Internship Report

Arrival to Exam Start Time Predict

A report submitted
In partial fulfillment of the requirements
for the degree of

B.Tech.
(Artificial Intelligence and Data Science)

by

Harphool Singh (23AI023)
Pradeep Kumar (23AI038)

to the

School of Technology
GATI SHAKTIVISHWAVIDYALAYA
Central University, Under Ministry of Railways
Vadodara, India– 390004

Date: 30 July 2025

SUMMER INTERNSHIP

Certificate of Completion

This is to certify that the work contained in the project titled **“Arrival To Exam Start Time”** has successfully completed by the following students under our supervision during 8-week summer internship. During the internship period the students found to be hard working, punctual and discipline and they met all the criteria satisfactorily.

No.	Student Name	Roll Number	Signature
1.	Harphool Singh	23AI023	
2.	Pradeep Kumar	23AI038	

Signature with Seal

Date: 30 July 2025

Acknowledgement

We take this opportunity to express our heartfelt gratitude to our project mentors, **Mr. Randhir Kumar** and **Mr. Sidhartha Attri**, for their continuous guidance, valuable suggestions, and constant support throughout the development of this project. Their expert insights, encouragement, and trust in our abilities greatly enhanced our learning and played a crucial role in shaping the project to its final outcome.

We extend our sincere thanks to **CRIS (Centre for Railway Information Systems)** for giving us the opportunity to work on a real-world problem faced by Indian Railways. The exposure to actual data and operational challenges helped us develop a deeper understanding of the railway domain and apply our knowledge of artificial intelligence and data science in a practical environment.

We are also grateful to the **Department of Artificial Intelligence and Data Science, Gati Shakti Vishwavidyalaya**, for providing the academic foundation, mentorship, and necessary infrastructure throughout the project duration.

Finally, we thank all individuals—friends, peers, faculty, and railway staff—who directly or indirectly contributed to the successful completion of this project. Your encouragement and support made this journey both meaningful and memorable.

Harphool Singh (23AI023)

Pradeep Kumar (23AI038)

Abstract

This project focuses on a real issue in Indian Railways — the delay between a train rake's arrival at a railway yard and the start of its safety examination. Every rake (a group of coaches or wagons) must be checked before it can move again, but this check does not always start immediately after arrival. This waiting time is called the "Arrival to Exam Start" delay.

The delay happens due to several reasons — overcrowded yards, not enough staff to do the checks, and poor planning. Sometimes, too many rakes are present at the same time, and there's no clear idea of when each exam can begin. These delays lead to longer train turnaround times and poor use of yard space and manpower.

To solve this, our project uses past data and machine learning to predict how much delay a new rake might face before its exam starts. The goal is to help railway staff plan better by giving early estimates of the expected waiting time.

With the help of this model, Indian Railways can manage yards more efficiently, reduce idle time, and improve train movement. It's a step toward smarter, data-based railway management using AI and real operational data.

Contents

List of Figures

1. Introduction

1.1 Motivation

In Indian Railways, once a train rake (a group of coaches or wagons) arrives at a yard, it must undergo a safety examination before it can be used again. However, this examination often gets delayed due to factors like congestion, staff shortage, and poor scheduling. These delays impact overall efficiency, train punctuality, and yard planning. Motivated by this issue, our project aims to predict this "Arrival to Exam Start" delay using machine learning models, helping Indian Railways improve their operations.

This prediction can support better manpower planning and help avoid unnecessary rake hold-ups. It also promotes data-driven decisions that can lead to smarter railway yard management.

1.2 Objectives

The main goals of this project are:

- To build a predictive model that estimates how long it takes for a rake to begin examination after arrival.
- To help railways manage yard congestion by providing better visibility into potential delays.
- To optimize allocation of resources such as staff and space.
- To reduce idle time and improve overall efficiency.

1.3 Organization of Report

This report is structured into multiple sections:

- Introduction explains the problem and motivation.
- Methodology covers the tools used, data handling, model development, and challenges faced.
- Results and Discussion presents model performance and insights.
- Conclusion summarizes the outcomes and suggests future improvements.

