

Expense Categorizer

An ML-powered app that tracks and classifies your expenses like a smart accountant.

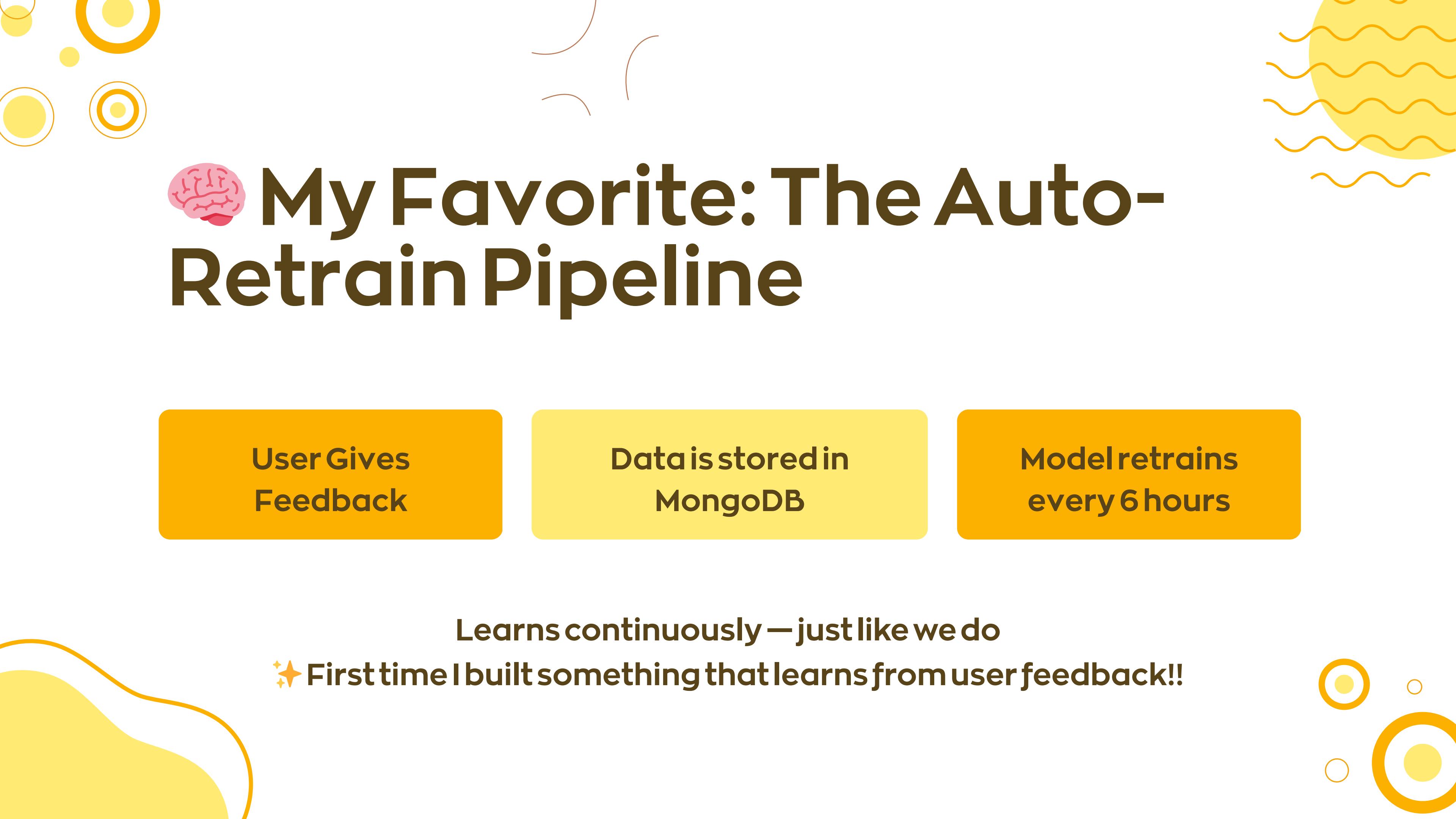
Introduction

Expense Categorizer is a smart, ML-powered web application designed to automatically classify your financial transactions into meaningful categories — helping you track where your money goes without the manual effort.

The system doesn't just make predictions — it learns continuously from user feedback, retraining itself every few hours to improve over time. Users can submit feedback on incorrect predictions, explore insightful analytics dashboards, and review their transaction history — all stored securely in MongoDB Atlas.

What can it DO?

