https://doi.org/10.46335/IJIES.2023.8.6.13 **e-ISSN: 2456-3463**
Vol. 8, No. 6, 2023, PP. 50-57

International Journal of Innovations in Engineering and Science, www.ijies.net

Time Series Analysis: Financial Engineering

Priya Tripathi [1], Suraj Bhanarkar[2], Rahul Gaherwar[3], Kartik Shrivas[4] , Charan Pote[5]

[1,2,3,4]Students, [5] Assistant Professor
Priyadarshini College of Engineering, Nagpur, India, 440019

priyatripathi3988@gmail.com

***Received on**: 05 April, 2023* ***Revised on**: 30 April, 2023* ***Published on**: 02 May, 2023*

Abstract – *In a subscription business, the most valuable assets are customers. It is incredibly important to monitor and proactively manage the health of customer relationships and follow informed metrics for growth strategy. Successful subscription companies achieve positive non-linear growth by retaining customers and expanding their relationships with them. Measuring recurring monetary retention and churn rate can help ensure that these relationships are on the right track. Recurring revenue is forward-looking and keeps track of how much revenue a business can count on in the future. Time-series revenue forecasting is based on a quantitative forecasting model, a data-driven technique taking personal company history into account. One looks at datasets that document cyclical fluctuations, behavioral patterns, or even certain seasonal trends. By uncovering these three fundamental factors specifically services and products to be offered and the resulting impact it has on monthly revenue, a clearer indicator of how to navigate future financial hurdles is put into prospects.*

Keywords- *Time Series, Frequency, Recency, Monetary Value, Trends, Seasonality, Anomaly Detection, Annual Recurrent Revenue, Lifetimes, Prophet, Streamlit, Product-Based Prediction*

I- INTRODUCTION

Companies across every industry are generating massive amounts of data covering all sorts of metrics. By analyzing the data and finding patterns, these organizations can discover interesting insights about their business. Much of the focus has been on using data insights to improve day to day operations and cyber security. Now companies are turning their attention to utilizing business metrics to support better business decision making. This Project is in session with the purpose of creating a commercial system for real-time analytics and automated outlier detection. To that we shall add the means to do autonomous forecasting to predict business growth and demand. It is AI-powered forecasting in a turn-key experience, meaning the solution can be used without needing to have a data scientist to forecast the time series metrics. Our Autonomous Forecast automatically manages the machine learning required to create, train, tune and deploy a customized forecasting model.

METHODOLOGY

LIFETIMES: (RECENCY, FREQUENCY, MONETARY VALUE)

Lifetimes can be used to analyze our users based on a few assumptions:

- Users interact with us when they are “alive”.
- Users under study may “die” after some period of time.

https://doi.org/10.46335/IJIES.2023.8.6.13 e-ISSN: 2456-3463
Vol. 8, No. 6, 2023, PP. 50-57

Applications

If this is too abstract, consider these applications:

- Predicting how often a visitor will return to our website. (Alive = visiting. Die = decided the website wasn't for them)
- Understanding how frequently a patient may return to a hospital. (Alive = visiting. Die = maybe the patient moved to a new city or became deceased.)
- Predicting individuals who have churned from an app using only their usage history. (Alive = logins. Die = removed the app)
- Predicting repeat purchases from a customer. (Alive = actively purchasing. Die = became disinterested with our product)
- Predicting the lifetime value of our customers

The shape of data

For all models, the following nomenclature is used:

- Frequency represents the number of repeated purchases the customer has made. This means that it's one less than the total number of purchases. This is actually slightly wrong. It's the count of time periods the customer had a purchase in. So if using days as units, then it's the count of days the customer had a purchase on.
- T represents the age of the customer in whatever time units chosen (weekly, in the above dataset). This is equal to the duration between a customer's first purchase and the end of the period under study.
- Regency represents the age of the customer when they made their most recent purchases. This is equal to the duration between a customer's first purchase and their latest purchase. (Thus, if they have made only 1 purchase, the recency is 0.)
- Monetary value represents the average value of a given customer's purchases. This is equal to the sum of all a customer's purchases divided by the total number of purchases. Note that the denominator here is different than the frequency.

Basic Frequency/ Regency analysis using the BG/NBD model.

