1.Introduction

1.1) Problem Statement- The goal of this project is to develop a machine learning model that accurately predicts insurance prices based on various customer features. By analysing features, the model should be able to estimate the price of insurance premiums for new customers. The prediction will help insurance companies optimize pricing, assess risk more effectively, and provide personalized quotes to potential clients.

1.2) Abstract- Insurance pricing is a complex process influenced by multiple factors such as customer demographics, vehicle details, driving history, and claim records. Accurate price prediction can improve decision-making for insurance companies and provide tailored offers to clients. The objective of this project is to predict insurance premiums using machine learning techniques. The goal is to build a model that can predict accurate insurance premiums for new customers, ultimately aiding in risk assessment, pricing strategy optimization, and customer satisfaction.

1.3) Scope: The scope of our project focuses on leveraging data analytics and machine learning techniques to predict the premiums for various types of insurance policies, such as health, life, or auto insurance. The project involves analysing historical data, customer demographics, claims history, and other relevant features to identify patterns and factors that influence pricing decisions.

2.Overall Description

2.1) Workflow diagram-

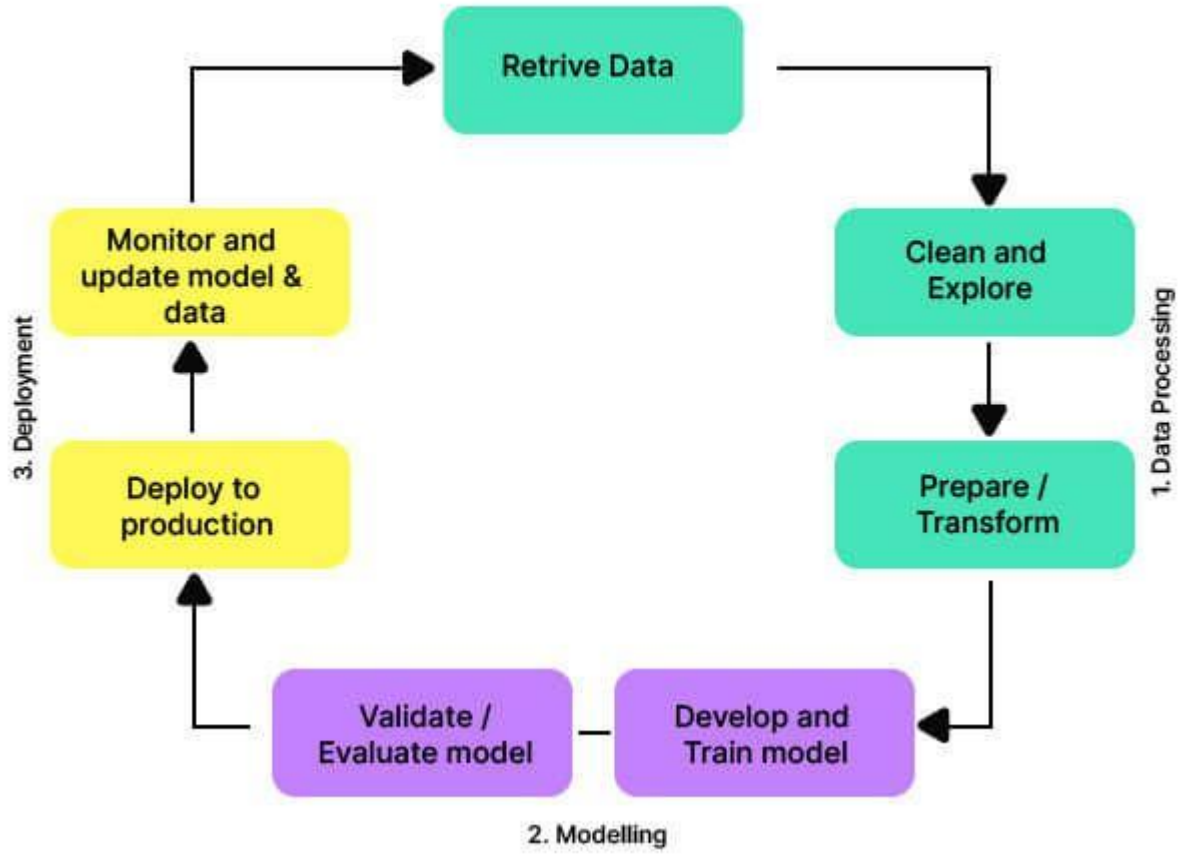

Fig2.1- Workflow

2.2) Data Cleaning and Preprocessing-

- Since our dataset is very huge of 80lakh rows of different companies their products and different state spanning from 2003 January to 2023 . Hence we decided to focus on a product of a company which has the highest premiums .

- Converting year_month column to datetime

- Checking for missing values in the data if present replacing by sum

- Outlier handling of our target column ‘premiums’

