**🧠 Project Walkthrough: Predicting Trial-to-Paid Conversion**

Welcome! In this project, we’ll work on predicting whether a user will become a paying customer after trying out a SaaS product. This project is inspired by real business problems and designed completely **within your syllabus** (Excel, Python, Data Analysis, and Intro to ML).

Let’s walk through it step by step!

**Step 1: What's the Problem We're Solving?**

**Imagine this:**  
A company offers a free trial for its product. Lots of users sign up, but only a few actually end up paying.

**So the question is:**  
Can we predict — early on — who is likely to convert into a paid customer?

**Why it matters:**

* Saves the company time and money
* Helps marketing and sales teams focus on high-potential leads
* Improves overall conversion rates

**Step 2: What Kind of Data Do We Have?**

We've already cleaned the dataset for you. Here's what's inside:

| **Column** | **What It Means** |
| --- | --- |
| user\_id | Unique ID for each user |
| signup\_days\_ago | How long ago the user signed up |
| num\_logins | How many times they logged in |
| time\_spent\_minutes | Total time spent on the platform |
| features\_used | Number of features/tools they used |
| invited\_others | Did they invite others? |
| clicked\_email | Did they engage with marketing emails? |
| support\_tickets | How many times they contacted support |
| company\_size | Team size (Small or Medium) |
| is\_converted | 1 = Converted, 0 = Not Converted |

**Step 3: Tools We’ll Use (All From Your Syllabus!)**

| **Task** | **Tool Used** |
| --- | --- |
| Data Handling | Excel, Pandas |
| Data Visualization | Matplotlib, Seaborn |
| Statistics & Cleaning | Python, Numpy |
| ML Modeling | Logistic Regression, Decision Tree |

**Step 4: Project Blueprint (Our Plan of Action)**

Here’s the journey we’ll take:

1. Load the dataset into Python
2. Explore the data (basic statistics & checks)
3. Visualize key patterns
4. Prepare the data for modeling
5. Build a simple machine learning model to predict conversion
6. Test how accurate our model is

**Step 5: Let’s Get to Work — Code Time!**

**🔹 Load the Data**

python

CopyEdit

import pandas as pd

df = pd.read\_excel("cleaned\_trial\_to\_paid\_dataset.xlsx")

df.head()

**🔹 Check the Data**

python

CopyEdit

df.info()

df.isnull().sum()

# If there are missing values

df.dropna(inplace=True)

**🔹 Visualize the Patterns**

python

CopyEdit

import seaborn as sns

import matplotlib.pyplot as plt

sns.countplot(x="is\_converted", data=df)

plt.title("Who Converted and Who Didn't?")

plt.show()

**🔹 Prepare the Data (Encode Text Fields)**

python

CopyEdit

df["company\_size"] = df["company\_size"].map({"Small": 0, "Medium": 1})

**🔹 Build the Logistic Regression Model**

python

CopyEdit

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import classification\_report

X = df.drop(columns=["user\_id", "is\_converted"])

y = df["is\_converted"]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2)

model = LogisticRegression()

model.fit(X\_train, y\_train)

predictions = model.predict(X\_test)

print(classification\_report(y\_test, predictions))

**🔹 (Optional) Try a Decision Tree**

python

CopyEdit

from sklearn.tree import DecisionTreeClassifier

tree = DecisionTreeClassifier()

tree.fit(X\_train, y\_train)

tree\_preds = tree.predict(X\_test)

print(classification\_report(y\_test, tree\_preds))

**Step 6: What We’ve Achieved**

* 🧹 Cleaned and analyzed a real-world dataset
* 📊 Visualized key behavior patterns
* 🧠 Built a prediction model using tools from your course
* 📈 Made predictions that can guide business decisions