, and week_of_year help the model capture seasonality and cyclical patterns that are common in time series data. By providing context and smoothing the data, moving averages and lag features can help reduce overfitting, as the model can focus on significant patterns rather than noise. The combination of these features provides the model with a comprehensive view of the data, improving its ability to make accurate forecasts.

Exploratory Data Analysis

The billing amount by date plot shows below that there can be a recognizable pattern for the monthly billing revenue data.



When we overlap the monthly billing revenue data after we set 26th of the prior month as day 1 and 25th of the current month as day 30, the pattern becomes more clear as shown below and also we can detect some potential outliers.



We performed the Anomaly Detection analysis using the Z-score, IQR, and the Isolation Forest model, and the Isolation Forest model was able to detect some anomalies. Those outliers are removed from the dataset.

Methods

Model List

We compared several models, including Random Forest, Support Vector Machine (SVM), ARIMA, and Gradient Boosting Machine (GBM). Cross-validation with different splits (`n_splits`) was employed to identify the best-performing model.

1. ARIMA: used for understanding and predicting future points in a time series. The model is characterized by three parameters: the autoregressive (AR) terms, the differencing (I) term, and the moving average (MA) terms.
2. Random Forest: Use time-series cross-validation to avoid overfitting.
3. Gradient Boosting Machines (GBM): Use XGBoost or LightGBM with time-series cross-validation.
4. Support Vector Machines (SVM): Use SVR (Support Vector Regression) with appropriate kernels.

Model Performance

We measured RMSE and R^2 to compare performance of the models.

- RMSE (Root Mean Squared Error): Measures the average magnitude of the errors between predicted and actual values. Lower values indicate better performance.
- R^2 (R-squared): Indicates the proportion of the variance in the dependent variable that is predictable from the independent variables. Higher values (closer to 1) indicate better performance, while negative values indicate poor model fit.

The results are summarized below:

	train_rmse	train_r2	val_rmse	val_r2
RandomForest	7.731131e+06	0.990420	7.052229e+06	0.990767
GBM	5.496158e+06	0.995158	5.235957e+06	0.994910
SVM	8.467195e+07	-0.149156	8.098080e+07	-0.217520
ARIMA	7.898598e+07	0.000000	7.348562e+07	-0.002575

Random Forest shows excellent performance on both training and validation sets, with high R^2 scores (close to 1) and relatively low RMSE. This indicates that the model fits the data well and generalizes effectively to the validation set. GBM also performs exceptionally well, with even lower RMSE and higher R^2 compared to Random Forest. This suggests that GBM might be slightly better at capturing the underlying patterns in the data. SVM performs poorly on this dataset, as indicated by the very high RMSE values and negative R^2 scores. Negative R^2 means that the model is performing worse than a simple mean-based model. This indicates that SVM is not suitable for this particular problem or the hyperparameters may need significant tuning. ARIMA also shows poor performance with high RMSE values and near-zero or negative R^2 scores. This suggests that the ARIMA model is not capturing the time series patterns effectively, possibly due to inappropriate hyperparameters or the nature of the data.

Cross-Validation

Cross-validation was performed with `n_splits` values of 3, 5, 7, and 10. The performance of each model was evaluated using Root Mean Squared Error (RMSE) and R-squared (R^2) metrics. Based on the results from each `n_splits` values, we can conclude that the trained

model performs well on both validation and training datasets as R-squared (R^2) scores are greater than 0.995 for the training dataset and 0.930 for the validation dataset. Although R^2 for the validation dataset is lower than for the training dataset, the drop was foreseeable and above the threshold set by business for the initial analysis (90%).

