

Intel® AI for Manufacturing Certification Course

Project Report

Predictive Lead Conversion Using Metadata

Organisation: Tara Metal Industries, Ahmedabad



Group ID: G00060

Submitted By:

Memon Mohammad Ayan Anwar
Paresh Gorakh Patil

Submission Date: 29 – 06 – 2025

Acknowledgement

We sincerely thank the organizing team of the **Intel® AI for Manufacturing Certification Course**, supported by **Intel, Gujarat Chamber of Commerce and Industry (GCCCI)**, and the **Digital Readiness Team**, for providing us with the opportunity to work on this industry-relevant AI project.

We are grateful for the structured learning resources, hands-on labs, and real-world problem statements that helped us gain practical experience in applying artificial intelligence to manufacturing and marketing domains.

We would also like to express our appreciation to the faculty coordinators and mentors for their support, guidance, and valuable feedback throughout the development of this project.

Abstract

This project presents a machine learning approach to improve lead conversion rates for businesses by analyzing user metadata. Despite high website traffic, many businesses struggle with low conversion rates, often as low as 1.5%. This project leverages historical lead data including engagement behavior, source of traffic, time spent, and device usage to predict whether a lead is likely to convert. Using logistic regression as the predictive model, the solution was developed in Python, evaluated on real data, and deployed using Streamlit for user interaction. The resulting application allows sales or marketing teams to enter new lead details and receive an instant prediction on the likelihood of conversion. The project demonstrates how AI can provide actionable insights that improve marketing efficiency and reduce manual prioritization efforts.

Table of Content

Sr. No.	Title	Page No.
*	Acknowledgement	I
*	Abstract	II
*	Table of Content	III
1	Project Overview	1
	1.1 - Introduction	1
	1.2 - Problem Statement	1
	1.3 - Objectives	1
	1.4 - Benefits	1
	1.5 - Timeline	1
2	Methodology	2
	2.1 - Approach	2
	2.2 - Dataset Description	2
	2.3 - Data Cleaning and Preprocessing	3
	2.4 - Model Training	3
	2.5 - Deployment	3
3	Technologies Used	4
	3.1 - Programming Language	4
	3.2 - Libraries and Frameworks	4
	3.3 - Development Tools	4
	3.4 - Deployment Platform	4
	3.5 - Security & Data Protection	4
4	Results	5
	4.1 - Model Performance	5
	4.2 - Deployment Output	5
	4.3 - Public Access	5
5	Conclusion	6
	5.1 - Summary	6
	5.2 - Key Takeaways	6
	5.3 - Challenges Faced	6
	5.4 - Future Enhancements	6
6	References	7
7	Appendix	8
	A. Streamlit App Link	8
	B. Sample Input (via app)	8
	C. Sample Output	8
	D. Project Folder Structure (GitHub)	8

1. Project Overview

1.1 Introduction

In the modern digital economy, businesses often attract a high volume of leads through websites, ads, and social media platforms. However, only a small fraction of these leads ultimately convert into actual customers. In some cases, conversion rates are as low as 1.5%. Manually prioritizing and qualifying leads based on limited attributes is inefficient and prone to human error.

This project seeks to solve that problem using artificial intelligence and machine learning. The objective is to build a predictive model that can analyze lead metadata—such as time spent on the website, pages visited, device used, and source of traffic—and predict whether a given lead is likely to convert.

1.2 Problem Statement

Despite high website traffic and large lead databases, businesses face difficulty in identifying which leads are most likely to convert. As a result, marketing teams waste time and resources on leads that ultimately do not turn into customers.

This project aims to use AI to provide a predictive scoring mechanism that helps filter high-potential leads from low-priority ones, thus improving marketing efficiency and conversion rates.

1.3 Objectives

- To analyze a real-world dataset of lead interactions and metadata.
- To clean, preprocess, and engineer features relevant to conversion prediction.
- To build and evaluate a binary classification model to predict lead conversion.
- To deploy the model as an interactive web application using Streamlit.
- To make the solution accessible for testing and demonstration via public deployment.