1.4 Organization Information

This project was developed under the guidance of CRIS (Centre for Railway Information Systems), an organization under Indian Railways. The mentors for this project were **Mr. Randhir Kumar** and **Mr. Sidhartha Attri**, whose expertise and continuous support helped shape the direction and outcomes of this work.

2. Methodology

2.1 Overview of the Approach

We used historical operational data of Indian Railways, processed it to extract meaningful features, and applied machine learning models to predict the delay. The model was trained and tested using various algorithms, and the best-performing one was selected.

2.2 Tools and Technologies Used

- Python: For data analysis and modeling.
- Pandas & NumPy: Data preprocessing.
- Matplotlib & Seaborn: Data visualization.
- Scikit-learn: Model pipelines and evaluation.
- XGBoost, LightGBM, CatBoost: Regression modeling.
- VS Code & Jupyter Notebook: Development environments.

2.3 System Architecture

The project follows a pipeline architecture:

- Data loading and cleaning
- Feature extraction (e.g., arrival hour, congestion count)
- Guided ordinal encoding for categorical features
- Model training using hyperparameter tuning
- Model evaluation on test data
- Prediction and visualization of results

2.4 Data Collection and Preprocessing

- Source: Indian Railways dataset provided by CRIS
- Size: 5,87,028 lakh records (Dec 2022 to July 2025)
- Key Fields: yard_id, rake_type, rake_id, creation_date, bpc_categ, arrival_times, bpc_no., exam_start_time, dispatch_time, notification_no..
- Cleaning Steps:
    1. Removed duplicates based on rake_id and creation_date
    2. Dropped 2022 records due to irregularities
    3. Removed extreme outliers using IQR (Interquartile Range) Method per yard wise
- Final Size: ~3.35 lakh cleaned rows

2.5 Model Implementation and Optimization

- Guided Ordinal Encoding was applied to encode categories based on target mean

- Multiple models were tested:
  RandomForest, XGBoost, LightGBM, CatBoost, Staking
- Hyperparameter tuning was done using RandomizedSearchCV
- Final evaluation was based on MAE and R^2 score

2.6 Development Workflow

1. Data cleaning and feature engineering
2. Splitting into train-test sets
3. Applying encoding and scaling
4. Model training with cross-validation
5. Performance comparison and final selection

2.7 Challenges and Mitigations

- Missing or incomplete data: Handled using NA filtering and feature imputation
- Categorical noise: Addressed using guided encoding
- Outliers: Cleaned using yard-wise IQR filtering
- Imbalanced rakes: Distribution analyzed and grouped

2.8 Scalability and Deployment Considerations

The final model is scalable for deployment across multiple yards. The code can be adapted for daily predictions if real-time data is available. Further integration with CRIS dashboards is possible for live monitoring.

3. Exploratory Data Analysis (EDA)

Before building and training the prediction model, we explored the dataset to understand its structure and behaviour. This process is known as Exploratory Data Analysis (EDA). It helped us identify the key patterns, important features, delays, and data quality issues.

The dataset covered over **5.87 lakh records** from December 2023 to July 2025. After consolidation, where duplicate entries were removed based on rake_id and creation_date, the dataset was reduced to **3.73 lakh rows**. Following further data cleaning — including removal of invalid values and extreme outliers — the final dataset used for model training contained around **3.35 lakh rows**.

To understand the patterns in our dataset, we performed Exploratory Data Analysis (EDA) using visualizations. The key focus was to explore how different yard IDs, rake types, and BPC (Brake Power Certificate) categories impact the delay between arrival and exam start.

3.1 Top 25 Yards by Highest & Fastest Arrival-to-Exam Start Time

The first graph shows the **1** inefficiencies. On the other hand, yards like JNPT and FZR show much lower delays, which implies better yard planning or higher resource availability.

The second graph highlights the **Top 25 Yards with the Fastest Arrival-to-Exam Start Time**. Yards such as BSP, DER, and CCJSYD consistently start examinations within 20 minutes of arrival. This efficient handling can be due to either fewer incoming rakes, better staffing, or organized scheduling systems.

