Project Report on Time Series Analysis of Food Delivery Metrics Using Zomato Dataset

I. Introduction and Objective

Background and Motivation

In our speedy online world, services like Zomato that deliver food are super important. They make sure folks get their meals fast and without a hitch. With more and more people asking for things on-demand, figuring out and trimming down how fast things can be delivered is top on the list for businesses. What amount of time it takes to deliver something can change based on lots of stuff. This could be the weather, how bad traffic is, the time of day, the day of the week, or even special times like holidays.

Looking at things over time can help understand these problems. This can also show how things change and trends. By keeping an eye on how deliveries change over time, businesses can guess what might happen in the future. They can then plan out resources. Guessing what comes next can help make better choices, keep customers happy, and cut down costs.

Objective of the Project

This project is like a food delivery puzzle. We're playing detective with Zomato's records. It's not a dusty old book we're examining - it's vibrant real-time records! We want to see if we can spot any patterns tied to timing. Tick-tock, when do their deliveries speed up or slow down? We plan to:

- Find the repeating patterns, any ups-and-downs, and odd one-outs in how deliveries are made.
- Spot the clues that tell us when deliveries happen faster.
- Try to guess future delivery times by making our own predictive models.
- Check how well different guess-making methods like ARIMA, SARIMA, and LSTM are doing.
- Use all this treasure trove of data to offer ways to make deliveries zippy quick..

Research Questions

Let's make this easy to understand. We're on a mission to answer some questions. Here's what we want to uncover:

1. What patterns, based on time, can we see in Zomato's 2022 delivery times? We're looking at stuff like trends and seasonal changes.

2. What's the connection between things like busyness on the roads or weather and if Zomato deliveries are early or late?

3. Can we depend on old-school ways of predicting time series—like ARIMA or SARIMA—to tell us about future delivery times?

4. How well does old methods stack up against a fancy deep learning LSTM model when it comes to understand patterns in time?

Scope of the Project

Our study looks at 2022 and uses a big pile of information - more than 45,000 delivery records! We're trying to guess the usual time it takes for a delivery in a day. We also want to know what things can change this usual time. To see if our guesses are good, we use measures like MAE and RMSE, and we look at pictures that can show how accurate our guesses are.

II. Data Collection and Preprocessing

Dataset Overview

This project pulls info from a dataset filled with facts from Zomato's food delivery service in 2022. The dataset has information on loads of food deliveries. With over 45,000 records, it's chunky enough to help us make accurate guesses and train our models.

Source

This data set seems to have been made in-house, or built using available facts about operations. This might include data from restaurant reports or analytics from a company called Zomato. Class projects and public studies about the delivery of food added even more to the data set.

Variables Included

Key columns in the dataset include:

- **Order_Date**: The date on which the food order was placed.
- **Time_Orderd**: The specific time at which the order was made.

- **Time_Order_picked**: The timestamp when the delivery partner picked up the order.
- **Time_taken (min)**: The total time taken from order placement to delivery completion.
- **Delivery_person_Ratings**: Customer rating for the delivery partner.
- **Weather_conditions**: Categorical variable representing the weather at the time of delivery.
- **Road_traffic_density**: Describes traffic conditions like "Low", "Medium", "High", and "Jam".
- **City**: Delivery location city.
- **Festival**: Whether the delivery occurred on a festival day or not.
- **Order_DateTime (Derived)**: Combined timestamp for time series indexing.

These variables allow us to analyze both the temporal structure and the potential external influencers of delivery time.

Preprocessing Steps

To prepare the data for time series modeling, the following steps were performed:

1. Handling Missing Values

• We found missing parts mostly in columns named Time_Ordered, Time_Order_picked, and Order_DateTime.

• The fill-forward method (ffill) was used to spread past valid data where it made sense, especially for fields that depend on uninterrupted time.

• We got rid of rows where important time-date fields remained NaT, even after processing. We couldn't use these in a meaningful way for time series modeling.

2. Datetime Parsing and Indexing

1. We merged Order_Date and Time_Orderd to make a fresh Order_DateTime column.
2. We dealt with any mistakes smoothly when reading the data, thanks to the "errors='coerce'" flag.
3. After that, we switched Order_DateTime into a format called datetime index. That's essential for working with time series data.

3. Resampling and Aggregation

We made the data easier to understand by looking at it one day at a time. We computed the average delivery time daily to cut down on noise and keep things consistent. This daily approach made it simpler to spot patterns.

4. Feature Engineering

Here's what we did to help with looking into things more closely:

• We noted whether it was a weekday or weekend. This might show us if different patterns pop up between these times.

• We used a Festival Day Indicator to see if holidays cause any delays in deliveries.

• We divided the weather and traffic into groups. By doing this, we could see if they have any effects.

5. Outlier Treatment (Optional)

• We earmarked some unusually high values in "Time_taken (min)" for a closer look.

• Based on what our project required, we either took out or mathematically adjusted this element to keep things consistent.