1.4 Benefits

- Data-driven lead prioritization
- Increased conversion efficiency
- Reduced marketing cost per acquisition
- Faster decision-making for sales teams

1.5 Timeline

The project was carried out in the following phases:

1. Dataset selection, understanding, and cleaning
2. Model building and feature engineering
3. Evaluation and optimization
4. Streamlit deployment, testing, and documentation

2. Methodology

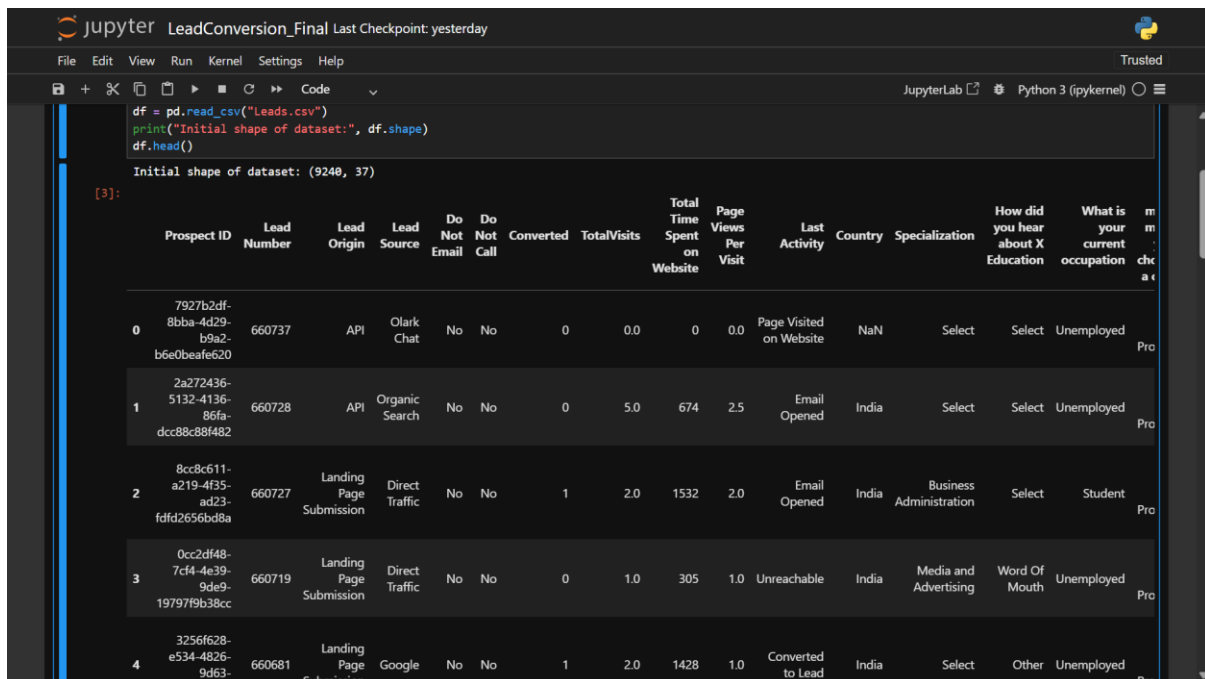
2.1 Approach

The project followed an iterative, step-by-step methodology typically aligned with the agile workflow in AI/ML model development. We began with exploratory data analysis (EDA), followed by preprocessing, modeling, evaluation, and finally deployment.

2.2 Dataset Description

The dataset was sourced from a public GitHub repository containing structured lead data, including the following columns:

- Lead Origin
- Lead Source
- Do Not Email
- Converted (Target variable)
- Total Time Spent on Website
- Page Views Per Visit
- Last Activity
- Country
- Specialization
- Occupation
- Tags
- Lead Quality
- and others



```
df = pd.read_csv("Leads.csv")
print("Initial shape of dataset:", df.shape)
df.head()
```

Initial shape of dataset: (9240, 37)

	Prospect ID	Lead Number	Lead Origin	Lead Source	Do Not Email	Do Not Call	Converted	TotalVisits	Total Time Spent on Website	Page Views Per Visit	Last Activity	Country	Specialization	How did you hear about X Education	What is your current occupation	m
0	7927b2df-8bba-4d29-b9a2-b6e0beafe620	660737	API	Olark Chat	No	No	0	0.0	0	0.0	Page Visited on Website	NaN	Select	Select	Unemployed	Pro
1	2a272436-5132-4136-86fa-dcc88c88f482	660728	API	Organic Search	No	No	0	5.0	674	2.5	Email Opened	India	Select	Select	Unemployed	Pro
2	8cc8c611-a219-4f35-ad23-fdfd2656bd8a	660727	Landing Page Submission	Direct Traffic	No	No	1	2.0	1532	2.0	Email Opened	India	Business Administration	Select	Student	Pro
3	0cc2df48-7cf4-4e39-9de9-19797f9b38cc	660719	Landing Page Submission	Direct Traffic	No	No	0	1.0	305	1.0	Unreachable	India	Media and Advertising	Word Of Mouth	Unemployed	Pro
4	3256f628-e534-4826-9d63-	660681	Landing Page Submission	Google	No	No	1	2.0	1428	1.0	Converted to Lead	India	Select	Other	Unemployed	Pro

