**Spending Insight Pro**

Monthly expenditure analysis using Data science

This project involves using the numpy, pandas,seaborn and matplotlib libraries in order to build a model that analyzes the category , amount spent , balance amount and the date on which the purchase was made.The data set was collected from my bank statement of each month and then it was uploaded as an Excel workbook. The report was visualized using PowerBI

Steps to implement the model:

1. Collect data on expenditures, such as amount spent, category, and date.
2. Load the data into a Python environment using the Pandas library.
3. Perform exploratory data analysis to understand the data and its distribution.
4. Visualize the data to identify patterns and relationships.
5. Perform feature engineering to extract relevant information from the data.
6. Pre-process the data by dropping irrelevant columns and one-hot encoding categorical variables.
7. Split the data into training and testing sets for model building.
8. Build and train a regression model using the Random Forest algorithm.
9. Evaluate the model performance using a metric such as Root Mean Squared Error.
10. Use the trained model to predict expenditure amounts for new data points, such as for a specific category and date.

The code used to build the model has been attached below:

# Import libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

# Load the dataset

df = pd.read\_csv('expenditure\_data.csv')

# Exploratory data analysis

print(df.head()) # View the first few rows of data

print(df.info()) # Get a summary of the dataset

print(df.describe()) # Get summary statistics of the dataset

# Data visualization

sns.pairplot(df, hue='Category', diag\_kind='hist')

plt.show()

# Feature engineering

df['Year'] = pd.to\_datetime(df['Date']).dt.year

df['Month'] = pd.to\_datetime(df['Date']).dt.month

# Data pre-processing

df = df.drop(['Date'], axis=1) # Drop the date column

df = pd.get\_dummies(df, columns=['Category']) # One-hot encoding for categorical variables

# Split data into train and test sets

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

X = df.drop(['Amount'], axis=1)

y = df['Amount']

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

# Build and train the model

from sklearn.ensemble import RandomForestRegressor

rf = RandomForestRegressor(n\_estimators=100, random\_state=42)

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

# Evaluate the model

from sklearn.metrics import mean\_squared\_error

y\_pred = rf.predict(X\_test)

mse = mean\_squared\_error(y\_test, y\_pred)

rmse = np.sqrt(mse)

print('Root Mean Squared Error:', rmse)

# Make predictions

new\_data = pd.DataFrame({

'Year': [2022],

'Month': [6],

'Category\_Entertainment': [0],

'Category\_Food': [1],

'Category\_Housing': [0],

'Category\_Transportation': [0],

'Category\_Utilities': [0]

})

predicted\_amount = rf.predict(new\_data)

print('Predicted amount:', predicted\_amount)

The PowerBI report used to represent the visual model :