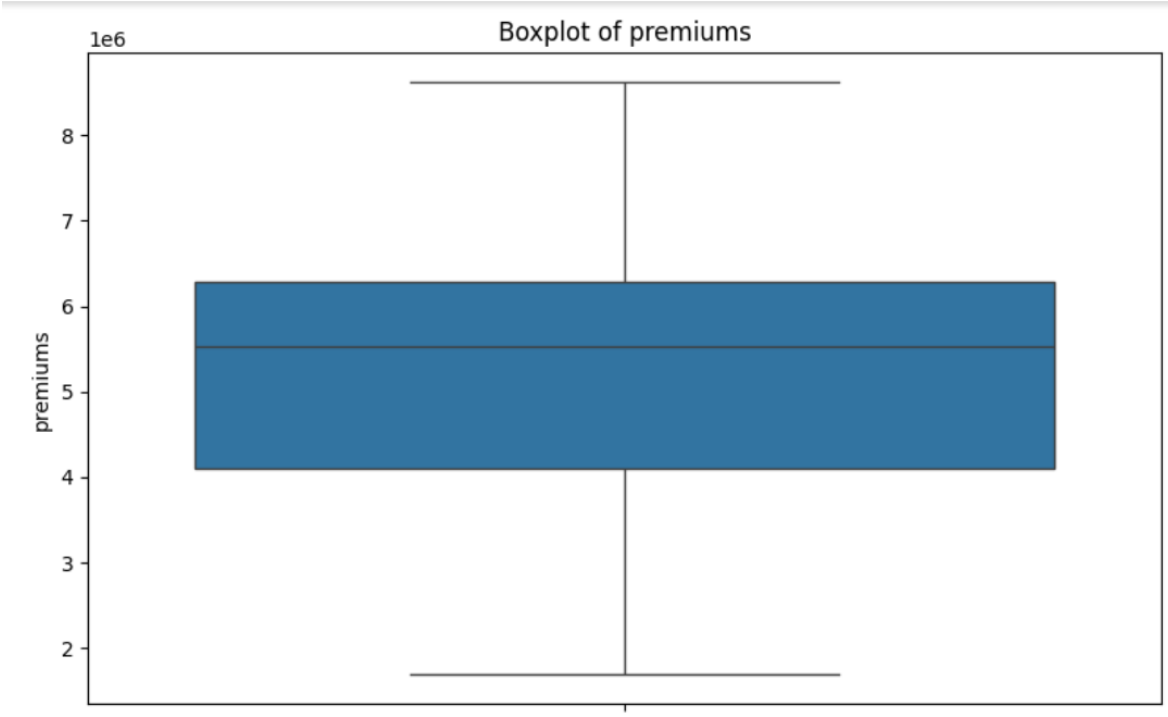

Fig 2.2.1- Boxplot to check outliers in Premium column

-Outlier Detection

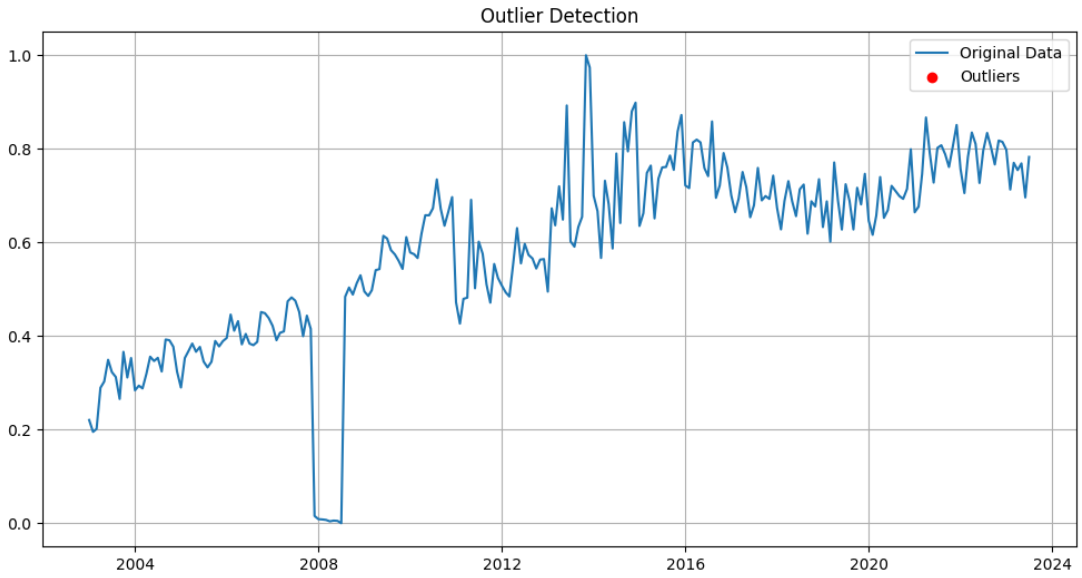

Fig 2.2.2- To check outliers

```
Outliers detected:
             company_code                   company_name  \
year_month_f
2006-09-01           6785  BRASILSEG COMPANHIA DE SEGUROS

                          product state  premiums   claims  \
year_month_f
2006-09-01    0993 - Vida em Grupo    TO       1.0 -5864.95

             claim_premium_ratio  premiums_diff    zscore
year_month_f
2006-09-01               -0.0024       0.780764  5.909851
```