The dataset contained over 9000 rows and 30+ columns, some of which had missing or non-informative data.

2.3 Data Cleaning and Preprocessing

- **Handling Null Values:** Dropped or imputed missing values based on frequency or default values.
- **Categorical Encoding:** Used one-hot encoding and label encoding as applicable.
- **Feature Selection:** Removed redundant or low-variance features after correlation analysis.
- **Target Variable:** ‘Converted’ was used as the binary target (1 = converted, 0 = not converted).

2.4 Model Training

We selected a simple yet interpretable model: **Logistic Regression**. The model was trained on an 80:20 split of the dataset using stratified sampling to preserve class distribution.

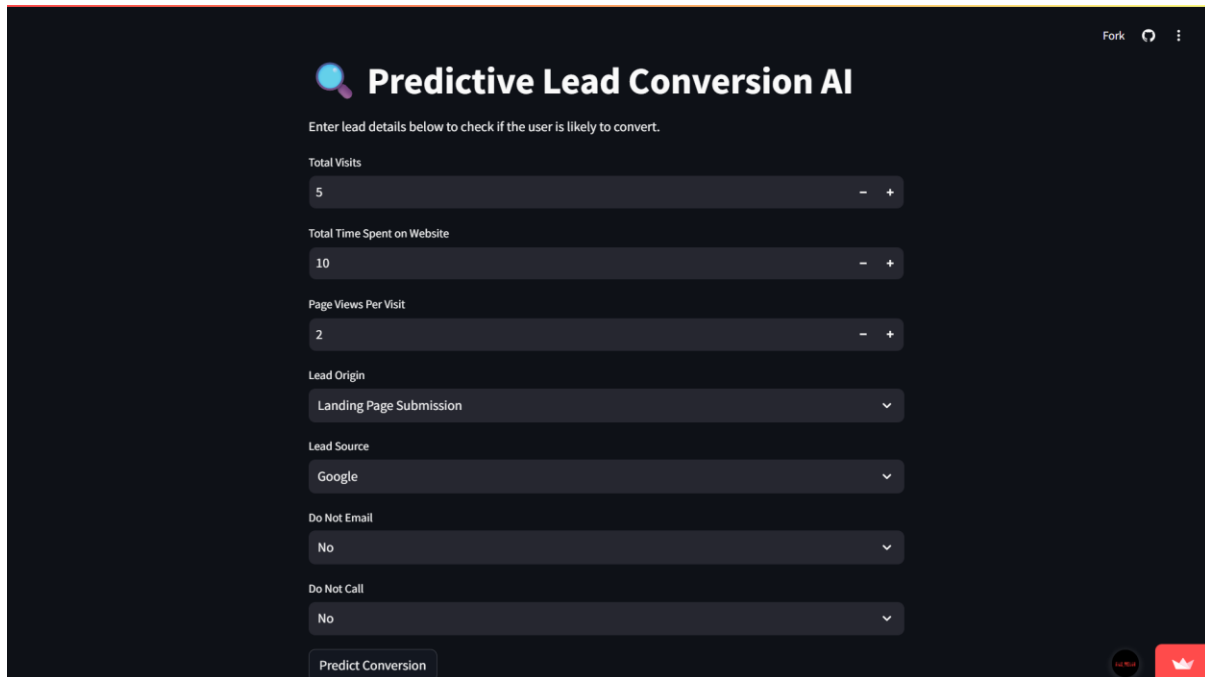
- **Training Score:** Evaluated on accuracy, precision, recall, and F1-score.
- **Cross-Validation:** Used basic K-Fold for verifying model consistency.

2.5 Deployment

The trained model was exported using `pickle` and integrated into a Streamlit application for real-time user input and prediction display.