- ✓ Auto-categorizes transactions using ML
- ⟳ Accepts user feedback to improve predictions
- 📊 Visualizes spending patterns with dashboards
- 🧠 Retrains model every 6 hours automatically
- 🏗️ Built using Flask, MongoDB, Plotly, MLflow



My Favorite: The Auto-Retrain Pipeline

User Gives
Feedback

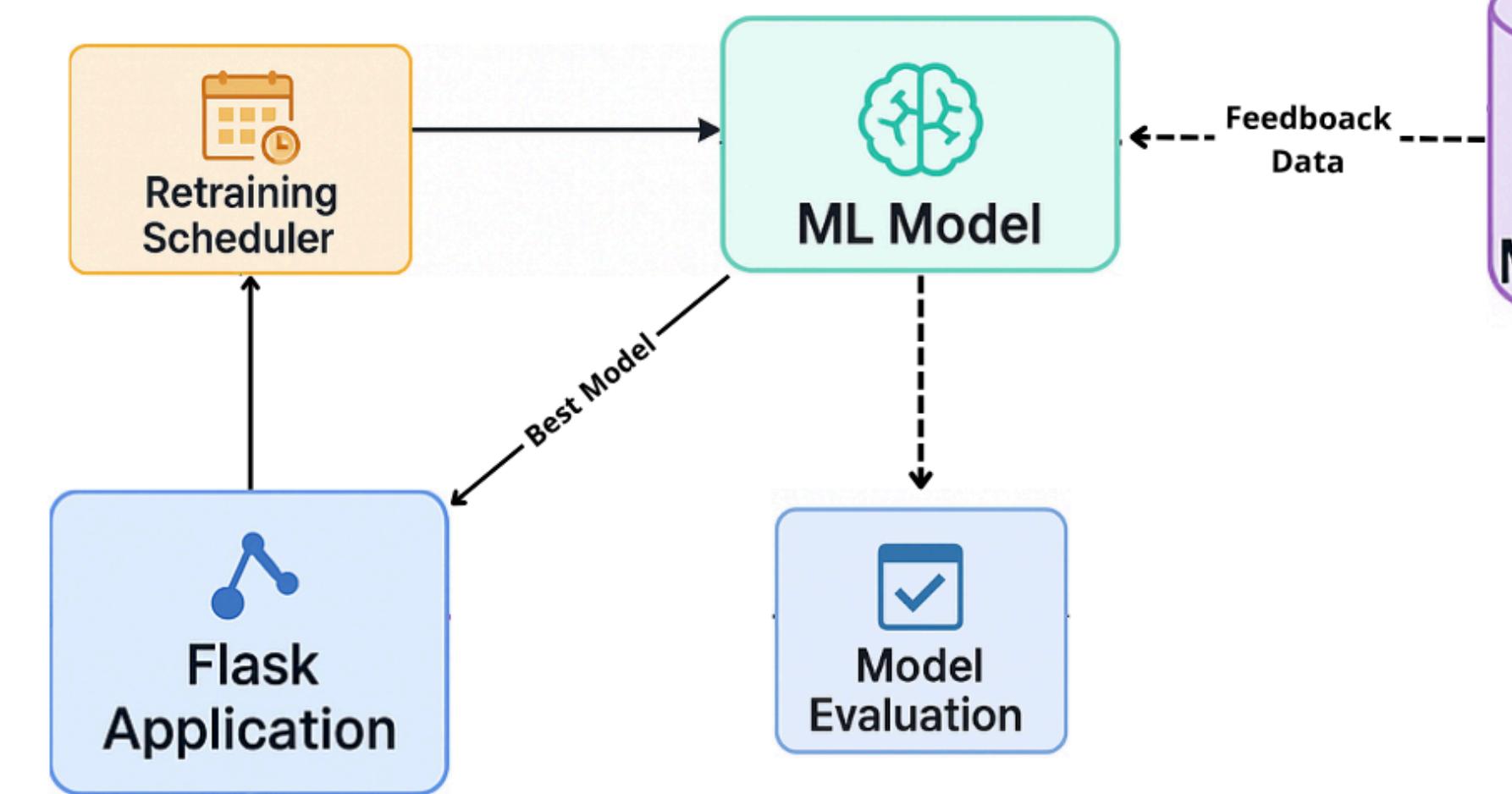
Data is stored in
MongoDB

Model retrains
every 6 hours

Learns continuously – just like we do

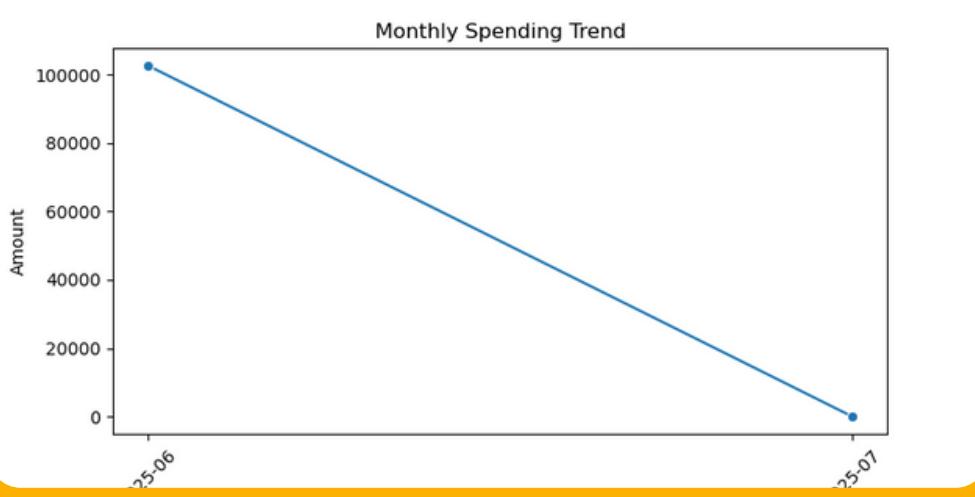
✨ First time I built something that learns from user feedback!!