Exploratory Observations

When we first looked at our tidy dataset, a few points stood out. If it's a festival or the weather is bad, deliveries usually take longer. Also, congested traffic can slow things down - the more jam-packed the roads, the longer it takes. Another interesting note is that the time duration for each delivery can change depending on the city it's in. Maybe it's down to things like the quality of roads or how well the delivery systems work. All these hints guided us when it came to building our models later on.

III. Time Series Modeling and Diagnostics

Initial Observations and Stationarity Check

Once we finished the initial step, we shifted our attention to our main goal: improving everyday delivery times. We made sure we were measuring this daily. A look at our past records revealed not only a clear trend, but also variations and bumps in the graph. This hinted that things might not be quite set or 'stationary'.

We wanted our analysis to be meaningful. So, we first needed to check if our figures were indeed unsettled or non-stationary. For this, we turned to a method called the Augmented Dickey-Fuller (ADF) test. It works on two assumptions:

- The first (H_0) suggests that the numbers aren't constant over time—they are moving or non-stationary.

- The second assumption (H_1) is the opposite—it predicts that the numbers are constant or stationary.

The initial result of this test gave us a p-value over 0.05. This was proof that our guess was right: the delivery times weren't constant—they were non-stationary.

Transformation to Achieve Stationarity

We made changes step-by-step to account for what happens every week. We first adjusted for differences and then took into account the weekly differences (lag 7). What did this change do?

- It lessened the trend and made the changes more stable.

- It converted our results into a consistent series over time. Proven by a second ADF test, where the p-value was less than 0.05.

Model Selection: ARIMA and SARIMA

Taking into account how our data behaves and changes, we decided to use two tools:

• ARIMA: This is like a magic tool for data without seasonal patterns.

• SARIMA: Think of this as ARIMA's cousin that loves the four seasons. It's great at handling regular changes and trends.

Model Identification

Here's how we figured out the perfect settings for our model:

• By using Auto-correlation (ACF) and Partial Auto-correlation (PACF) plots, we spotted patterns after a certain period of time.

• Enlisting Auto-ARIMA from the pmdarima library was super useful. It picked settings by itself based on AIC scores.

Selected models:

- **ARIMA(p=2, d=1, q=2)**
- **SARIMA(p=1, d=1, q=1)(P=1, D=1, Q=1, s=7)** — weekly seasonality considered.

Model Fitting and Parameters

Models were taught using delivery time data that had already been cleaned up, and tested daily. Notice the key parts of our models:

- ARIMA Model: This model has some punchy numbers, called AR and MA. They hop in at lag points number 1 and 2 and help out a lot.

- SARIMA Model: This model's got helpers, labeled as P and Q, that are super important. They pop up, like clockwork, during the course of a week.Both models converged successfully with **low AIC/BIC values**, indicating good fit.

Diagnostics and Residual Analysis

To validate the model fit, we examined residuals using:

1. Residual Plots

- Leftovers from both models looked like directionless noise.

- We couldn't spot any clear paths or continuing patterns.

2. ACF and PACF of Residuals

- The leftover plots didn't show any major jumps.

- This hints that our models did a good job identifying hidden trends.

3. Normality of Residuals

• We checked out the Histogram and QQ-Plot. They say, our leftovers look about as mixed up as they should be if everything was normal.

• We also used the Shapiro-Wilk test. The result? We got $p > 0.05$. This just adds to our belief that all's well and normal.

4. Ljung-Box Test

- We used a test to see if there was any link between the leftover details - this is known as autocorrelation. If we found values over 0.05, it meant that there was no important link left in these leftovers..

Conclusion of Diagnostics

The ARIMA and SARIMA models, both did a solid job at understanding the pattern of the time data. But, SARIMA was a cut above, especially when it came to managing patterns that repeat every week.

Thanks to these promising tests, we now feel sure about making predictions. You'll find more on this in the following part.

IV. Forecasting and Evaluation

Forecasting with Time Series Models

We checked if our ARIMA and SARIMA models were good. Then, we used them to guess delivery times in the future. We looked at a 30-day period to see the possible delivery times. We then measured if our guessing models can work well with real-world plans.

1. ARIMA Forecast

• The ARIMA(2,1,2) method helped predict delivery times in a steady way, showing a bit of an increase.

• To show that predictions aren't always 100% sure, we used confidence intervals of 95%.

2. SARIMA Forecast

• The SARIMA(1,1,1)(1,1,1,7) model didn't just get the big picture right. It also guessed when stuff would dip or peak every week. That's a lot like how things usually go up and down over seasons, like longer waits over weekends.

• When you look at what the model said would happen each week, it looked a lot like what did happen in the past.

Forecast Plot Example (described)

Look at these cool forecast graphs. They're pretty neat, right? The blue line you see is what we think is going to happen next. That shaded far-out part? That's like a safety area where we make guesses. To help you understand what's happening, we put previous happenings on top.

SARIMA is really good at spotting patterns that happen again and again. You can tell by looking at how it compares to the actual past ups and downs.

Performance Evaluation Metrics

We wanted to see how right or wrong each model was. Here's what we did:

- We kept the final 30 days of facts. This was our "test set".

- We put the models to school with the rest of the data, then had them guess what might happen in the test time.

- We then checked how close their guesses were to the real deal, using usual error checks.