- Input: Form-based metadata entries (lead source, time on site, etc.)
- Output: A prediction result showing whether the lead is likely to convert

The application was deployed on **Streamlit Cloud** and is publicly accessible.



The screenshot shows a web application titled "Predictive Lead Conversion AI" with a dark theme. It features a form for inputting lead details to check conversion likelihood. The form includes the following fields and values:

- Total Visits: 5
- Total Time Spent on Website: 10
- Page Views Per Visit: 2
- Lead Origin: Landing Page Submission
- Lead Source: Google
- Do Not Email: No
- Do Not Call: No

At the bottom of the form is a "Predict Conversion" button. The application interface also includes a "Fork" button and a GitHub logo in the top right corner, and a Streamlit logo in the bottom right corner.

3. Technologies Used

3.1 Programming Language

- **Python 3.10**
Used for the entire development process including data preprocessing, model training, and application development.

3.2 Libraries and Frameworks

- **Pandas** – Data manipulation and analysis
- **NumPy** – Numerical computations
- **Scikit-learn** – Machine learning algorithms and model evaluation
- **Streamlit** – Interactive web application development
- **Pickle** – Model serialization and saving
- **Matplotlib / Seaborn** – Visualizations (optional for EDA)

3.3 Development Tools

- **Jupyter Notebook** – Used for code writing, EDA, and testing
- **Git & GitHub** – Version control and public repository hosting
- **VS Code / Text Editor** – For editing Streamlit and deployment files

3.4 Deployment Platform

- **Streamlit Cloud**
Used to host the deployed web app and make it accessible for demo or user testing.

3.5 Security & Data Protection

- No sensitive data or personal identifiers were present in the dataset.
 - The deployed application accepts user input only for demonstration purposes.
 - No external APIs or databases are integrated, hence data is handled locally during prediction.
-

4. Results

4.1 Model Performance

The Logistic Regression model performed effectively on the cleaned dataset. The following are the key evaluation metrics on the test set:

- **Accuracy:** 77.86%
- **Precision (Class 1):** 76%
- **Recall (Class 1):** 72%
- **F1 Score (Class 1):** 74%

```
## 7. Evaluate the Model
y_pred = model.predict(X_test)
print("Accuracy Score:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

Accuracy Score: 0.7786802030456853

Confusion Matrix:

```
[[459  96]
 [122 308]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.79	0.83	0.81	555
1	0.76	0.72	0.74	430
accuracy			0.78	985
macro avg	0.78	0.77	0.77	985
weighted avg	0.78	0.78	0.78	985

The model showed balanced performance across both classes, indicating that it can reasonably distinguish between converting and non-converting leads.

4.2 Deployment Output

The model was successfully integrated into a functional Streamlit app that allows users to input new lead information and view the predicted result in real time.

- **Prediction Output:** Displays “Yes” or “No” indicating the conversion likelihood
- **User Interface:** Clean, form-based input panel with one-click prediction

4.3 Public Access

The application is publicly accessible at:

<https://predictive-lead-conversion.streamlit.app>

This allows stakeholders, mentors, and evaluators to interact with the model directly.

5. Conclusion

5.1 Summary

This project demonstrated how artificial intelligence can be applied to solve a real-world problem in digital marketing and sales. By using historical lead metadata, a machine learning model was developed to predict the conversion potential of leads.

5.2 Key Takeaways

- Preprocessing plays a critical role in model performance, especially when dealing with real business data.
- Logistic Regression, despite being simple, provided strong and interpretable results.
- Streamlit is a powerful tool for deploying models and making them accessible to non-technical users.
- Proper version control and library compatibility are crucial during deployment.

5.3 Challenges Faced

- **Missing Data:** Many fields had null values, which required careful imputation or removal.
- **Imbalanced Target Variable:** The 'Converted' class was skewed, so performance tuning was needed.
- **Deployment Errors:** Streamlit Cloud initially faced issues with Python version and package compatibility, which were resolved by adjusting `requirements.txt`.