For small samples sizes, the parameters can get implausibly large, so by adding an l2 penalty the likelihood, we can control how large these parameters can be. This is implemented as setting as positive penalizer_coef in the initialization of the model. In typical applications, penalizers on the order of 0.001 to 0.1 are effective.

Visualizing Frequency/Recency Matrix

Consider: a customer bought from us every day for three weeks straight, and we haven't heard from them in months. What are the chances they are still "alive"? Pretty small. On the other hand, a customer who historically buys from us once a quarter, and bought last quarter, is likely still alive. We can visualize this relationship using the Frequency/Recency matrix, which computes the expected number of transactions an artificial customer is to make in the next time period, given his or her recency (age at last purchase) and frequency (the number of repeat transactions he or she has made).

Ranking customers from best to worst

Rank customers from "highest expected purchases in the next period" to lowest. Models expose a method that will predict a customer's expected purchases in the next period using their history.

Customer Probability Histories

Given a customer transaction history, we can calculate their historical probability of being alive, according to our trained model.

Estimating customer lifetime value using the Gamma-Gamma model

Taking into account the economic value of each transaction, we focus mainly on transactions' occurrences. To estimate this, we can use the Gamma-Gamma sub-model. But first we need to create summary data from transactional data also containing economic values for each transaction (i.e., profits or revenues).

Estimating customer lifetime value using the Gamma-Gamma model

For this whole time, we didn't take into account the economic value of each transaction and we focused mainly on transactions' occurrences. To estimate this, we can use the Gamma-Gamma sub-model. But first we need to create summary data from transactional data also

https://doi.org/10.46335/IJIES.2023.8.6.13 **e-ISSN: 2456-3463**
Vol. 8, No. 6, 2023, PP. 50-57

International Journal of Innovations in Engineering and Science, www.ijies.net

containing economic values for each transaction (i.e., profits or revenues)

DESIGN

Graphs

Graphs are plotted with functions coming from the plotting.py file. The main functions are cited below, alongside a brief description of how they are created.