Retraining Workflow

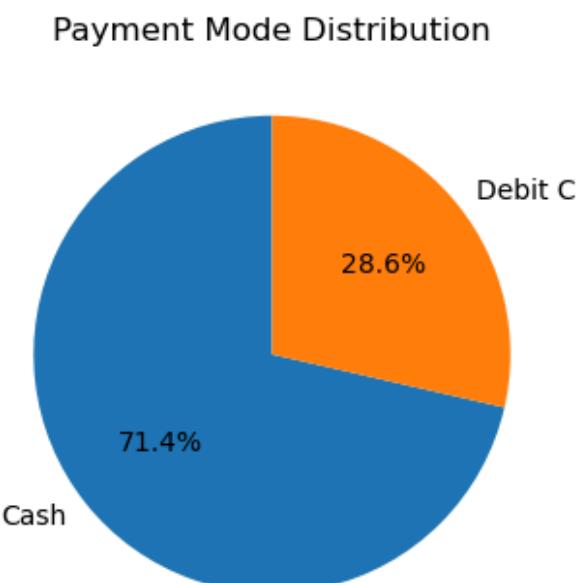


Insightful Dashboards

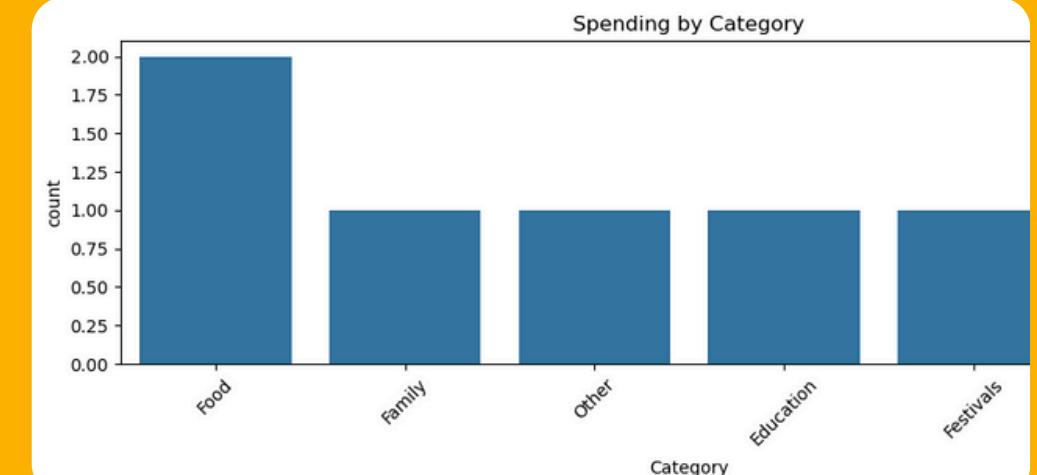
Monthly Trend



Payment Modes



Top Category



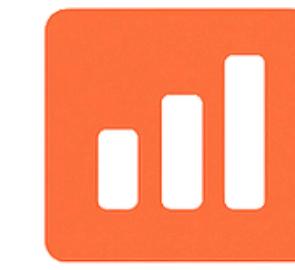
Tech Stack



python



mongoDB



Plotly

ApSc

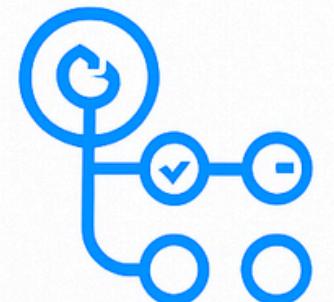
MLflow



Azure



CatBoost