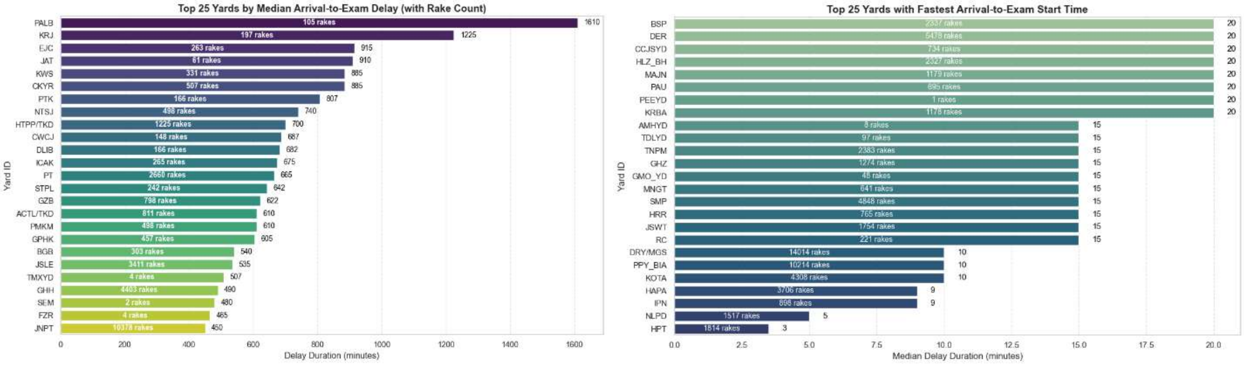


Figure 3.1 Top 25 Yards by Highest Arrival-to-Exam Start Delay and Top 25 Yards with the Fastest Arrival-to-Exam Start Time

3.2 Top 25 Rake Type by Highest & Fastest Arrival-to-Exam Start Time

These graphs analyze **Rake Types by Delay Duration**. The Top 25 rake types by average delay include BRNAHA, BWTA, and BRNHS, all showing delays above 1200–1300 minutes. These types could require more time-consuming examination procedures or are prioritized lower. Conversely, rake types such as BRNM1, ACT2, and POH show very short average delays—some under 60 minutes. These might be smaller, lighter rakes or pre-cleared in some way.

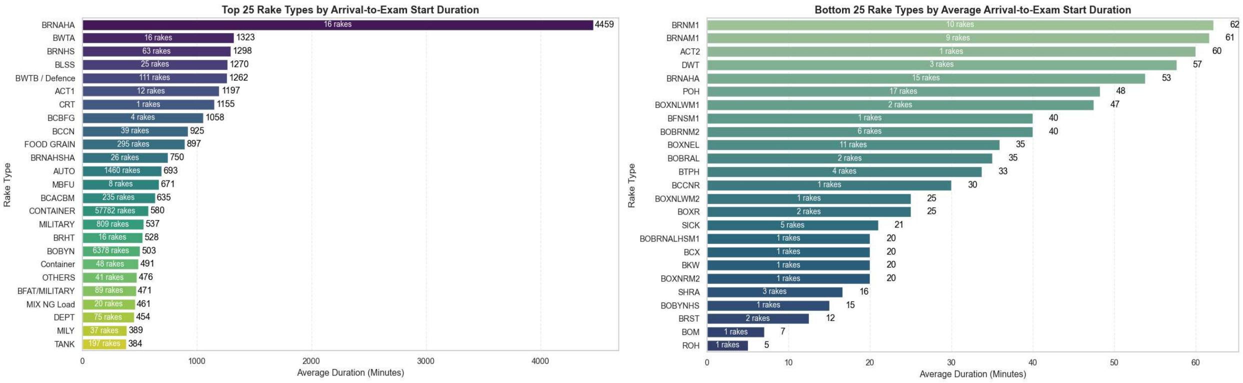


Figure 3.2 Top 25 Rake Type by Highest Arrival-to-Exam Start Delay and Top 25 Rake Type with the Fastest Arrival-to-Exam Start Time

3.3 Top BPC Categories by Arrival-to-Exam Start Time

This graph shows that the **REVALID** category has the highest average examination duration (~525 minutes), followed by **DMT_EXAM** and **CC_EXAM**. On the lower end, categories like **BCUCG_EXAM** take significantly less time (~81 minutes). This suggests that examination requirements vary considerably across different categories.