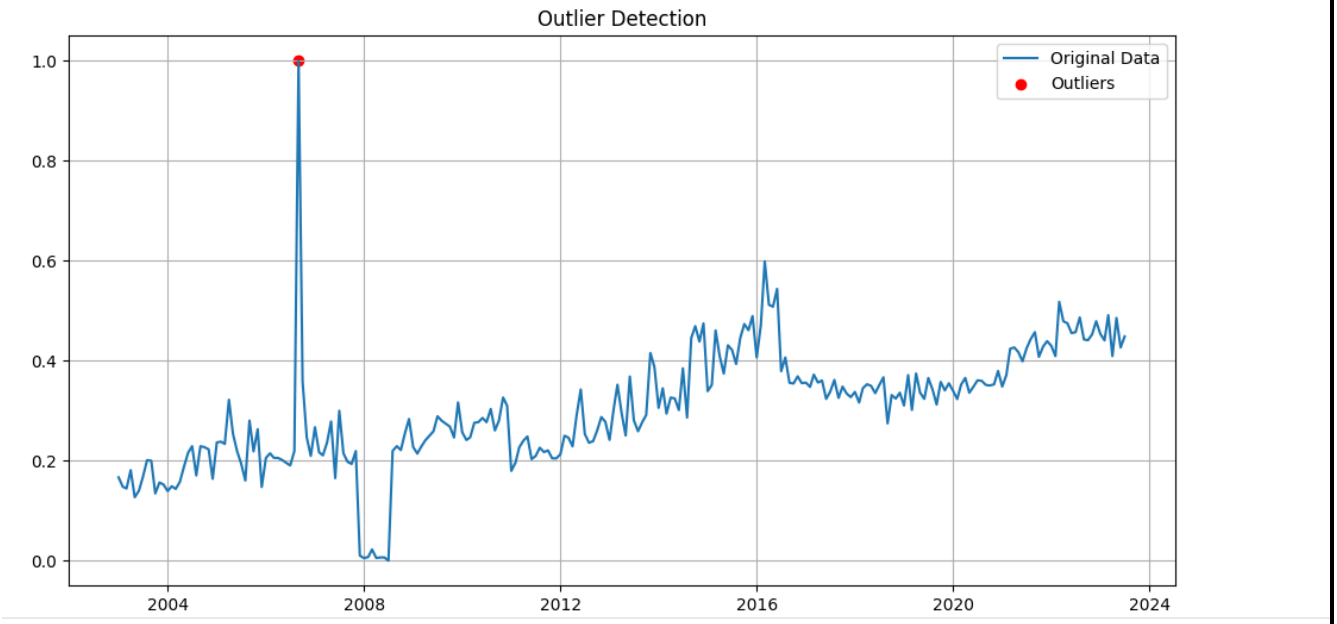

Fig 2.2.3- Significant outliers

- In 2008, the Brazilian market experienced a strong economic boom fueled by domestic demand, but this was abruptly interrupted towards the end of the year by the global financial crisis, leading to a sharp slowdown in growth as exports declined and credit became tighter, with the industrial sector being particularly affected by the downturn.

-Checking distribution of other columns in dataset

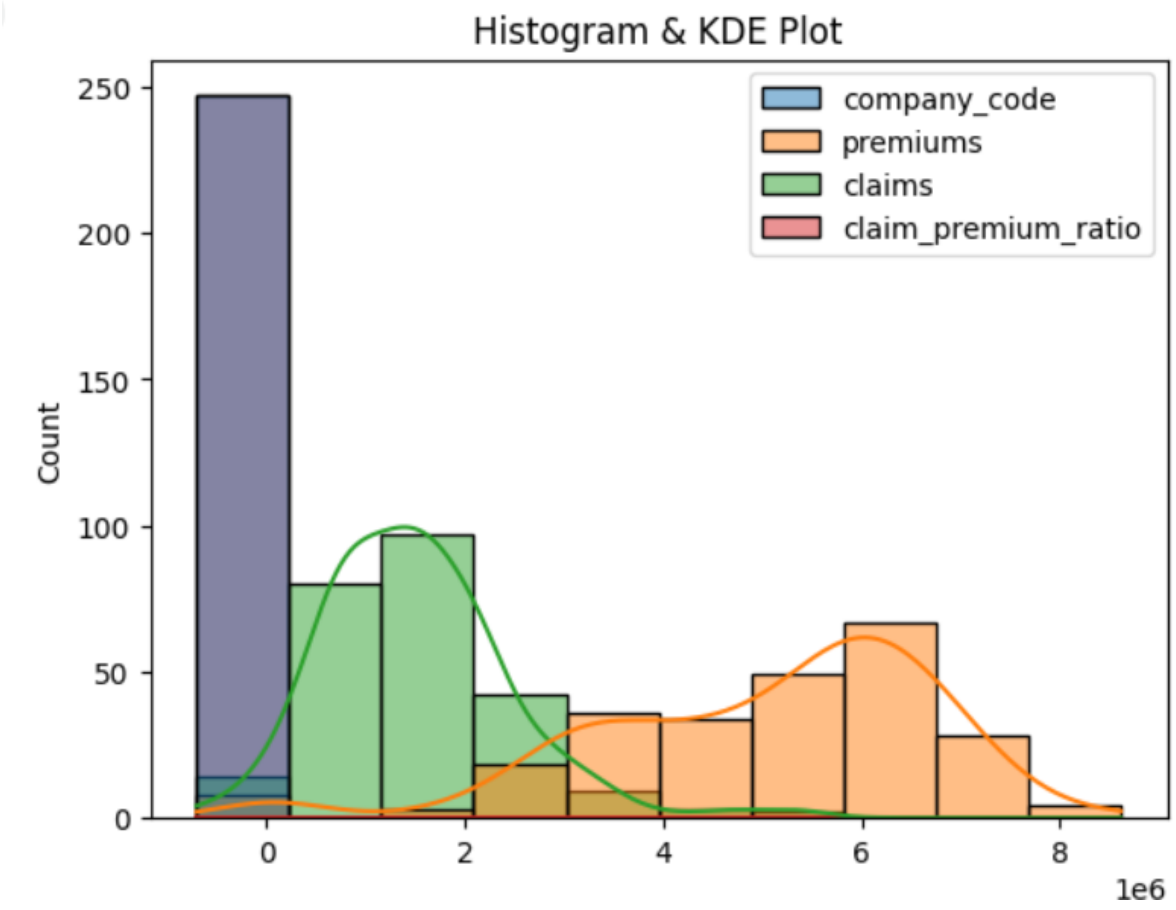

Fig 2.2.4- Plot to check distribution of various columns

-Checking the shape of data (skewness, kurtosis) and then applying min-max scaler for normalization

- One hot encoding for 3 columns (product,state and company_name) as they are categorical

- Creating plots for premium column with rolling windown , mean and std deviation