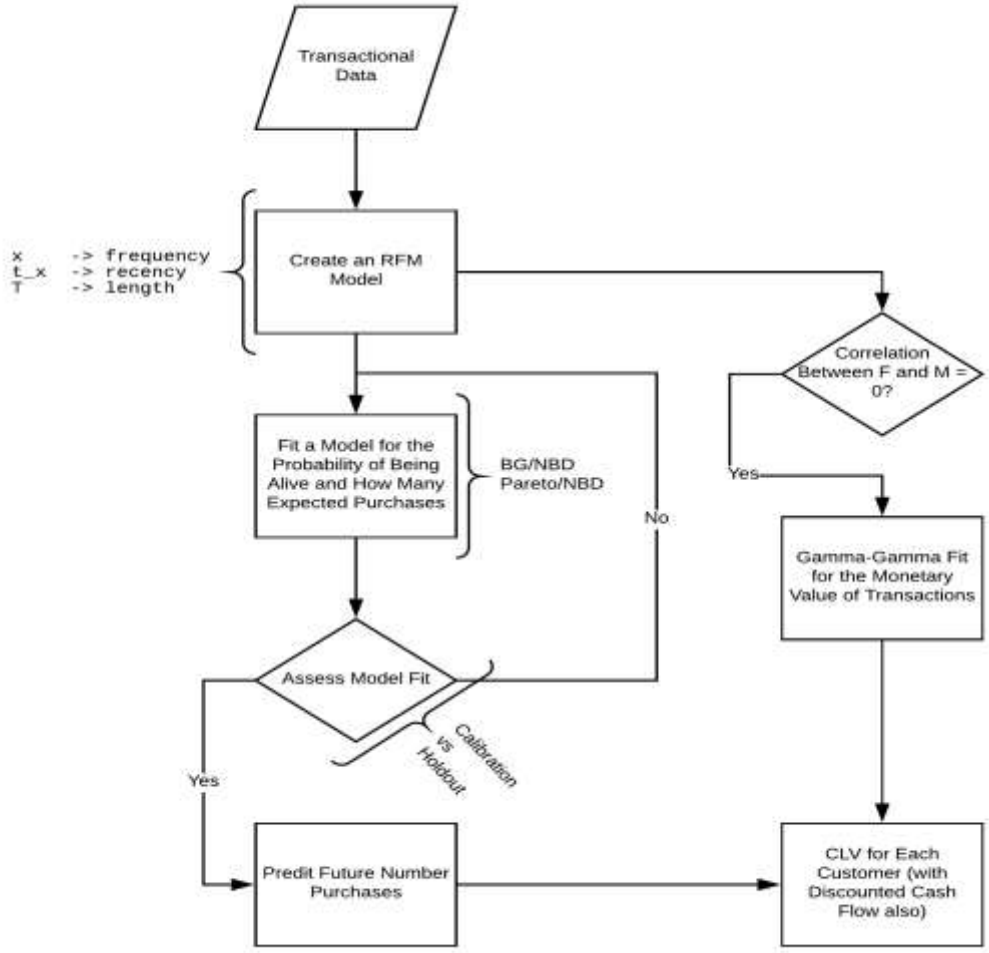
plotting.py

```
coalesce(*args)
plot_period_transactions(model, max_frequency, title, xlabel,  ylabel, **kwargs)
plot_calibration_purchases_vs_holdout_purchases(model, calibration_holdout_matrix, kind, n, **kwargs)
plot_frequency_recency_matrix(model, T, max_frequency, max_recency, title, xlabel, ylabel, **kwargs)
plot_probability_alive_matrix(model, max_frequency, max_recency, title, xlabel, ylabel, **kwargs)
plot_expected_repeat_purchases(model, title, xlabel, ax, label, **kwargs)
plot_history_alive(model, t, transactions, datetime_col, freq, start_date, ax, **kwargs)
plot_cumulative_transactions(model, transactions, datetime_col, customer_id_col, t, t_cal, datetime_format, freq, set_index_date, title,  xlabel, ylabel, ax, **kwargs)
plot_incremental_transactions(model, transactions, datetime_col, customer_id_col, t, t_cal, datetime_format, freq, set_index_date, title, xlabel, ylabel, ax, **kwargs)
plot_transaction_rate_heterogeneity(model, suptitle, xlabel, ylabel, suptitle_fontsize, **kwargs)
plot_dropout_rate_heterogeneity(model, suptitle, xlabel, ylabel, suptitle_fontsize, **kwargs)
forceAspect(ax, aspect)
```

Fig. 1- plotting.py (lifetimes core functions)

Workflow

The usual workflow of using the Lifetimes library can be represented through the following fluxogram :



https://doi.org/10.46335/IJIES.2023.8.6.13 **e-ISSN: 2456-3463**
Vol. 8, No. 6, 2023, PP. 50-57

International Journal of Innovations in Engineering and Science, www.ijies.net

Fitters

The core fitter is the Base Fitter class is inside the __init__.py, which serves as a superclass for most of the the other fitters. So far, only the Modified BetaGeoFitter is set on a higher layer, inheriting from the BetaGeoFitter. The following image shows the simplified interaction of the main fitter classes.