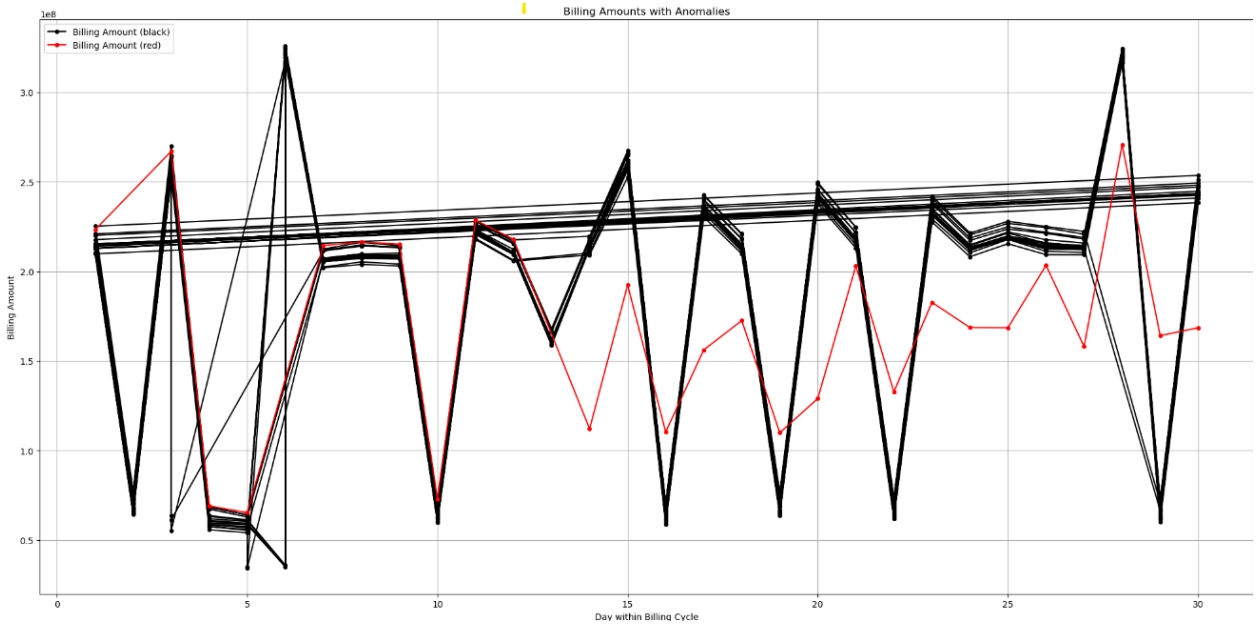
```
      train_rmse      val_rmse  train_r2    val_r2
3   4.487899e+06  1.963819e+07  0.996749  0.934494
5   4.978519e+06  2.030301e+07  0.995985  0.930138
7   5.284154e+06  1.891029e+07  0.995489  0.938309
10  5.309763e+06  1.810954e+07  0.995458  0.944087
Best number of splits: 10
```

GBM emerged as the best performer with the lowest RMSE and highest R^2 scores. As a result, we decided to use GBM to train the billing revenue dataset. The final trained model showed the following performance:

```
Final Average Train RMSE: 5309762.807454823
Final Average Validation RMSE: 18109535.887463864
Final Average Train R2: 0.9954579846594933
Final Average Validation R2: 0.9440870814237051
```

Visualization

The billing amounts were plotted over the billing cycle days, with predicted values for July highlighted in red and other data in black.



Interpretation of the Graph

The graph presents a visualization of the billing amounts within a billing cycle, with:

- **Black Lines**: Representing the actual billing amounts for various months (overlapping each other).
- **Red Line**: Representing the predicted billing amounts for July 2024.

Key Observations:

1. **Billing Patterns Across Months**:
    - The black lines show significant variation in billing amounts across different days within the billing cycle for previous months.
    - There are spikes and drops indicating high variability in billing amounts, possibly due to cyclical patterns or specific billing events.
2. **Predicted Billing Amounts for July 2024**:
    - The red line shows the predicted billing amounts for July 2024.
    - The predictions generally follow the trend seen in the actual data but with noticeable deviations at later points.
    - Around days 1-5 and 21-25, there are considerable fluctuations, reflecting the inherent volatility captured by the model.
3. **Alignment with Historical Data**:
    - The predicted values (red line) show some alignment with historical data patterns (black lines), suggesting that the model has captured some underlying trends.
    - However, there are instances where the red line deviates from the cluster of black lines, indicating periods where the model predicts significantly different billing amounts compared to historical averages.
4. **Anomalies and Consistency**:
    - The red line highlights possible anomalies or deviations that the model predicts for July 2024, which might require further investigation.
    - The consistency in certain sections indicates that the model's predictions are more stable and in line with past data during those periods.

Conclusion

Gradient Boosting Machine (GBM) proved to be the most effective model for predicting monthly billing amounts. It outperformed other models, demonstrating excellent generalization capabilities. However, the predicted values using the model showed deviation from the pattern of historical data, indicating that the model failed to capture the trends accurately. There is a room for improvements as we will focus more on detecting/removing anomalies and closely examine features used to train the model.