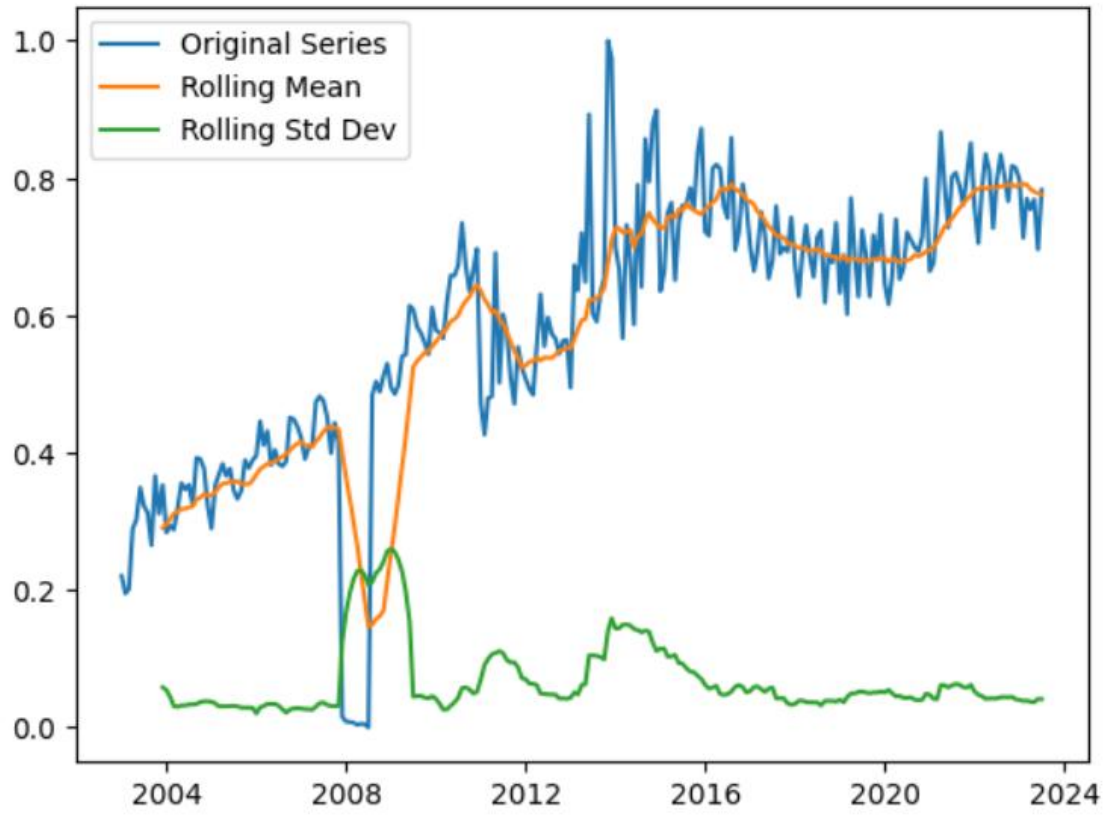

Fig 2.2.5- Line chart to check Rolling mean, Rolling Std_Dev

Key Insights:

- The period around 2008 shows a significant disruption, which reflects the Global Financial Crisis of 2008-2009
- After 2008, the rolling mean shows a smoother trend with fewer sharp fluctuations, and the standard deviation stabilizes.
- This suggests that the data becomes more predictable and less volatile after this period.

- Summary statistics of premiums-

count	247.000000
mean	0.583706
std	0.196570
min	0.000000
25%	0.447281
50%	0.632887
75%	0.725663
max	1.000000

- ADF test to check stationarity of data –

```
ADF Statistic: -1.5794087195485458
p-value: 0.4939283833112437
Critical Values: {'1%': -3.4589796764641, '5%': -2.8741347158661448, '10%': -
The data is not stationary. Differencing is required.
```

p-value clearly shows data isn't stationary

- Techniques to make data is stationary

a) Differencing – Differencing removes non-stationary components like trends and seasonality, making the data suitable for statistical modelling.

After applying differencing -

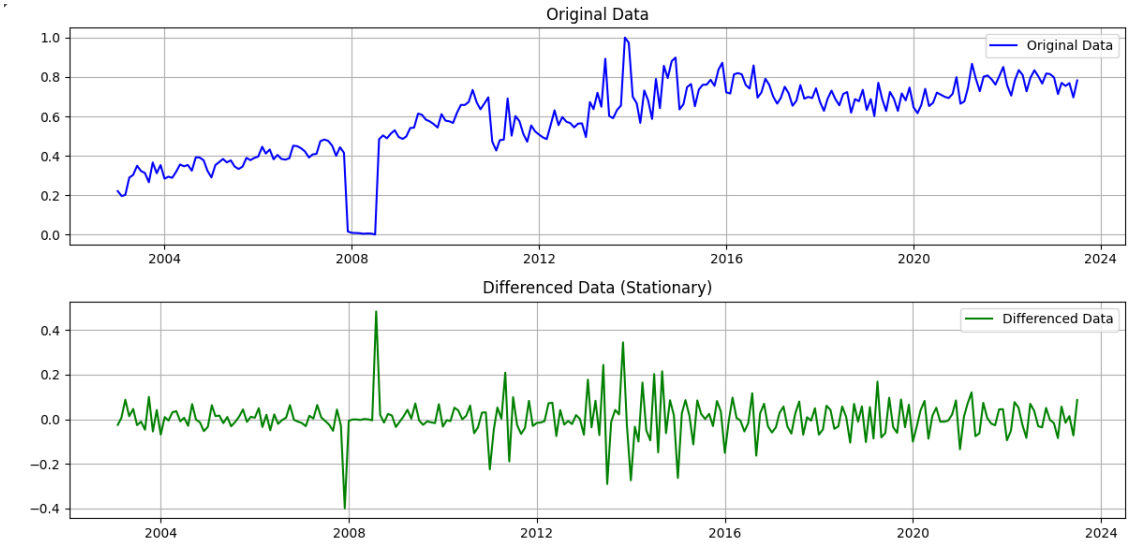

Fig 2.2.6- Original and Differenced data plots

b) Seasonal decompose- Breaks the series into components (trend, seasonality, residual) for analysis.

