

```
In [1]: %%writefile predict.py
# predict.py - prediction helper used by app.py
import joblib
import pandas as pd
from datetime import datetime

# Recreating the exact function used in the training pipeline
def extract_date_features(df_in):
    df2 = df_in.copy()
    dates = pd.to_datetime(df2["date"], errors="coerce")
    df2["dt_year"] = dates.dt.year.fillna(0).astype(int)
    df2["dt_month"] = dates.dt.month.fillna(0).astype(int)
    df2["dt_day"] = dates.dt.day.fillna(0).astype(int)
    df2["dt_dow"] = dates.dt.dayofweek.fillna(0).astype(int)
    return df2[["dt_year", "dt_month", "dt_day", "dt_dow"]]

#Loading model(the function above must exist before unpickling)
model = joblib.load("output/nb_tuned_pipeline.joblib")

def predict_category(notes, amount, payment_mode, location, transaction_type, date):
    """
    Predict category for a single transaction.
    Returns (predicted_label, top3_list_of_tuples(label, prob))
    """
    #building single-row DataFrame exactly as pipeline expects
    df = pd.DataFrame([{
        "notes": notes,
        "amount": amount,
        "payment_mode": payment_mode,
        "location": location,
        "transaction_type": transaction_type,
        "date": date
    }])
    #predicting label
    pred = model.predict(df)[0]
    #predicting probabilities
    probs = model.predict_proba(df)[0]
    classes = model.classes_
    top3_idx = probs.argsort()[-3:][::-1]
    top3 = [(classes[i], float(round(probs[i], 3))) for i in top3_idx]
    return pred, top3
```

Overwriting predict.py