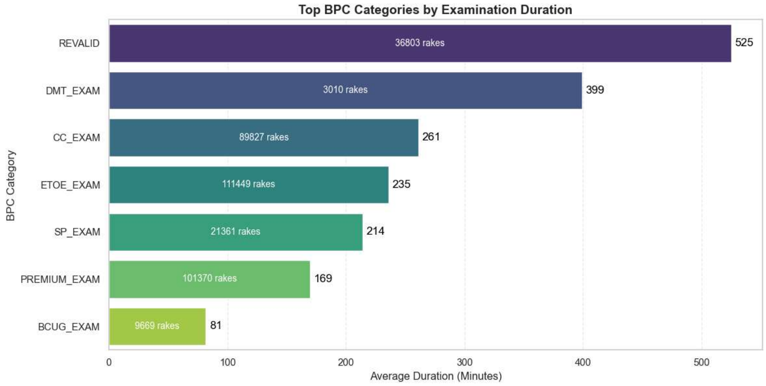


Figure 3.3 Top BPC Categories by Highest Arrival-to-Exam Start Time

3.4 Monthly Average Delay Trends

The bar chart comparing delays across months from 2023 to 2025 gives a deeper insight into seasonal trends. In general, the delays vary across months and years. For example, in July 2025, the average delay dropped to 176 minutes, which is much lower compared to July 2024 (277 minutes). On the other hand, months like September and October of 2024 show high delays of over 280 minutes. This type of analysis helps in identifying peak congestion periods and planning staff and space accordingly.

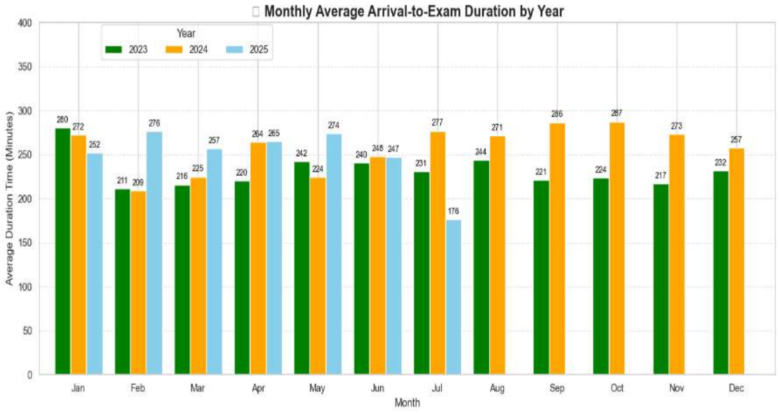


Figure 3.4: Monthly Trend of Arrival to Exam Start Duration

3.5 Weekday Delay Trends

The weekday-based analysis shows interesting patterns. Wednesday has the highest average delay of 264 minutes, while Thursday has the lowest at 234 minutes. This indicates that mid-week operations might be more crowded, possibly due to bunching of rakes or staff shortages. These insights can help in better workforce scheduling and load balancing throughout the week.

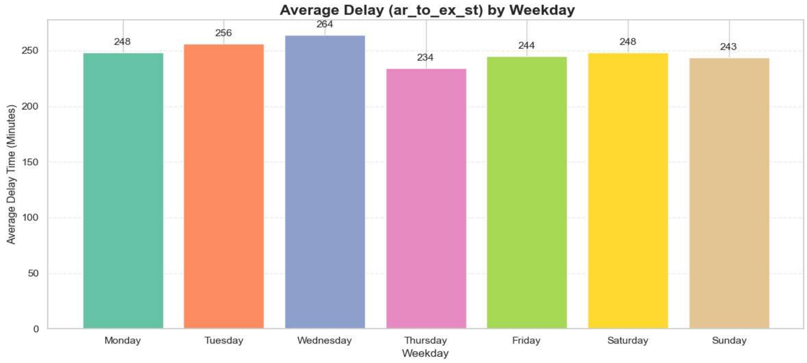


Figure 3.5: Weekday Trend of Arrival to Exam Start Duration

3.6 Daily Arrival-to-Exam Trend Analysis

The chart (Daily Analysis) presents how the average arrival-to-exam delay fluctuates on a daily basis. This time-series view shows daily variations and helps detect unusual delay spikes that could be tied to events like public holidays, staff strikes, or equipment issues. For example, a sudden rise in delay on a certain date might indicate a system bottleneck or failure. Such analysis can trigger operational reviews for specific days or events.