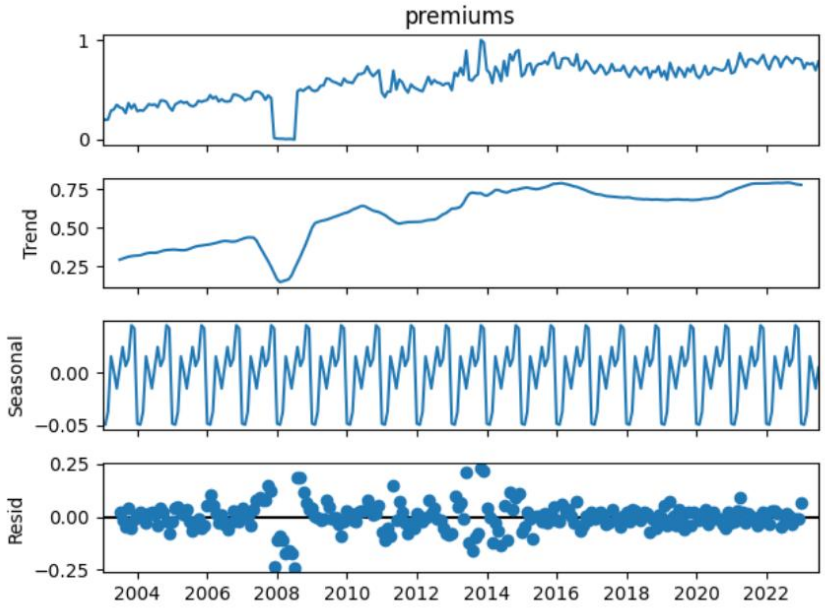

Fig 2.2.7-Breakdown into plots showing Trend, Seasonality, Residuals

3.Model Training

1) SARIMA-

SARIMA (Seasonal Autoregressive Integrated Moving Average) extends ARIMA by incorporating a **seasonal component**, making it suitable for time series data that exhibits repeating patterns over fixed intervals (e.g., monthly sales, quarterly revenue). It adds seasonal terms to ARIMA, denoted as **(p, d, q) × (P, D, Q, s)**, where:

(P, D, Q, s) represent the **seasonal** counterparts:

- **P** (Seasonal AR terms)
- **D** (Seasonal differencing)
- **Q** (Seasonal MA terms)
- **s** (Seasonal period, e.g., 12 for yearly seasonality in monthly data)

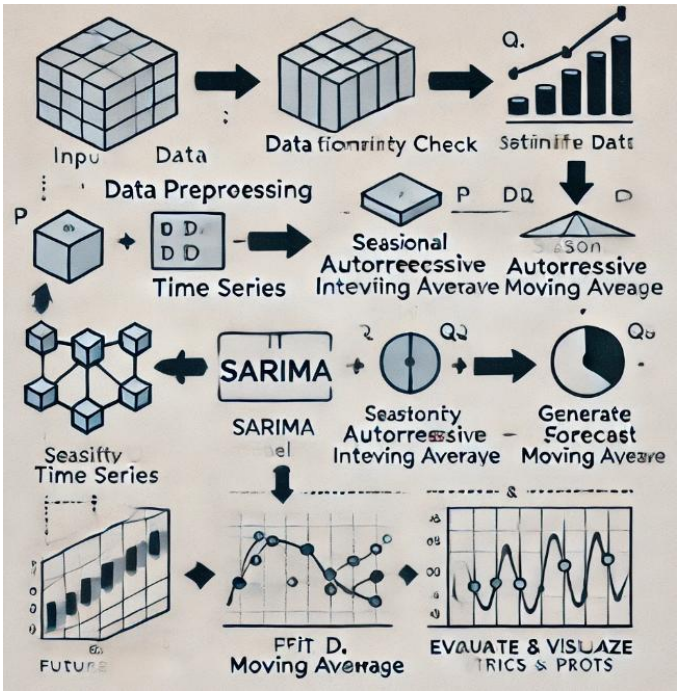

Fig 3.1- Working of SARIMA

Results-

```
Best model:  ARIMA(0,1,1)(1,0,0)[12]
Total fit time: 43.926 seconds
                                     SARIMAX Results
==========================================================================================
Dep. Variable:                                   y   No. Observations:                  247
Model:             SARIMAX(0, 1, 1)x(1, 0, [], 12)   Log Likelihood                410.721
Date:                             Mon, 03 Feb 2025   AIC                           -815.441
Time:                                     17:28:17   BIC                           -804.925
Sample:                                 01-01-2003   HQIC                          -811.207
                                      - 07-01-2023
Covariance Type:                              opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ma.L1         -0.4359      0.046     -9.564      0.000      -0.525      -0.347
ar.S.L12       0.1467      0.057      2.563      0.010       0.035       0.259
sigma2         0.0021      0.000     18.652      0.000       0.002       0.002
===================================================================================
Ljung-Box (L1) (Q):                   0.01   Jarque-Bera (JB):               141.66
Prob(Q):                              0.94   Prob(JB):                         0.00
Heteroskedasticity (H):               0.24   Skew:                             0.24
Prob(H) (two-sided):                  0.00   Kurtosis:                         6.69
===================================================================================
```

Fig 3.2- Results of ARIMA

Model Fit (AIC = -815.441, BIC = -804.925)

- A **lower AIC/BIC** indicates a better model fit. These values suggest the model is relatively good.
- **Heteroskedasticity Test (H = 0.24, p < 0.01)**: Variance is **not constant**, indicating some periods may be more volatile.
- The low variance of residuals indicates **a good fit**.

2) ARIMA

- Auto ARIMA is an advanced time series forecasting technique that automates the process of selecting the optimal ARIMA model parameters. By analysing historical data, it identifies the best combination of autoregressive (AR), differencing (I), and moving average (MA) terms to minimize forecasting error. This eliminates the need for manual parameter tuning, making the modelling process more efficient and accurate.

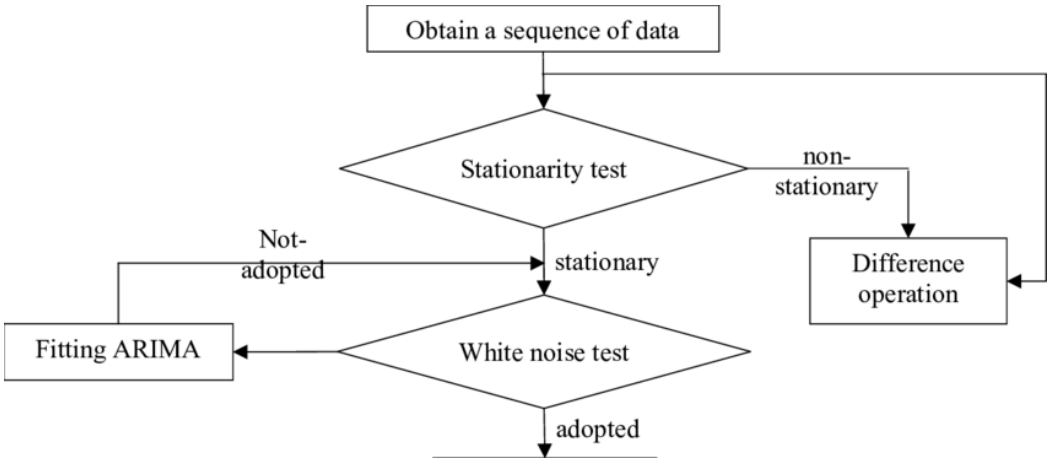

Fig 3.3 ARIMA Working

- **ARIMA Evaluation-**

SARIMAX Results

Dep. Variable:	premiums	No. Observations:	247
Model:	ARIMA(1, 1, 1)	Log Likelihood	408.359
Date:	Mon, 03 Feb 2025	AIC	-810.718
Time:	17:28:17	BIC	-800.202
Sample:	01-01-2003	HQIC	-806.484
	- 07-01-2023		
Covariance Type:	opg		