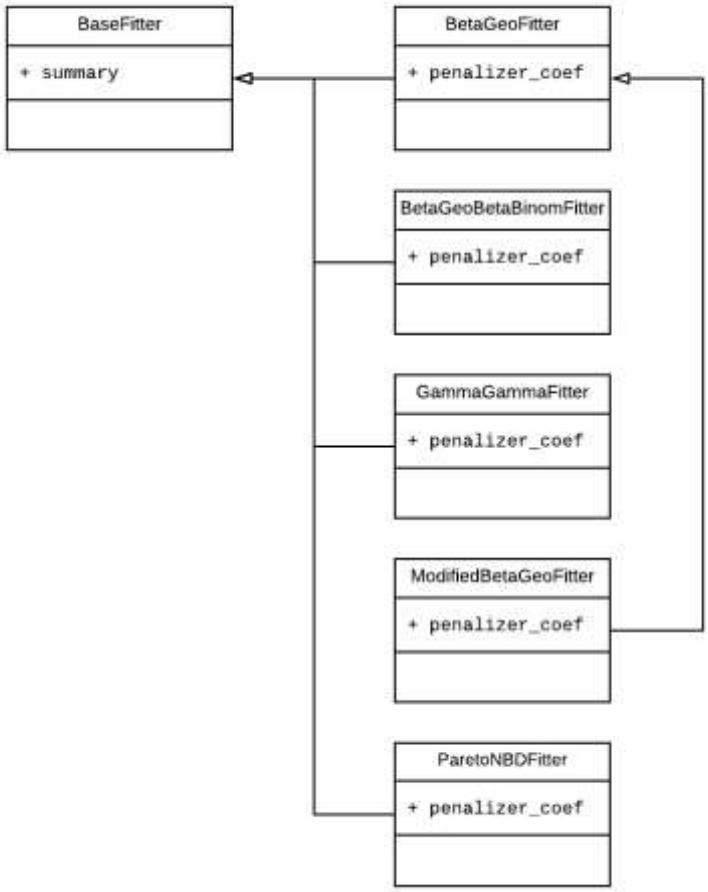


Fig. 3- lifetimes fitters workflow

PROPHET: (HANDLING SHORTCOMINGS OF LIFETIMES)

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in the trend, and typically handles outliers well.

Prophet is open source software released by Facebook's Core Data Science team. It is available for download on CRAN and PyPI.

Accurate and fast.

Prophet is used in many applications across Facebook for producing reliable forecasts for planning and goal setting. We've found it to perform better than any other approach in the majority of cases. It fit models in Stan so that we get forecasts in just a few seconds.

Fully automatic.

Get a reasonable forecast on messy data with no manual effort. Prophet is robust to outliers, missing data, and dramatic changes in our time series.

Tunable forecasts.

The Prophet procedure includes many possibilities for users to tweak and adjust forecasts. We can use human-interpretable parameters to improve our forecast by adding our domain knowledge.

Forecasting is a common data science task that helps organizations with capacity planning, goal setting, and anomaly detection. Despite its importance, there are serious challenges associated with producing reliable and high-quality forecasts – especially when there are a variety of time series and analysts with expertise in time

https://doi.org/10.46335/IJIES.2023.8.6.13 **e-ISSN: 2456-3463**
Vol. 8, No. 6, 2023, PP. 50-57

International Journal of Innovations in Engineering and Science, www.ijies.net

series modeling are relatively rare. To address these challenges, prophet describe a practical approach to forecasting "at scale" that combines configurable models with analyst-in-the-loop performance analysis. It proposes a modular regression model with interpretable parameters that can be intuitively adjusted by analysts with domain knowledge about the time series. It describes performance analyses to compare and evaluate forecasting procedures, and automatically flag forecasts for manual review and adjustment. Tools that help analysts to use their expertise most effectively enable reliable, practical forecasting of business time series.

Streamlit: (Full stack Application)

Streamlit is a framework that turns Python scripts into interactive apps, giving data scientists the ability to quickly create data and model-based apps for the entire company.

A simple Streamlit app is:

```
import streamlit as st
number = st.slider("Pick a number: ", min_value=1, max_value=10)
st.text("our number is " + str(number))
```

When we streamlit run my_app.py, we start a web server that runs the interactive application on our local computer at http://localhost:8501. This is great for local development. When we want to share with our colleagues, Streamlit Community Cloud enables us to deploy and run these applications in the cloud. Streamlit Community Cloud handles all the details of scaling, reliability, and security as well as providing us an interface for easily managing our deployed apps.

Running Streamlit script

Working with Streamlit is simple. First, we sprinkle a few Streamlit commands into a normal Python script, and then we run it. We list few ways to run our script, depending on our use case.

Use streamlit run.

Once we've created our script, say our_script.py, the easiest way to run it is with streamlit run:

streamlit run our_script.py As soon as we run the script as shown above, a local Streamlit server will spin up and our app will open in a new tab in our default web browser.