5.4 Future Enhancements

- Add advanced models such as XGBoost or Random Forest for comparison
 - Include probability-based scoring instead of binary labels
 - Integrate a real-time database for lead inputs and storage
 - Provide analytics dashboard features within the app
-

6. References

1. **Dataset Source:**
GitHub Repository – [Predictive-Lead-Conversion](#)
Dataset File – `Leads.csv`
 2. **Libraries & Tools:**
 - Scikit-learn: <https://scikit-learn.org>
 - Streamlit: <https://streamlit.io>
 - Pandas: <https://pandas.pydata.org>
 - NumPy: <https://numpy.org>
 3. **Deployment Platform:**
 - Streamlit Cloud: <https://streamlit.io/cloud>
 4. **Model Saving:**
 - Pickle module: <https://docs.python.org/3/library/pickle.html>
-

7. Appendix

A. Streamlit App Link

The deployed app can be accessed at:

<https://predictive-lead-conversion.streamlit.app>

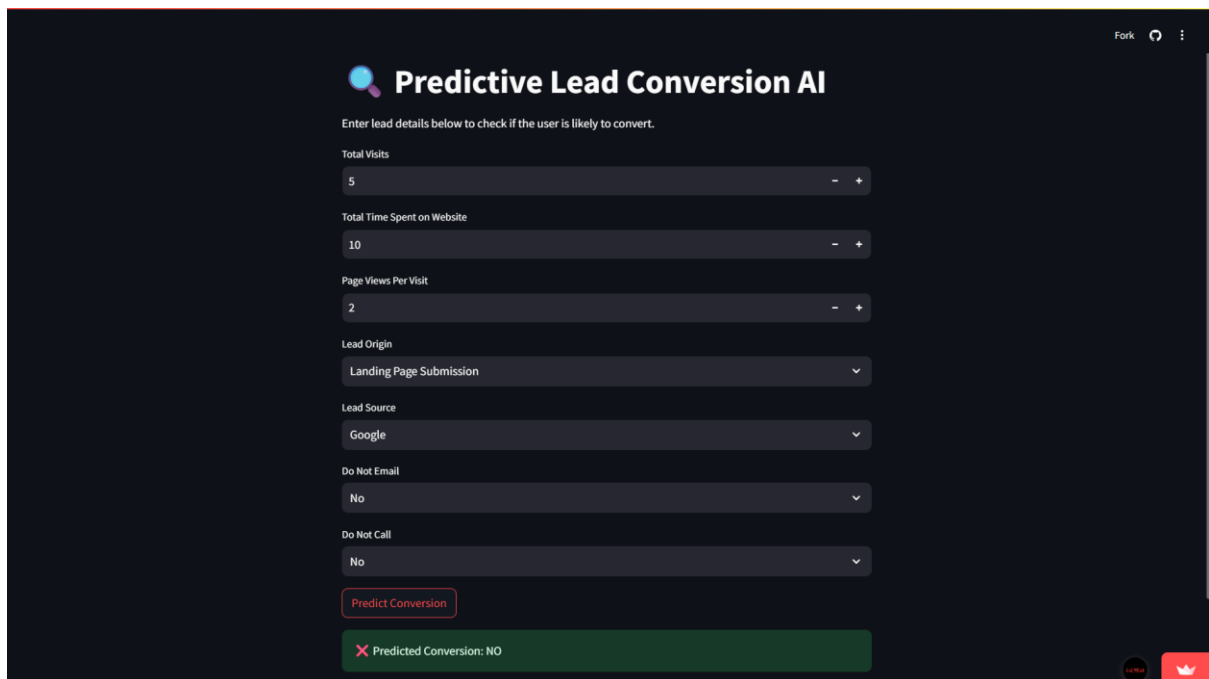
GitHub Link:

<https://github.com/AyanMemon296/Predictive-Lead-Conversion>

Presentation (PPTX) Link:

[https://github.com/AyanMemon296/Intel-AI-Certification/tree/main/Project/Predictive-Lead-Conversion/Predictive Lead Conversion Presentation.pdf](https://github.com/AyanMemon296/Intel-AI-Certification/tree/main/Project/Predictive-Lead-Conversion/Predictive%20Lead%20Conversion%20Presentation.pdf)

B. Sample Input (via app)



C. Sample Output

Predicted Conversion: NO

D. Project Folder Structure (GitHub)

```
Predictive-Lead-Conversion/  
├── Leads.csv  
├── LeadConversion_Final.ipynb  
├── app.py  
├── lead_model.pkl  
├── model_columns.pkl  
├── requirements.txt  
└── README.md
```