	coef	std err	z	P>|z|	[0.025	0.975]
ar.L1	0.1141	0.132	0.863	0.388	-0.145	0.373
ma.L1	-0.5424	0.119	-4.557	0.000	-0.776	-0.309
sigma2	0.0021	0.000	18.619	0.000	0.002	0.002

Ljung-Box (L1) (Q):	0.00	Jarque-Bera (JB):	141.33
Prob(Q):	0.97	Prob(JB):	0.00
Heteroskedasticity (H):	0.26	Skew:	0.14
Prob(H) (two-sided):	0.00	Kurtosis:	6.70

Fig 3.4 ARIMA Results

Results-

AIC = -810.718, BIC = -800.202:

- Lower values suggest a good fit, but these should be compared with other models for confirmation.
- J**arque-Bera Test (JB = 141.33, p < 0.01)**:Residuals **are not normally distributed**, which may affect prediction intervals.
- **Heteroskedasticity (H = 0.26, p < 0.01)**:The variance of residuals is **not constant**, indicating potential volatility in premiums.

3) Facebook Prophet –

Prophet is a powerful time series forecasting model developed by Meta, designed for handling **trend, seasonality, and holiday effects** in data. Prophet is particularly useful because it automatically detects patterns in time series data and provides robust predictions with minimal parameter tuning. It follows an **additive model structure**, where the overall forecast is a combination of a **trend component**, **seasonal effects**, and **holiday adjustments**. In out project, Prophet is applied to predict **insurance premiums**, leveraging its ability to handle missing data, detect yearly/weekly seasonality, and incorporate external factors if needed. Unlike traditional ARIMA models, Prophet can effectively **capture nonlinear trends** and **adjust dynamically** to changes in data patterns, making it well-suited for real-world financial forecasting. Its interpretability and automated hyperparameter selection make it a valuable tool for improving the accuracy of my premium predictions.

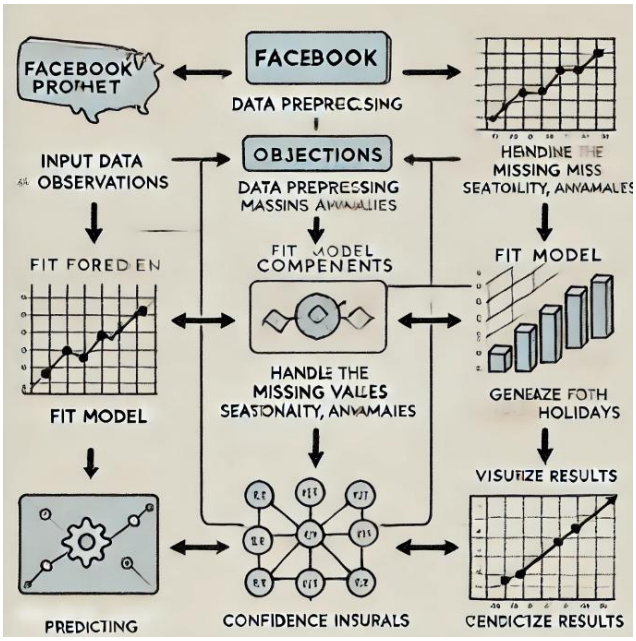

Fig 3.5- Prophet Working

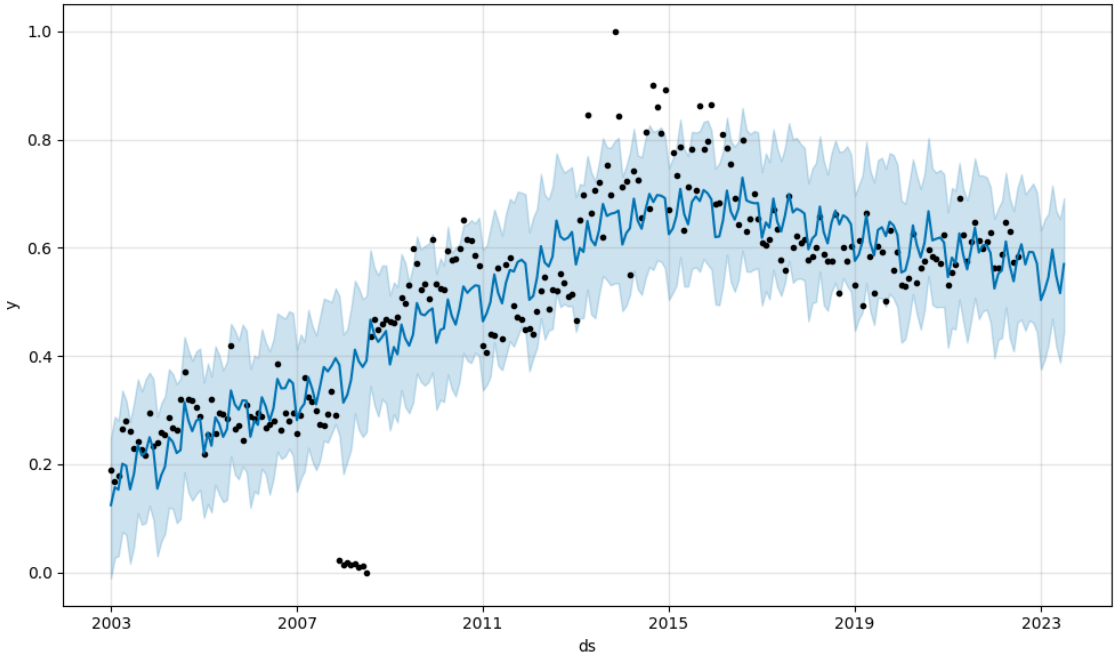

Fig 3.6-Checking outliers with Prophet

Overall Trend:

- The model captures a clear upward trend in the data until around 2015, followed by a gradual downward trend.
- This suggests a peak in the observed variable around 2015, followed by a decline.