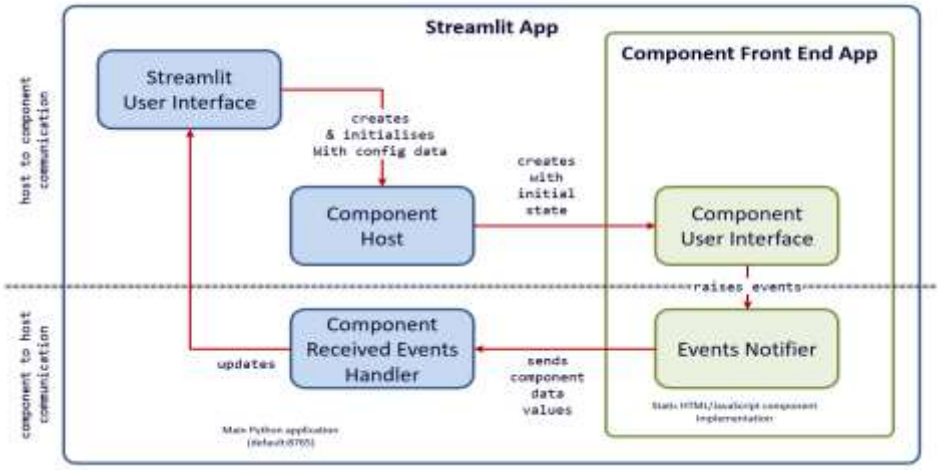
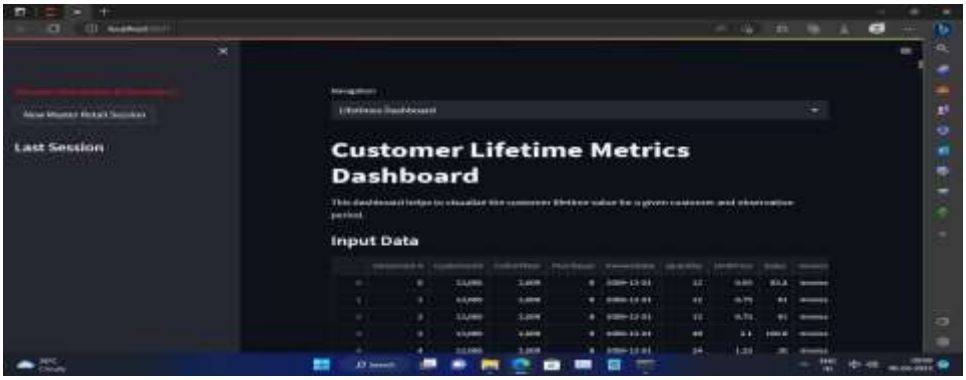


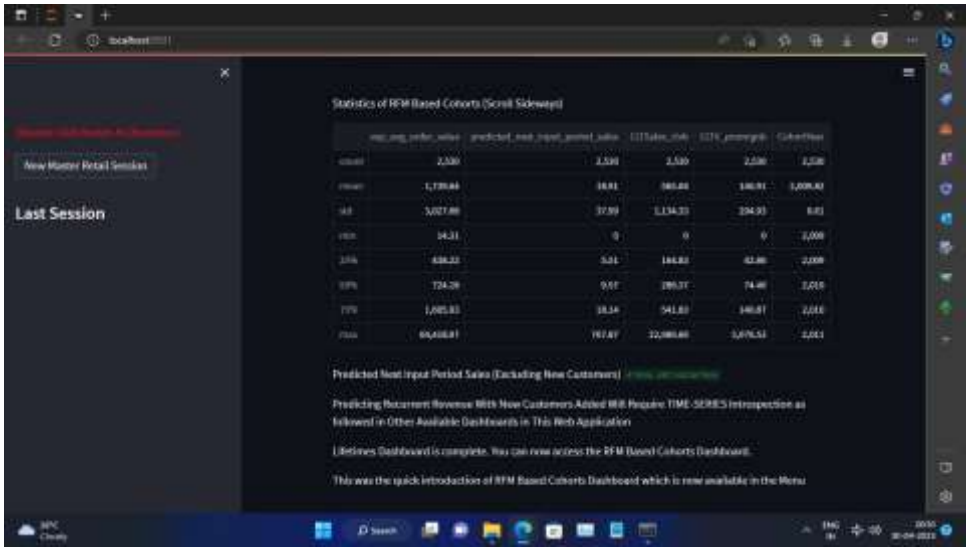
Fig. 4- streamlit's full stack aplication workflow