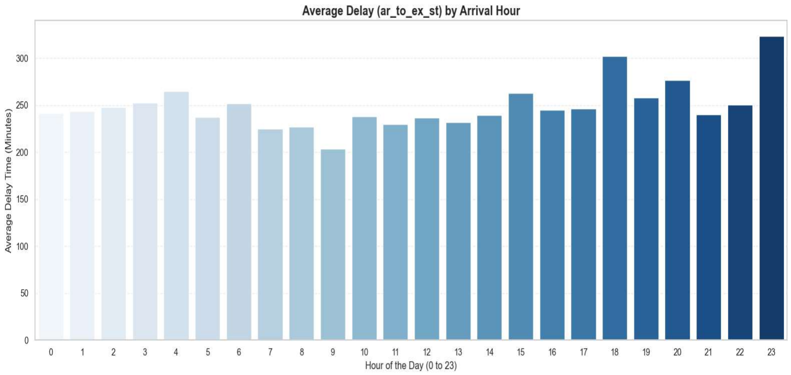
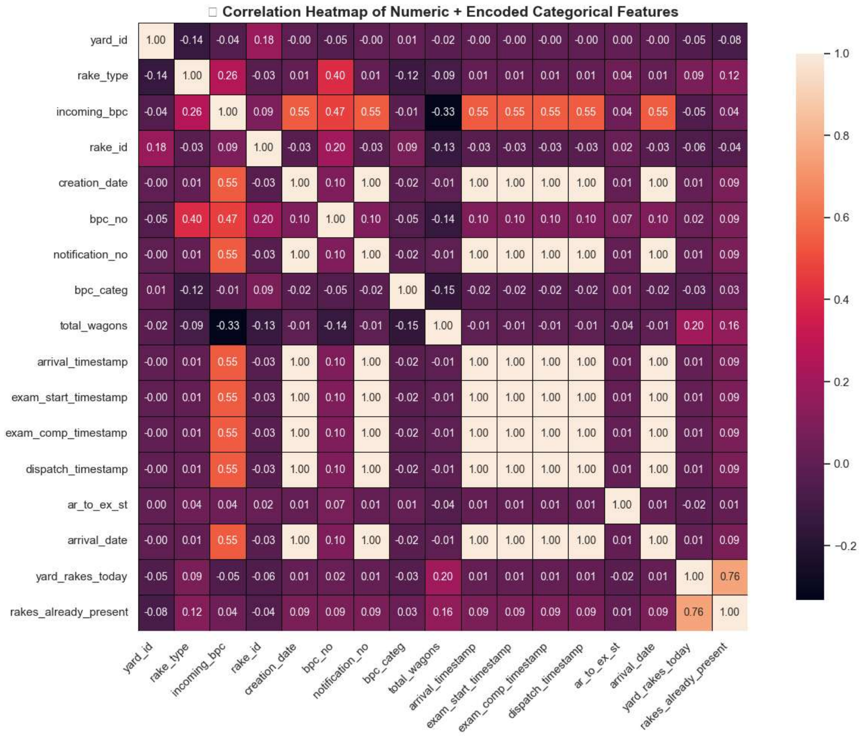


Figure 3.6: Daily Trend of Arrival to Exam Start Duration

3.7 Correlation Matrix Heatmap Analysis

In our project, the heatmap shows how the target variable, ar_to_ex_st (Arrival to Exam Start delay), is influenced by other variables like rakes_already_present, is_peak_hour, weekday, bpc_categ, and others. For example, if rakes_already_present shows a strong positive correlation with delay time, it means that more rakes in the yard lead to longer waiting times. On the other hand, features like is_weekend or month may have weaker or no clear correlation.