Seasonality:

- The oscillating pattern in the blue line indicates seasonal effects, meaning the observed variable fluctuates periodically (e.g., annually or quarterly).

Confidence Intervals:

- The light blue shaded area represents the model's uncertainty intervals. Wider intervals suggest higher uncertainty in the predictions, especially at the edges of the forecast range.

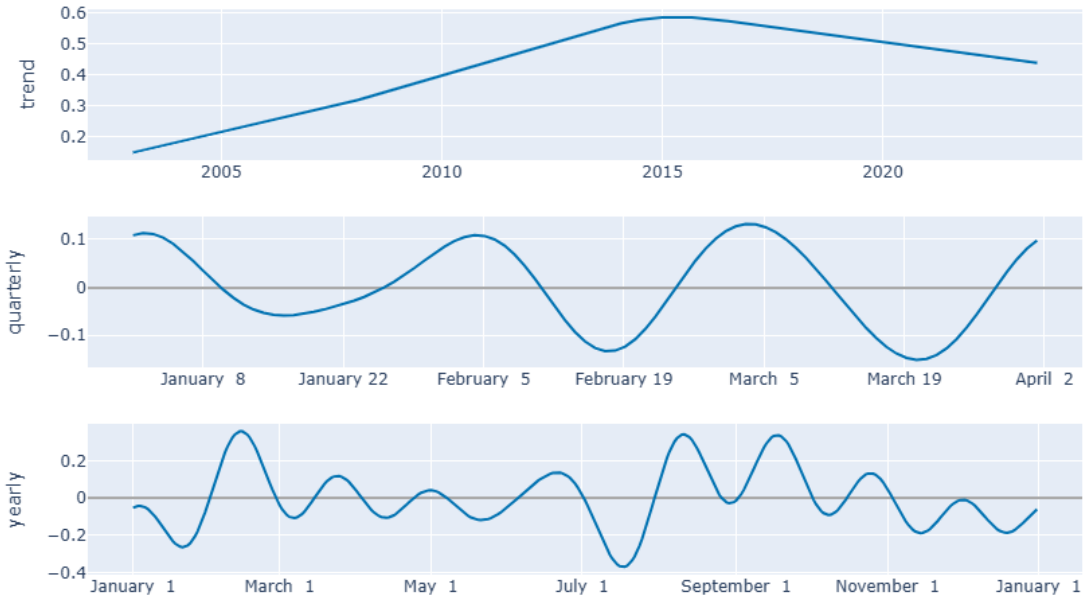

Fig 3.7- Plots of yearly, quarterly trends

Evaluation metrics-

MAE: 28.10

RMSE: 1476.45

MAPE: 4.74%

Performance:

- The low MAPE (4.74%) indicates the model performs well in capturing relative changes in the data.
- However, the high RMSE (1476.45) suggests that while most predictions are accurate, there are some significant errors or outliers affecting the overall performance.

4.User Interface

After training the models and determining that Facebook Prophet performed best for our forecasting tasks, we proceeded with building a user interface for seamless interaction with our models. To achieve this, we utilized the Flask web framework, known for its simplicity and robustness. Additionally, we deployed the application on AWS EC2 using Flask, ensuring efficient and scalable access to our models for making predictions.

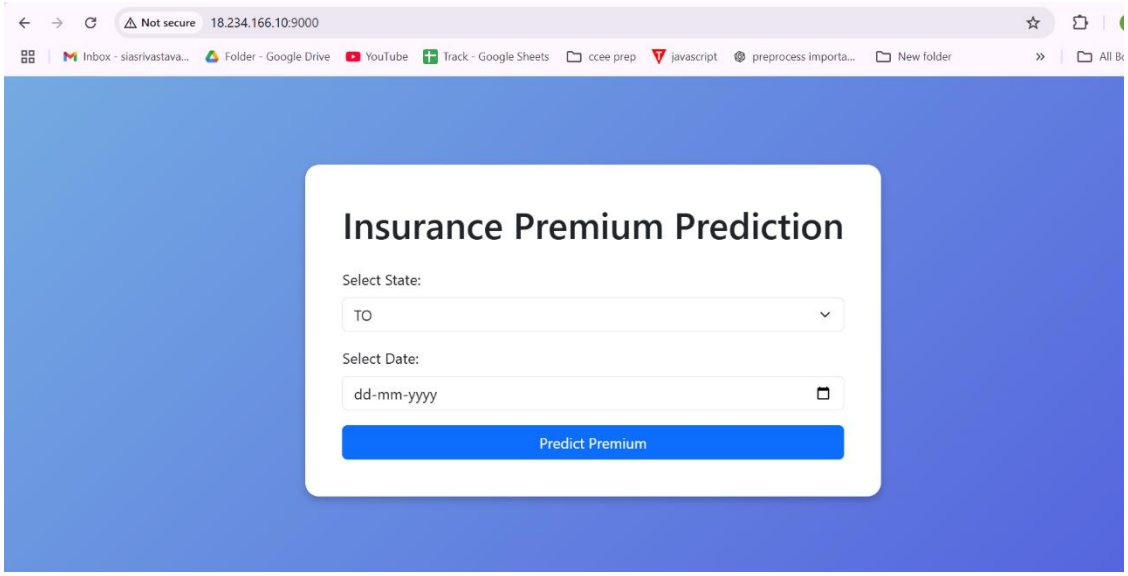

Fig 4.1- Main Landing page

- The user will select state in dropdown and date for which they want to predict.
- On selecting predict button, it will generate results.