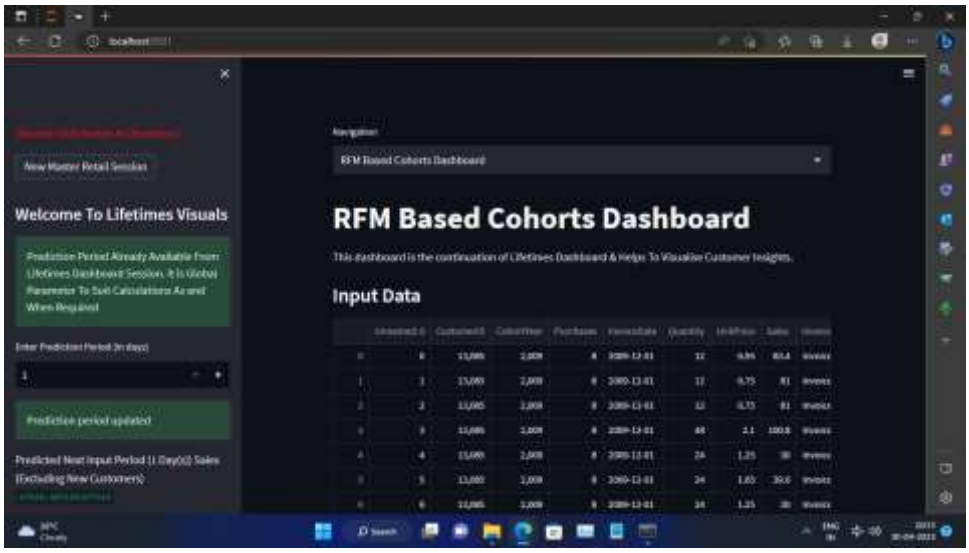
USER INTERFACE

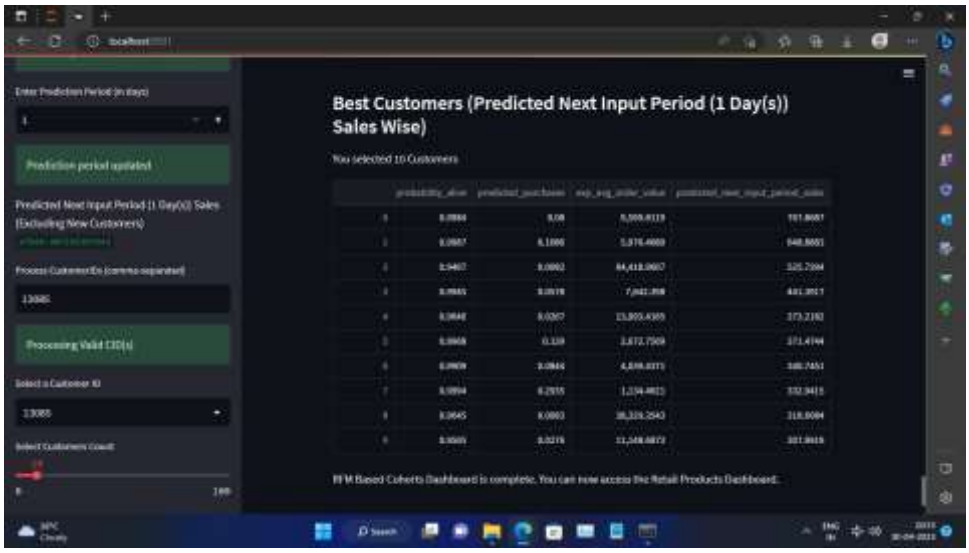


https://doi.org/10.46335/IJIES.2023.8.6.13 **e-ISSN: 2456-3463**
Vol. 8, No. 6, 2023, PP. 50-57