Assumptions

The analysis and predictions presented in this study are based on several key assumptions. Firstly, it is assumed that the dataset used is representative of future billing patterns, meaning that the historical data reflects the trends and variations that are likely to continue. Additionally, the features selected and engineered are considered relevant to the prediction task, capturing the essential factors that influence billing amounts. Lastly, it is assumed that the model will generalize well to future unseen data, maintaining its predictive accuracy and reliability when applied to new billing records.

Limitations

The analysis and predictions presented in this study are subject to several limitations. Firstly, the dataset is relatively small, which might limit the model's ability to generalize effectively to new data. This could potentially affect the robustness of the predictions when applied to larger or more varied datasets. Additionally, the model's performance is heavily dependent on the quality and completeness of the data. Inaccurate or missing data could significantly impact the model's accuracy and reliability. Furthermore, the model does not account for external factors that could influence billing amounts, such as economic conditions, market trends, or changes in regulatory policies. These external factors, if included, might enhance the model's predictive power and provide a more comprehensive understanding of the billing patterns.

Challenges

Several challenges were encountered during the analysis and model development process. Firstly, handling missing data and ensuring data quality were critical tasks that required careful attention. Incomplete or inaccurate data can lead to erroneous predictions and reduced model performance, necessitating robust data cleaning and validation procedures. Secondly, selecting the right features and engineering new ones was a complex task. It involved identifying the most relevant variables that influence billing amounts and creating new features that capture underlying patterns in the data. This step is crucial for improving the model's predictive accuracy. Lastly, balancing model complexity with the risk of overfitting posed a significant challenge. While complex models can capture intricate patterns in the data, they are also more prone to overfitting, which can result in poor performance on unseen data.

Future Uses/Additional Applications

The predictive model developed in this analysis holds significant potential for various future applications and extensions. The model can be extended to predict other financial metrics such as adjusted revenue and expenses. By incorporating additional financial data, the model could provide a comprehensive financial forecasting tool, aiding in more accurate budgeting and financial planning.

Recommendations

To maintain and enhance the model's performance, several recommendations are proposed. Firstly, it is crucial to continuously monitor the model's performance and retrain it with new data

as it becomes available. This will ensure that the model remains up-to-date and accurately reflects current billing patterns. Secondly, exploring hyperparameter tuning can further improve the model's performance. By systematically adjusting the model's parameters, we can optimize its accuracy and generalization capabilities. Lastly, investigating additional features that might enhance the model's predictive power is recommended. Incorporating new variables that capture more aspects of the billing process could lead to more accurate and reliable predictions. These steps will help in refining the model and ensuring its long-term effectiveness in predicting billing amounts.

Implementation Plan

1. **Data Collection**: Gather and preprocess new billing data monthly.
2. **Model Training**: Retrain the model with the updated dataset.
3. **Model Deployment**: Deploy the model in a production environment for real-time predictions.
4. **Monitoring and Maintenance**: Regularly monitor the model's performance and update it as necessary.

Ethical Assessment

- **Data Privacy**: Ensure that all billing data used for training and prediction is anonymized and handled in compliance with data privacy regulations.
- **Bias Mitigation**: Regularly evaluate the model for biases and ensure it provides fair predictions across different customer segments.
- **Transparency**: Maintain transparency in the model's decision-making process to build trust with stakeholders.

This white paper demonstrates a comprehensive approach to predicting monthly billing amounts using Gradient Boosting Machines. By following the outlined methods and recommendations, enterprises can enhance their financial planning and operational efficiency.

References

Crone, S. F., Hibon, M., & Nikolopoulos, K. (2011). Advances in forecasting with neural networks? Empirical evidence from the NN3 competition on time series prediction. *International Journal of Forecasting, 27*(3), 635–660. (https://www.sciencedirect.com/science/article/abs/pii/S0169207011000616?via%3Dihub)

Lazzeri, F. (2021). *Machine Learning for Time Series Forecasting with Python*. Wiley. (https://github.com/PacktPublishing/Machine-Learning-for-Time-Series-with-Python)

Lim, B., & Zohren, S. (2021). Time series forecasting with deep learning: A survey. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, 379*(2194). (https://royalsocietypublishing.org/doi/10.1098/rsta.2020.0209)