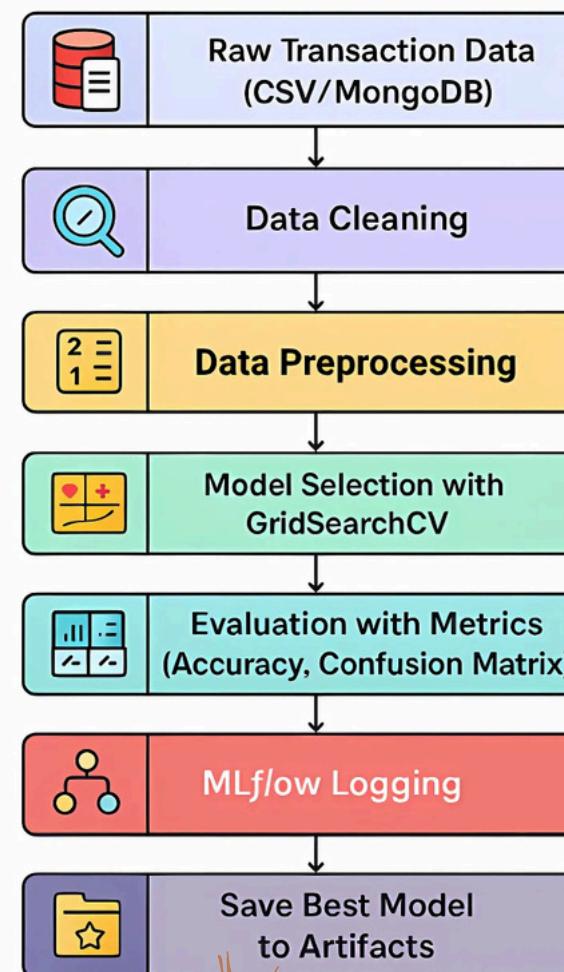
GitHub
Actions

XGBoost

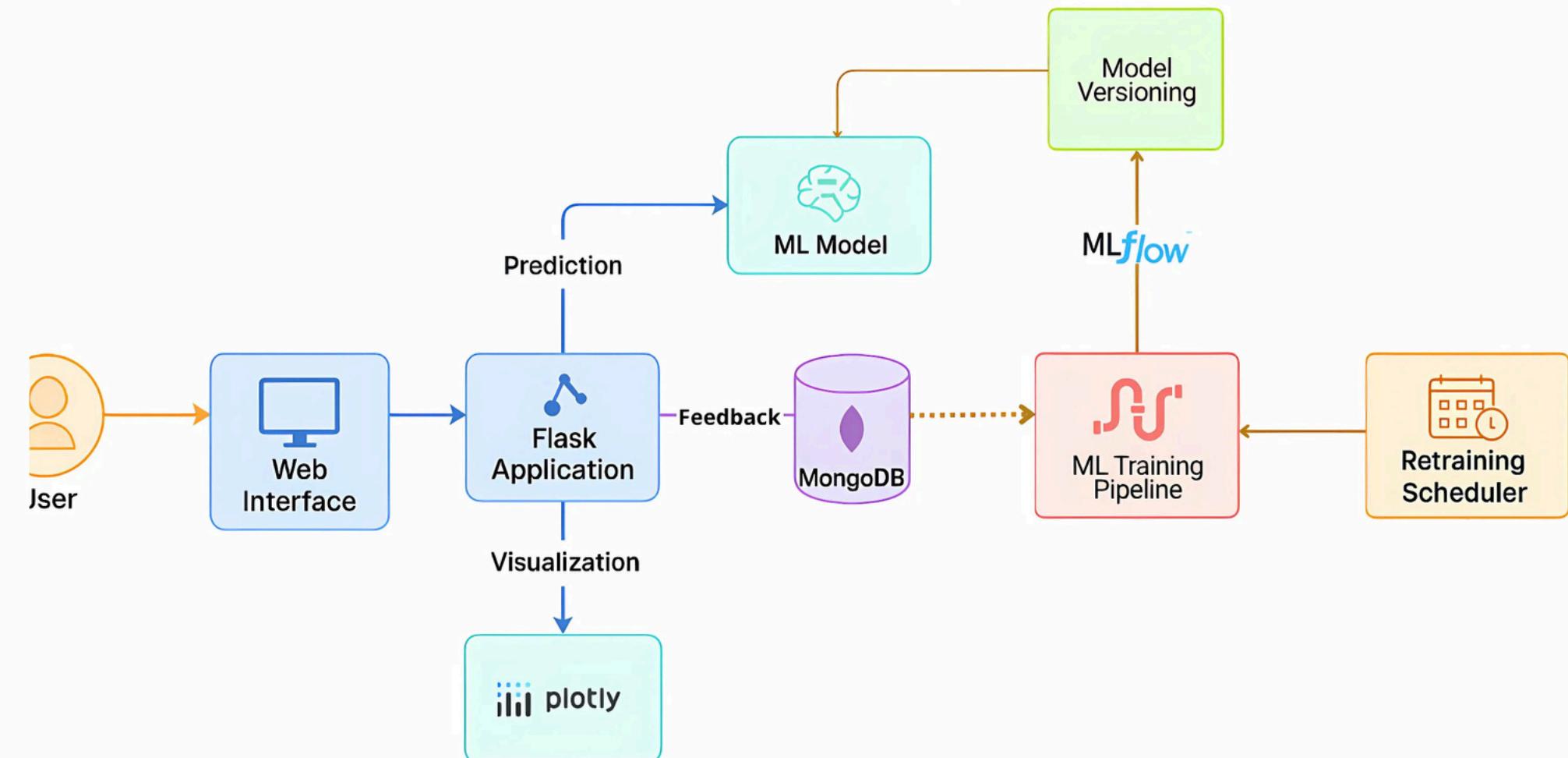
Render

Architecture

ML Training Pipeline for Expense Categorizer



Expense Tracker Application - ARCHITECTURE



More detailed Information and System Diagrams are available in the Repository.

Frontend

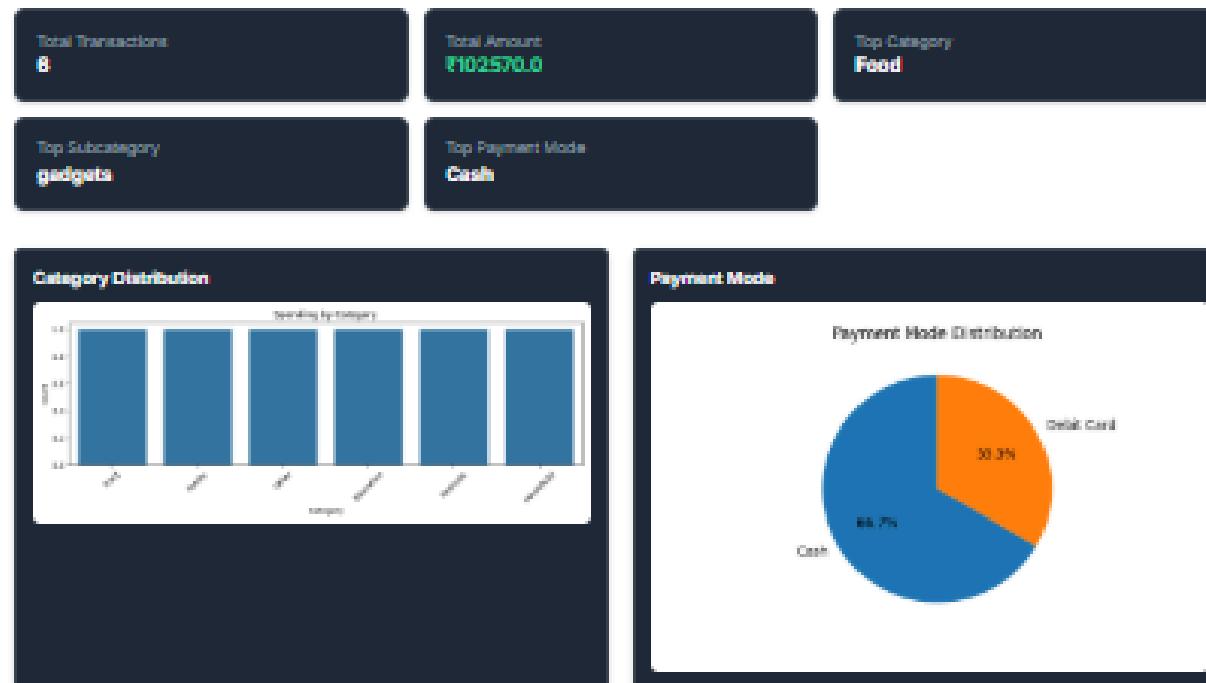
Expense History

Start Date End Date Category Mode

Apply Filters

Date	Amount	Mode	Subcategory	Category
2025-06-28	₹ 80000.00	Debit Card	gadgets	Household
2025-06-28	₹ 2020.00	Cash	Rakshabandhan	Festivals
2025-06-28	₹ 50.00	Cash	Stationary	Education
2025-06-28	₹ 17500.00	Debit Card	gadgets	Other
2025-06-28	₹ 2000.00	Cash	Kirana	Family
2025-06-27	₹ 1000.00	Cash	Sweets	Food