International Journal of Innovations in Engineering and Science, www.ijies.net

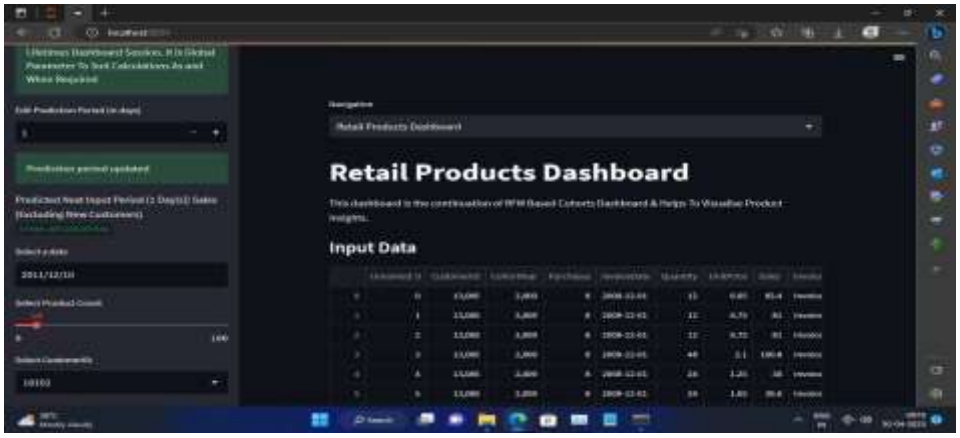






https://doi.org/10.46335/IJIES.2023.8.6.13 **e-ISSN: 2456-3463**
Vol. 8, No. 6, 2023, PP. 50-57

International Journal of Innovations in Engineering and Science, www.ijies.net



CONCLUSION

As companies work to recalibrate their businesses to meet the demands of the subscription era, a proven framework for transformation is crucial. And the good news is that in the Subscription Economy, we can all learn from each other; we don't have to reinvent the wheel by ourselves. The Paper reflect the experiences of others who've gone before us to shine a bright light on what to do—and what not to do—to find way. No matter which stage we're in, what industry we're in, or where we are located, the strategies and tactics we've covered can help us build a subscriber-centric business that's ready for today, tomorrow—and for decades to come.

- **Industry Compliant and Licensed Sophisticated Models**
    - The working ensemble models of churn and revenue as licensed under Apache 2.0 providing customer and enterprise data security on Google Cloud Platform (Similar) Cloud.
- **Working Progressive Application (Web/Native):**
    - Upon entering sample values for feature parameters of customers, the server application returns expected churn and revenue prediction probability.
- **Single Code Base:**
    - A single code base, enabling every customer to run their business on the latest release without disruption.
- "Productized" the process of ML-based forecasting by developing a sophisticated system to automatically create, train, tune and deploy a customized forecasting model.
- Combated the "Curse of Dimensionality"
- Identified and accounted for different data anomalies and time series behaviour**.**
- Tracked growth and customer retention by examining the total number of subscribers.
- Identifies and mitigated churn risk, and ensured the business has healthy cash flow.
- Integrated key systems together while planning for consolidation and simplification.
- Rationalized the enterprise tech stack.
- Undertook exploratory Time Series data analysis.
- Continuous monitoring of customer data for preventing churn.
- Examined and calculated recurring revenue metrics for understanding predictable, regular income stream.
- Compared the total number of subscribers with one or more subscription charges in a given period to the previous time period to examine growth characteristics**.**
- Tested the accuracy of a forecasting model by capturing the temporal dynamics of time series.
- Worked with seasonal data pattern as temporally ordered data matters.
- Created ensemble of models by factoring in multitude of machine learning algorithms.

REFERENCES

[1] *(99+) Amanda Iglesias Moreno | LinkedIn*
[2] *Aileen Nielsen/Time Series Analysis With Python (github.com)*
[3] *Practical Time Series Analysis [Book] (oreilly.com)*
[4] *Advances in Time Series Data Methods in Applied Economic Research: International Conference on Applied Economics (ICOAE) 2018 - PDF Drive*
[5] *All: time series : Search (acm.org)*

[6] *Credit Risk Score - Credit Scoring Model Solutions | Experian*
[7] Subscription Economy Index
[8] *Fighting Churn with Data*
[9] *Revenue Forecast Accuracy*
[10] *Journey to Usership*
[11] *The Essential Guide to Time Series Forecasting*
[12] *Anomaly Detection System*
[13] *THE SUBSCRIBED STRATEGY GROUP*
[14] *The Key Principles of a Successful Time Series Forecasting System for Business*
[15] *Advancing Computing as a Science & Profession*
[16] *Learning platform your tech teams need to stay ahead*
[17] *Python's reduce(): From Functional to Pythonic Style*