This analysis is useful for selecting important features for our machine learning model. By looking at which features are most closely related to our target, we can choose the best inputs and remove unrelated ones, improving model performance and reducing complexity.



	yard_id	rake_type	incoming_bpc	rake_id	creation_date	bpc_no	notification_no	bpc_categ	total_wagons	arrival_timestamp	exam_start_timestamp	exam_comp_timestamp	dispatch_timestamp	ar_to_ex_st	arrival_date	yard_rakes_today	rakes_already_present
yard_id	1.00	-0.14	-0.04	0.18	-0.00	-0.05	-0.00	0.01	-0.02	-0.00	-0.00	-0.00	-0.00	0.00	-0.00	-0.05	-0.08
rake_type	-0.14	1.00	0.26	-0.03	0.01	0.40	0.01	-0.12	-0.09	0.01	0.01	0.01	0.01	0.04	0.01	0.09	0.12
incoming_bpc	-0.04	0.26	1.00	0.09	0.55	0.47	0.55	-0.01	-0.33	0.55	0.55	0.55	0.55	0.04	0.55	-0.05	0.04
rake_id	0.18	-0.03	0.09	1.00	-0.03	0.20	-0.03	0.09	-0.13	-0.03	-0.03	-0.03	-0.03	0.02	-0.03	-0.06	-0.04
creation_date	-0.00	0.01	0.55	-0.03	1.00	0.10	1.00	-0.02	-0.01	1.00	1.00	1.00	1.00	0.01	1.00	0.01	0.09
bpc_no	-0.05	0.40	0.47	0.20	0.10	1.00	0.10	-0.05	-0.14	0.10	0.10	0.10	0.10	0.07	0.10	0.02	0.09
notification_no	-0.00	0.01	0.55	-0.03	1.00	0.10	1.00	-0.02	-0.01	1.00	1.00	1.00	1.00	0.01	1.00	0.01	0.09
bpc_categ	0.01	-0.12	-0.01	0.09	-0.02	-0.05	-0.02	1.00	-0.15	-0.02	-0.02	-0.02	-0.02	0.01	-0.02	-0.03	0.03
total_wagons	-0.02	-0.09	-0.33	-0.13	-0.01	-0.14	-0.01	-0.15	1.00	-0.01	-0.01	-0.01	-0.01	-0.04	-0.01	0.20	0.16
arrival_timestamp	-0.00	0.01	0.55	-0.03	1.00	0.10	1.00	-0.02	-0.01	1.00	1.00	1.00	1.00	0.01	1.00	0.01	0.09
exam_start_timestamp	-0.00	0.01	0.55	-0.03	1.00	0.10	1.00	-0.02	-0.01	1.00	1.00	1.00	1.00	0.01	1.00	0.01	0.09
exam_comp_timestamp	-0.00	0.01	0.55	-0.03	1.00	0.10	1.00	-0.02	-0.01	1.00	1.00	1.00	1.00	0.01	1.00	0.01	0.09
dispatch_timestamp	-0.00	0.01	0.55	-0.03	1.00	0.10	1.00	-0.02	-0.01	1.00	1.00	1.00	1.00	0.01	1.00	0.01	0.09
ar_to_ex_st	0.00	0.04	0.04	0.02	0.01	0.07	0.01	0.01	-0.04	0.01	0.01	0.01	0.01	1.00	0.01	-0.02	0.01
arrival_date	-0.00	0.01	0.55	-0.03	1.00	0.10	1.00	-0.02	-0.01	1.00	1.00	1.00	1.00	0.01	1.00	0.01	0.09
yard_rakes_today	-0.05	0.09	-0.05	-0.06	0.01	0.02	0.01	-0.03	0.20	0.01	0.01	0.01	0.01	-0.02	0.01	1.00	0.76
rakes_already_present	-0.08	0.12	0.04	-0.04	0.09	0.09	0.09	0.03	0.16	0.09	0.09	0.09	0.09	0.01	0.09	0.76	1.00

Figure 3.7: Correlation Matrix Heatmap Showing Relationships Between Key Features and Exam Start Delay

4. Outcomes / Discussion of the report

4.1 Test Scenarios

We trained machine learning models to predict the delay (ar_to_ex_st) based on features like yard_id, rake_type, bpc_categ, and other timestamp-based and congestion-based features. The dataset was split into training and testing subsets using an 80-20 split. Performance was evaluated using MAE (Mean Absolute Error) and R² Score.

4.2 Evaluation Metrics

We used two primary evaluation metrics:

- Mean Absolute Error (MAE): Measures the average absolute difference between actual and predicted delay times. Lower MAE indicates better performance.