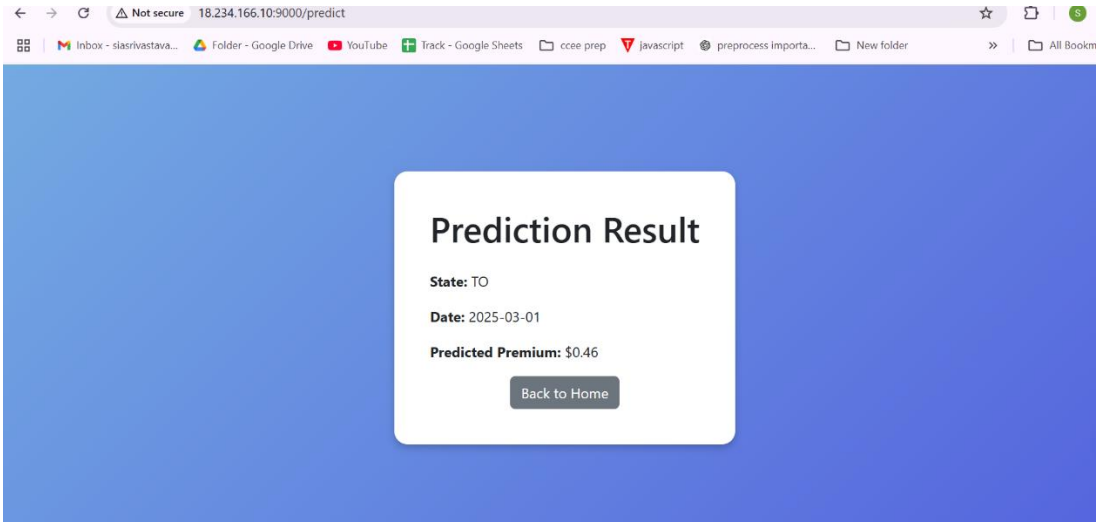

Fig 4.2- Result page

- The results are displayed for the particular user inputs.

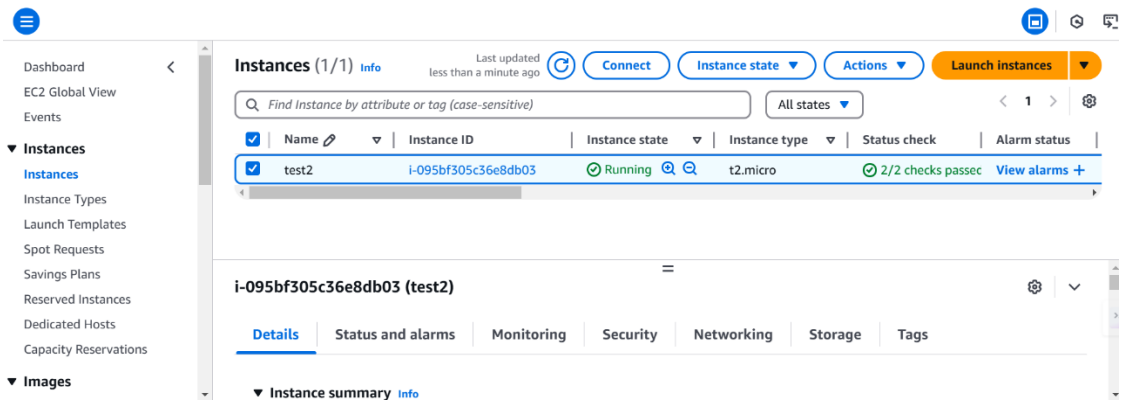

Fig 4.3- Deployment using EC2

5. Requirements & Specifications

5.1 Hardware Requirement:

- 500 GB hard drive (Minimum requirement)
- 8 GB RAM (Minimum requirement)
- PC x64-bit CPU

5.2 Software Requirement:

- Windows/Mac/Linux
- Python-3.9.10
- VS Code/Anaconda/Google Colab/Jupyter
- Python Extension for VS Code
- **Libraries:**
  - Flask=1.1.1
  - prophet==1.1.5
  - numpy=1.9.2
  - scipy>=0.15.1
  - scikit-learn=0.18
  - matplotlib=1.4.3
  - pandas=0.19
  - Any Modern Web Browser like Google Chrome
    To access the web application written in Flask
  - AWS Cloud Platform ==> EC2 service

6. Conclusion

This project successfully implemented a predictive modelling system using Facebook Prophet to forecast trends and make data-driven predictions. By leveraging time series forecasting techniques, we identified key patterns and insights, enabling more informed decision-making. The integration of Flask allowed us to build an interactive and user-friendly interface, making it easy to access and utilize the predictive models. Furthermore, deploying the application on AWS EC2 ensured scalability and efficient model serving, allowing for real-time predictions with minimal latency.

The project demonstrated the effectiveness of machine learning and cloud deployment in solving real-world forecasting challenges. Additionally, extensive evaluation and fine-tuning of the model ensured optimal accuracy, making it a reliable tool for predictive analytics. The seamless combination of data preprocessing, model selection, and deployment highlights the importance of a well-structured end-to-end machine learning pipeline.

This project serves as a foundation for further exploration in predictive analytics and showcases the potential of cloud-based AI solutions in various industries.

7.Future Scope and Enhancements

Moving forward, several enhancements can be made to further improve the project. One key area is incorporating multiple forecasting models and implementing an ensemble approach to improve accuracy and robustness. Additionally, integrating real-time data streaming using tools like Apache Kafka can help the model adapt dynamically to new information. Enhancing the user interface with interactive visualizations using Plotly or D3.js will provide deeper insights into trends and forecasts.

Another potential improvement is optimizing deployment by leveraging AWS Lambda or containerization with Docker and Kubernetes for better scalability. Furthermore, implementing automated model retraining pipelines will help keep the model updated with the latest data patterns, improving long-term reliability.

Expanding the scope by incorporating additional external data sources, such as economic indicators or social media sentiment analysis, could enhance predictive performance. Finally, integrating advanced AI techniques, such as deep learning-based forecasting models, can further refine predictions. By continuously evolving the project with these enhancements, it can serve as a more powerful and adaptable forecasting tool for diverse applications.