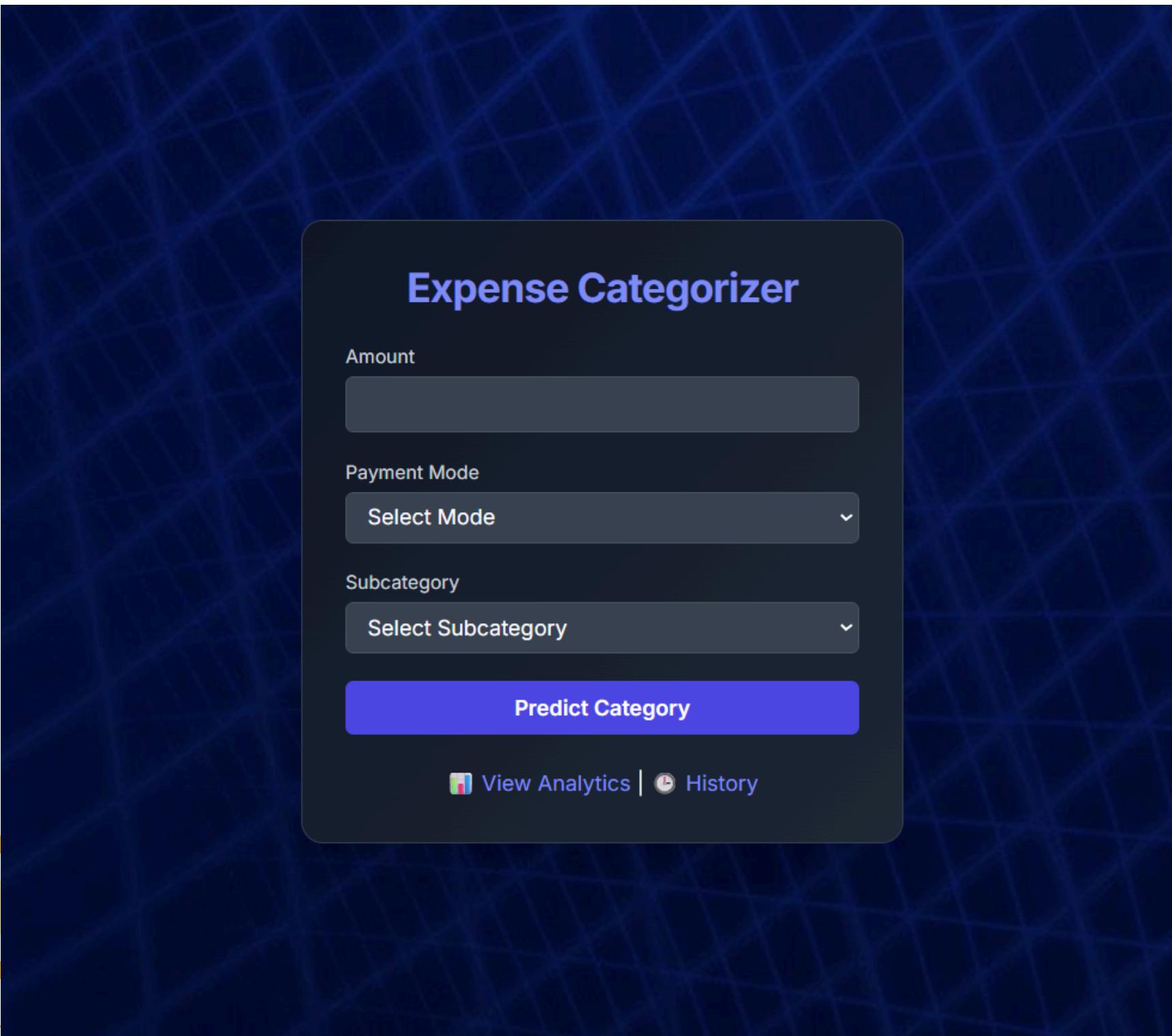
Expense Analytics Dashboard



