# Load dataset

Df = pd.read\_csv(“/content/financial\_behavior\_dataset.csv”)

# Drop rows with missing target

Df.dropna(subset=[‘financial\_behavior’], inplace=True)

# Show dataset summary

Print(“Columns:”, df.columns.tolist())

Df.head()

[06/05, 11:59 am] Sri Sharmishta (NM) Mam: from sklearn.preprocessing import LabelEncoder

# Target

Y = df[‘financial\_behavior’]

# Drop unused text fields (you can include sentiment\_score and stats)

X = df.drop(columns=[‘financial\_behavior’, ‘tweet\_content’])

# Encode categorical features

X\_encoded = pd.get\_dummies(X, drop\_first=True)

# Encode target

Le = LabelEncoder()

Y\_encoded = le.fit\_transform(y)

[06/05, 11:59 am] Sri Sharmishta (NM) Mam: from sklearn.model\_selection import train\_test\_split

From sklearn.linear\_model import LogisticRegression

From sklearn.metrics import classification\_report

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_encoded, y\_encoded, test\_size=0.2, random\_state=42)

Model = LogisticRegression(max\_iter=1000)

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

# Evaluate

Y\_pred = model.predict(X\_test)

Print(classification\_report(y\_test, y\_pred, target\_names=le.classes\_))

[06/05, 12:00 pm] Sri Sharmishta (NM) Mam: import gradio as gr

Input\_columns = X\_encoded.columns

Def predict\_behavior(\*\*inputs):

Import pandas as pd

Input\_df = pd.DataFrame([inputs])

Input\_encoded = pd.get\_dummies(input\_df)

Input\_encoded = input\_encoded.reindex(columns=input\_columns, fill\_value=0)

Prediction = model.predict(input\_encoded)[0]

Return le.inverse\_transform([prediction])[0]

# Build Gradio interface

Input\_components = []

For col in X.columns:

If X[col].dtype == ‘object’:

Options = sorted(df[col].dropna().unique().tolist())

Input\_components.append(gr.Dropdown(choices=options, label=col))

Else:

Input\_components.append(gr.Number(label=col))

Gr.Interface(

Fn=predict\_behavior,

Inputs=input\_components,

Outputs=gr.Label(),

Title=”💸 Financial Behavior Classifier”,

Description=”Predicts financial behavior type based on social media content features.”

).launch(share=True)