- R^2 Score: Shows how well the model explains the variability of the target variable. An R^2 close to 1 means a strong fit.

Model	**MAE**(minutes)	**R2 Score**
RandomForestRegressor	82.93	0.5644
XGBoostRegressor	77.45	0.6245
LightGBMRegressor	87.83	0.5291
CatBoostRegressor	84.51	0.4309
Staking Regressor	93.26	0.5126

The XGBoost model gave the best results with an MAE of 77.45 minutes and R^2 score of 0.6245, showing a good balance between error and explanation. Other models like Random Forest and CatBoost performed decently but with slightly higher MAEs and lower R^2 values.

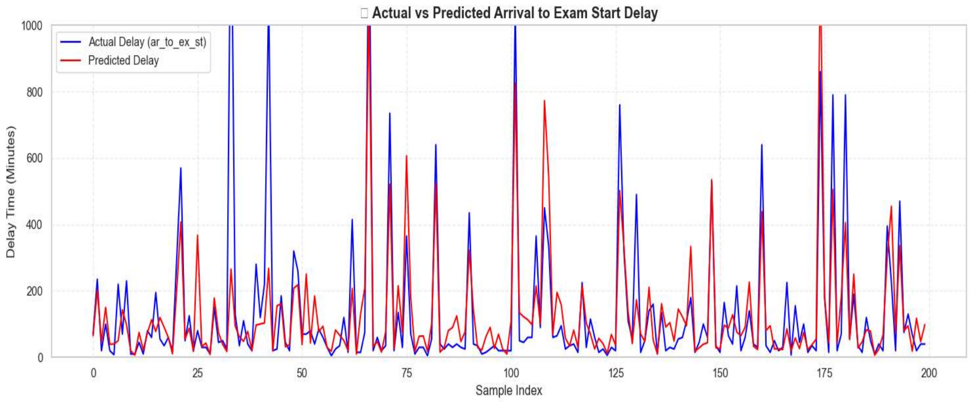


Figure 3.8: Actual vs Predict Arrival To Exam Start Time

4.3 Feature Importance and Engineering

To improve the model's prediction accuracy, several features were created and analysed Feature engineering involved extracting meaningful information from timestamp columns (like arrival and exam start times), calculating delays, and generating new features such as rakes_already_present, is_weekend.

After training the model, feature importance analysis was done to understand which features had the biggest impact on prediction. The results showed that yard_id was the most important feature. This means the location of the yard plays a key role in how quickly a rake gets examined, likely due to differences in yard size, workload, and available staff. Other important features included rake_type and arrival_hour, which also influence delays based on the type of train and time of day.

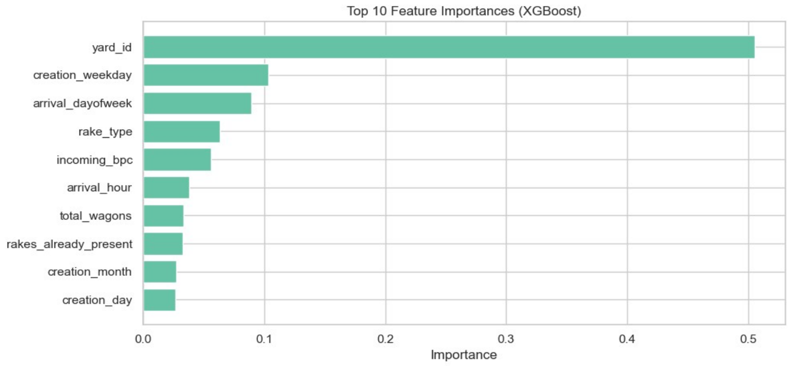


Figure 3.9: Top 10 Feature Important (XGBoost)

4.4 Limitations

One of the main limitations of this project is the model's prediction accuracy, which is around (R^2 score = 0.62). While this is a good start, it is not enough for fully reliable predictions in a real-time railway system. This lower accuracy shows that the model is not able to capture all the important patterns from the data.

Another issue is the presence of outliers in the dataset. Some data points have very large or very small delay times. These extreme values can confuse the model and reduce its ability to learn general trends. Although we applied some outlier removal techniques, some noisy data still remained.