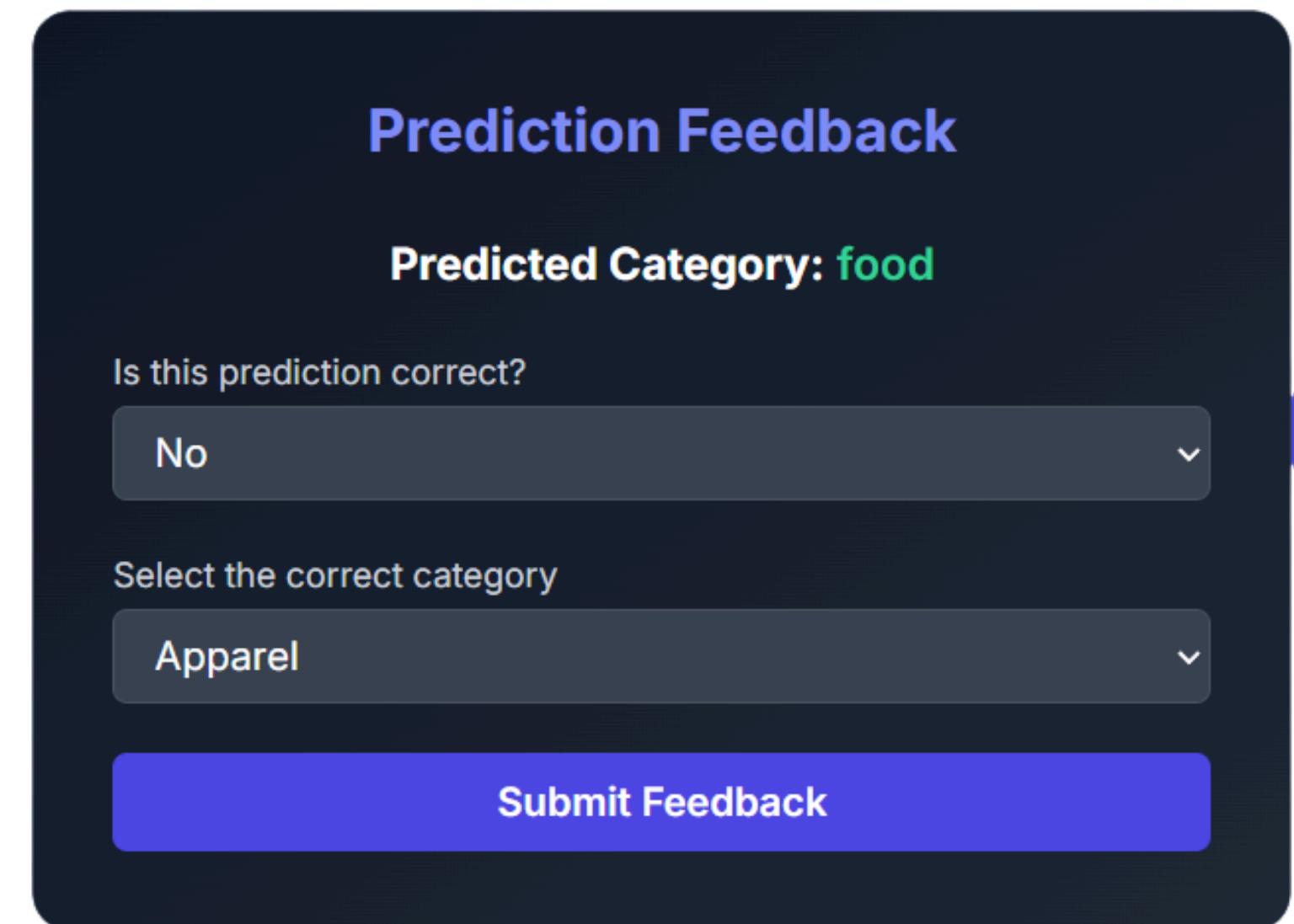
Interactive Charts (Plotly)