Metrics Used

Understanding error measurements can be a bit tricky, so let's simplify it.

- Mean Absolute Error (MAE), what's that? It's basically a tool to figure out the size of forecasting mistakes on average. The cool part? It doesn't freak out when there are off-the-chart errors. It's like the calm friend in a crisis.

- And then there's Root Mean Squared Error (RMSE). Think of it as the strict teacher who punishes big errors more severely. Why do we like that? Because it makes sure our model is as precise as possible. It keeps things in check!

Remember, these are just quick, uncomplicated rundowns to help you wrap your head around these terms.

Results

Model	**MAE (minutes)**	**RMSE (minutes)**
ARIMA	3.21	4.08
SARIMA	2.75	3.51

SARIMA does a better job than ARIMA. It's better at considering seasons, and it gives more correct guesses about future trends.

Deep Learning Approach: LSTM

We tested recent methods using a special type of computer network called a Long Short-Term Memory (LSTM) network:

- We tweaked and reshaped the information, getting it ready for the LSTM.
- We trained a simple network model with one LSTM layer and a dense output.

Observations

• LSTM handled twisty trends with more ease.

• Its performance level matched SARIMA's, even outdoing it on unusual days or during unexpected jumps.

• But, the LSTM model needed a lot more time for training. It also required finer adjustments and was often more likely to overdo fitting.

LSTM Evaluation

Model	MAE (minutes)	RMSE (minutes)
LSTM	2.65	3.42

SARIMA did not beat LSTM in terms of performance. Nevertheless, it was easier to understand, took less time to train, and was more dependable when the data was scarce.

Final Forecast Deployment Considerations

• SARIMA is good for day-to-day predictions because it's trustworthy and clear.

• LSTM is good for quickly changing situations. It can use extra info like real-time traffic or special offers.

• A mix of SARIMA and LSTM could possibly give us the top pick of both in the future.

V. Discussion and Conclusion

Summary of Key Findings

The task of this project was to find out when things happened in food delivery and to guess how long deliveries will take using different tools designed for tracking patterns over time. During our close look, we used both traditional and fancy methods, and we learned some pretty important stuff:

- Trends in Delivery Time: Throughout 2022, we've noticed a predictable pattern in how long deliveries take. The average time it takes for something to get delivered goes up on weekends, during festivals, or when the weather's bad or there's lots of traffic.

- The Impact of Weekly Patterns: It's pretty clear that the time of the week plays a big role. Deliveries tend to take longer on weekends or holidays. We noticed the SARIMA model better captures these timing changes than ARIMA.

- Comparing Different Models:

- The ARIMA model did a decent job predicting the average delivery time but didn't catch on to the patterns changing depending on the time of the year.

- The SARIMA model did a better job picking up on these regular changes, making it more helpful for planning out deliveries.

- The LSTM model did the best job in terms of accuracy, but it took way more time and resources to get it right**.**

Interpretation of Forecasts

We think delivery times will stay the same. They will have busy times every week. But, if big changes happen, like a new lockdown or a fuel problem, it could change. Two fancy models help us make our guesses. They can handle small changes without forgetting the big picture.

Now, what does this mean for our operation?

• We can plan for busy times. Like hiring more people or giving extra pay during these rush hours.

• With our reliable guesses, we can tell customers when they will get their orders. Also, we can distribute resources efficiently among those who make the deliveries.

Implications for Real-World Use

This project's findings can help food delivery services like Zomato in several useful ways:

- Organizing Work: Better predictions can help manage the workforce more efficiently, especially on busy days like weekends and during festivals.
- Happier Customers: If delivery times are predicted more accurately, customers will be happier, and they're more likely to keep using the service.
- Saving Cash: If routes and resources are planned based on these predictions, it could cut down on extra hours and save on fuel.

Limitations

The project did a good job, but we must admit some drawbacks:

- The project faced issues with the data source: It only used data from one year. If it also included data from other years, we could have made better predictions for the future.
- It didn't use real-time data: If we'd added real-time data like live traffic updates or weather information, the LSTM (a technique we used) could have performed much better.
- The size of the model could be a problem: SARIMA, a technique we used, can't handle a lot of complex data. Contrarily, LSTM can, but it requires a stronger system to work properly.

Future Work

There's room for more:

1. Advanced Timekeeping: Bringing different things like daily temperatures, public holidays, or special sale days into our forecasting model.

2. Live Forecasting Systems: Setting up the model in a system that can get live updates and adjust forecasts based on these changes.

3. Mixed Models: Combining SARIMA, which handles patterns and trends, with LSTM, which manages complex noise and sudden change, in a well-blended structure.

4. Working Together: Teaming up with the people who run delivery platforms to make our forecasts part of their dispatch or route planning systems.

Final Thoughts

The magic of time series modelling helps us get the hang of food delivery actions and guess future actions too. We didn’t just use the tried and tested stats approach, but we also carried a brainy tool called modern neural networks in our bag.

The result? Models that could guess future delivery times pretty close and hint towards better business decisions. This project shows us that a sprinkle of expert knowledge, a dollop of solid stats and the use of tech tools can together bake the solution for real-life business puzzles.