Also, a few of the features used in the model are not strongly related to the target variable (ar_to_ex_st). These weak features add noise instead of useful information, which reduces the model's performance. More careful feature selection could help improve the results.

A big limitation is the lack of certain real-world features. Important information like staff availability, yard capacity, examination schedules, or weather conditions were not included in the dataset. These factors can greatly affect delays, but the model could not consider them due to unavailability.

The model is also built on historical data, so it assumes that future conditions will be similar to past trends. However, railway operations may change due to policy updates, staffing changes, or new infrastructure. This makes it difficult for the model to adapt to sudden or unexpected changes.

Lastly, some manual errors or missing values in the original dataset may have affected the quality of training. Even after cleaning, a few inaccurate or incomplete entries could have influenced the learning process.

5.Implications and Policy Recommendations

5.1 Operational Implications for Arrival to Exam Start (ar_to_ex_st)

This project helps railway authorities better understand how delays occur between a rake's arrival and the start of its examination. By using historical data and machine learning, we can now estimate how long this delay might be, even before the rake arrives. This can help yard officers and control rooms plan their resources more efficiently.

For example, if we know that a particular rake is likely to face a long delay, steps can be taken in advance—like calling extra staff or freeing up space in the yard. It also helps in avoiding bottlenecks, where too many rakes are waiting at the same time. This system gives real-time insights into workload, which is important for daily scheduling and smooth operations.

Another important impact is improved turnaround time. If examinations start sooner, rakes can leave the yard earlier and reach their next destination on time. This supports overall punctuality in freight and passenger movement.

5.2 Policy Recommendations

Based on the findings from this project, the following policies or decisions can be recommended to improve efficiency:

- Implement Predictive Systems: Indian Railways should consider adopting AI-powered dashboards in yards to show expected exam start delays for each incoming rake. This can help in real-time decision-making.
- Data Collection Policies: More structured data collection is needed—such as logging the number of staff on duty, shift timing, and availability of examination lines. Including these features can greatly increase model accuracy.
- Flexible Resource Allocation: Policies should support dynamic manpower deployment. If one yard is under pressure, extra staff should be allowed to move temporarily from nearby yards.
- Training for Yard Officers: Staff should be trained to understand and use AI tools so that they can take data-based actions rather than relying only on manual judgment.
- Central Monitoring System: A central platform can be developed to monitor arrival to exam delays across all major yards. This will give a clear picture of where problems are happening regularly.

5.3 Future Scope

There is a lot of scope to improve and expand this project in the future. Some possible ideas include:

- Add More Contextual Features: In the future, we can include data such as available manpower, weather conditions, yard maintenance schedule, or train priority level. These can improve prediction quality.
- Real-Time Integration: This model can be connected with the real-time railway system (like FOIS or WMS) so that predictions are updated automatically as rakes move.
- Deploy as a Web or Mobile App: A user-friendly app can be developed for yard managers where they can input rake details and get the estimated delay instantly.
- Expand to Other Use Cases: The same model and technique can be used to predict exam completion time, dispatch time, or even arrival-to-dispatch full cycle.
- Continual Learning: The model can be made smarter over time by automatically learning from new data and improving its performance.

6. References

- Used real data from CRIS titled "_Narrow_the_dataset_early_using_CTEs_WITH_filtered_rake_details.csv", containing rake arrivals, examination details, timestamps, rake types, yard IDs, etc.
- Referred to the Indian Railways Operating Manual by the Ministry of Railways for understanding rake handling procedures, examination schedules, and yard operations.
- Used XGBoost for model building. https://xgboost.readthedocs.io
- Used Scikit-learn for data preprocessing, training, hyperparameter tuning, and evaluation. https://scikit-learn.org/stable/documentation.html
- Used Pandas for data cleaning, timestamp handling, missing values, and group-based feature creation. https://pandas.pydata.org/docs
- Used Seaborn and Matplotlib for visualizations like delay trends, correlation heatmaps, and feature importance. https://seaborn.pydata.org | https://matplotlib.org/stable/index.html
- Used LightGBM for model comparison and boosting-based predictions. https://lightgbm.readthedocs.io
- Used CatBoost for handling categorical features and gradient boosting. https://catboost.ai/docs