Frontend

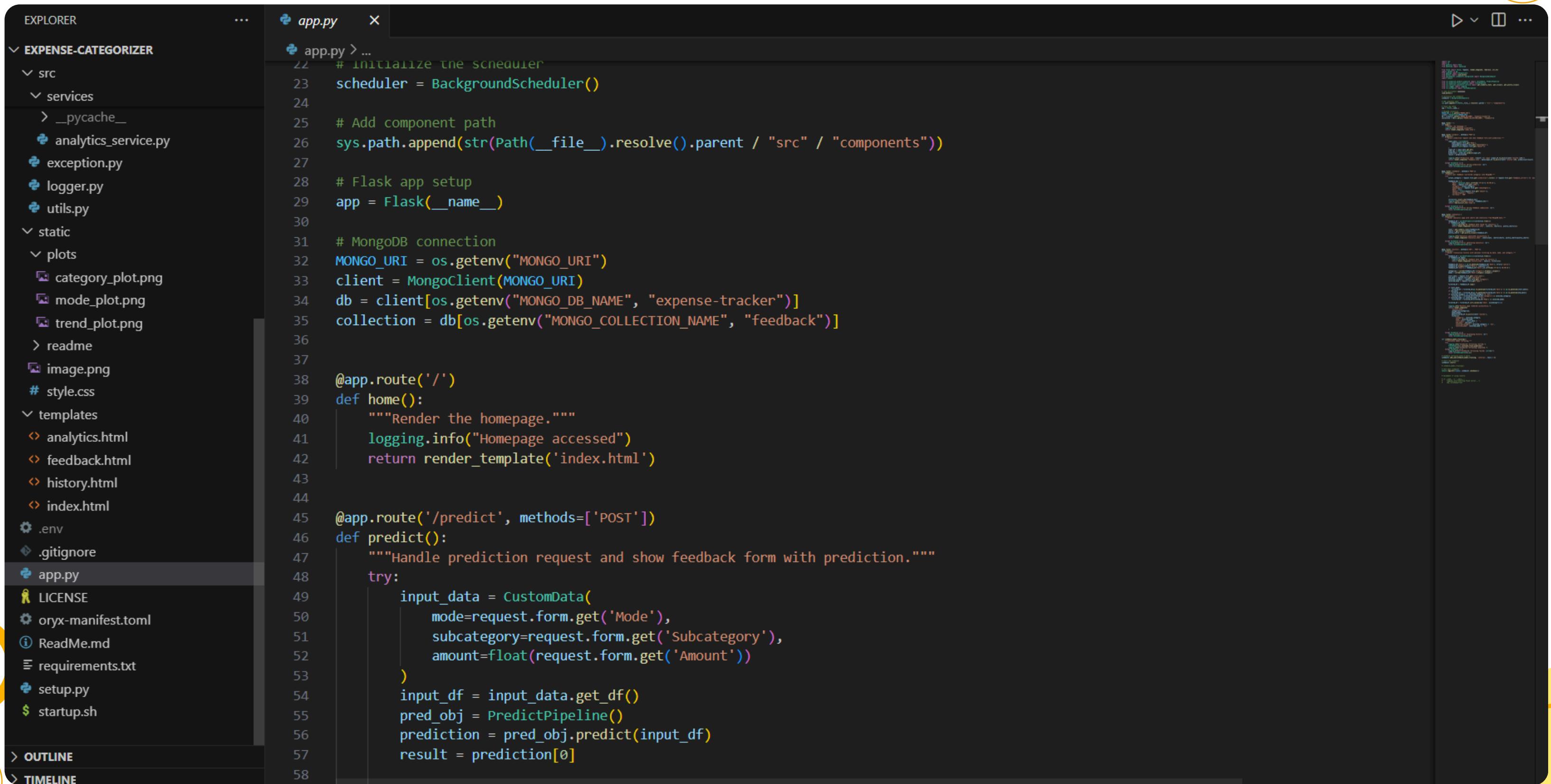


The Expense Categorizer interface features a dark-themed card with a light gray border. At the top is the title "Expense Categorizer". Below it are three input fields: "Amount" (text input), "Payment Mode" (dropdown menu with "Select Mode" placeholder), and "Subcategory" (dropdown menu with "Select Subcategory" placeholder). A large blue button labeled "Predict Category" is centered below these fields. At the bottom of the card are two links: "View Analytics" and "History".



The Prediction Feedback interface shows a dark-themed card with a light gray border. At the top is the title "Prediction Feedback". Below it is the text "Predicted Category: food". A question "Is this prediction correct?" is followed by a dropdown menu with "No" selected. A second question "Select the correct category" is followed by a dropdown menu with "Apparel" selected. A large blue button labeled "Submit Feedback" is at the bottom.

Backend



The image shows a dark-themed code editor window with a sidebar containing project files and a main editor area displaying Python code. The sidebar on the left lists files and folders for a project named 'EXPENSE-CATEGORIZER'. The main editor area shows 'app.py' with the following code:

```
# Initialize the scheduler
scheduler = BackgroundScheduler()

# Add component path
sys.path.append(str(Path(__file__).resolve().parent / "src" / "components"))

# Flask app setup
app = Flask(__name__)

# MongoDB connection
MONGO_URI = os.getenv("MONGO_URI")
client = MongoClient(MONGO_URI)
db = client[os.getenv("MONGO_DB_NAME", "expense-tracker")]
collection = db[os.getenv("MONGO_COLLECTION_NAME", "feedback")]

@app.route('/')
def home():
    """Render the homepage."""
    logging.info("Homepage accessed")
    return render_template('index.html')

@app.route('/predict', methods=['POST'])
def predict():
    """Handle prediction request and show feedback form with prediction."""
    try:
        input_data = CustomData(
            mode=request.form.get('Mode'),
            subcategory=request.form.get('Subcategory'),
            amount=float(request.form.get('Amount'))
        )
        input_df = input_data.get_df()
        pred_obj = PredictPipeline()
        prediction = pred_obj.predict(input_df)
        result = prediction[0]
    except Exception as e:
        logging.error(f"Error during prediction: {e}")
        return render_template('error.html', error=e)

    return render_template('result.html', prediction=result)
```

 Note:

The model may not perform perfectly yet – especially on new or unusual inputs – but that's kind of the point.

With continuous feedback and retraining, it's designed to get better over time. Just like humans do. 

Try It Yourself

Live Demo:

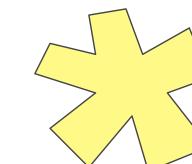
expense-categorizer.onrender.com

GitHub:

github.com/ksv-py/expense-categorizer



Built with code, caffeine, and curiosity by
Keshav Jangid





The background features several yellow decorative elements: a large vertical oval on the left, a horizontal bar at the top, and various circles of different sizes and positions. Some circles have a double outline, while others are simple. A small zigzag line is located near the top center.

Thank You