Project 1: Personal Expense Forecasting System

Project Title

Personal Expense Forecasting and Budget Optimization

Skills Take Away From This Project

Time Series Analysis, Feature Engineering, Machine Learning Models, LSTM Neural Networks, Transformer-Based Forecasting, Budget Optimization Algorithms, Streamlit Application Development, Data Visualization, Financial Analytics

Domain

Personal Finance Management and Predictive Analytics

Problem Statement

Managing personal finances effectively is a critical challenge faced by individuals worldwide. Traditional budgeting methods are often static and fail to account for seasonal variations, lifestyle changes, and unexpected expenses. Many people struggle to predict future expenses accurately, leading to overspending, insufficient savings, and financial stress.

This project aims to develop a **machine learning and deep learning-based forecasting system** that predicts personal expenses across different categories (housing, food, transportation, entertainment, etc.) based on historical spending patterns, seasonal trends, and external factors.

Using a **comprehensive dataset** that includes transaction history, income patterns, demographic information, and economic indicators, learners will create a robust model to forecast monthly expenses and provide **budget optimization recommendations**.

The project will also include building an **interactive Streamlit application** that allows users to input their financial data, visualize spending patterns, and receive personalized budget recommendations with expense forecasts for the next 3–6 months.

Business Use Cases

- **Personal Finance Management**: Plan monthly budgets effectively and identify overspending before it occurs.

- **Financial Planning Services**: Advisors can use forecasts to provide personalized recommendations.
- **Banking Applications**: Banks can integrate forecasts into mobile apps for customer insights.
- **Expense Management Apps**: Enhance existing fintech apps with predictive capabilities.
- **Insurance & Loan Services**: Assess spending patterns for creditworthiness and risk evaluation.

Approach

Data Collection

- **User-Collected Transaction Data**: Bank statements, credit card bills, UPI/Wallet exports (Paytm, Google Pay, PhonePe).
- **Kaggle Datasets**: *personal-expense-transaction-data* and other financial datasets.
- **Hybrid Dataset Creation**: Merge Kaggle datasets with real user-collected data for diversity.
- **Data Cleaning & Formatting**: Standardize into a unified schema with date, category, merchant, and amount.

Data Preprocessing

- Categorize expenses using **NLP** on merchant descriptions.
- Handle **missing values, duplicates, and outliers**.
- Create **time-based features** (day of week, month, season, holidays).

Exploratory Data Analysis (EDA)

- Analyze category-wise and time-based spending trends.
- Detect **seasonality and cyclical behavior** in expenses.
- Correlate **income and demographics** with spending behavior.

Feature Engineering

- Create **lag features**, rolling averages, and moving windows.
- Generate **category-specific ratios and volatility measures**.

- Add external indicators (inflation, interest rates, fuel prices).

Modeling

- **Baseline Models**:
  - Linear Regression, ARIMA, SARIMA, Facebook Prophet.
- **Machine Learning Models**:
  - Random Forest Regressor, XGBoost, LightGBM.
- **Deep Learning Models**:
  - LSTM, GRU, Bi-LSTM for sequential dependency learning.
  - 1D CNNs for detecting local spending patterns.
- **Transformer-Based Models (Advanced)**:
  - Temporal Fusion Transformer (TFT), N-BEATS, Autoformer.
- **Ensemble Methods**: Combine ML + DL for robust performance.

Evaluation

- Forecasting metrics: **MAE, RMSE, MAPE, Directional Accuracy**.
- Compare ML vs DL vs Transformers.
- Category-wise validation (housing, food, etc.).
- Short vs long-term horizon (1, 3, 6 months).

Deployment

- **Streamlit App** with dashboards, visualizations, and forecast reports.
- Budget optimization module suggesting category limits.
- Export results to **Excel/PDF** for user reports.

Results

By the end of the project, learners should achieve:

- **Processed Financial Dataset** combining Kaggle + personal transactions.
- **Comprehensive EDA** with professional visualizations.

- **Advanced Forecasting Models**: LSTM/GRU and Transformers achieving **<15% MAPE**.
- **Interactive Application**: Streamlit app for forecasting, budgeting, and financial scoring.
- **Performance Benchmarks**: Comparison of ML, DL, and Transformer-based approaches.

Project Evaluation Metrics

- **Forecasting Accuracy**: MAE, RMSE, MAPE, Directional Accuracy.
- **Category Performance**: Accuracy for each spending category.
- **Time Horizon Analysis**: Forecast accuracy for 1, 3, and 6 months.
- **Application Usability**: UI design, responsiveness, ease of use.
- **Business Value**: Actionable recommendations and real-world relevance.

Technical Tags

Time Series Forecasting, Machine Learning, LSTM, GRU, Neural Networks, Transformers, N-BEATS, Prophet, Personal Finance, Budget Optimization, Streamlit, Python, TensorFlow/PyTorch, Pandas, Plotly

---

Dataset

Primary Dataset

- **User-Collected Data**: Personal bank transactions, credit card expenses, UPI/Wallet payments.
- **Kaggle Datasets**:
  - *personal-expense-transaction-data*
  - Other financial datasets for model validation.

Additional Sources

- **Economic Indicators**: Inflation, unemployment, interest rates.
- **Synthetic Data**: To balance underrepresented categories.

Project Deliverables

- **Source Code**: Preprocessing, modeling, visualization, and app scripts.
- **Model Files**: Trained ML/DL/Transformer models and configs.
- **Data**: Cleaned datasets (Kaggle + user-collected).
- **Documentation**: Technical report, user manual, and API docs if applicable.
- **Performance Reports**: Forecasting accuracy benchmarks.
- **Streamlit App**: Interactive app with financial dashboard.

Project Guidelines

Coding Standards

- Follow **PEP 8**, modular design, and proper error handling.
- Add docstrings and logging for clarity.

Version Control

- Use **Git with branches** for preprocessing, modeling, and deployment.
- Maintain clear commit history.

Testing

- Validate preprocessing pipelines with various inputs.
- Unit test forecasting functions.
- End-to-end testing of the Streamlit app.

Documentation

- **README.md** with setup & usage.
- Technical report with methodology and findings.
- Screenshots/user manual for non-technical users.

Deployment

- Deploy with **Streamlit Cloud / Heroku / AWS**.
- Provide **Dockerfiles** for reproducibility.

- Enable logging & monitoring for production.

Timeline (8 Weeks)

- **Week 1-2**: Data collection (user + Kaggle), preprocessing, EDA.
- **Week 3**: Train baseline models (Linear Regression, ARIMA, Prophet).
- **Week 4**: Train ML models (Random Forest, XGBoost, LightGBM).
- **Week 5-6**: Train DL models (LSTM, GRU, Bi-LSTM, CNNs).
- **Week 7**: Experiment with Transformers (TFT, N-BEATS). Compare performance.
- **Week 8**: App development, deployment, documentation, presentation.