For an **Insurance Pricing Forecast** project, I built a Python model that predicts insurance premiums based on historical data and specific factors. Insurance companies use factors like age, income, the type of insurance, and other risk-related metrics to estimate premiums.

Steps:

1. **Gather Data**: Use a dataset with historical insurance premiums and corresponding features such as age, insurance type, etc.
2. **Feature Selection**: Extract useful features like age, income, insurance type, claims history, and location.
3. **Train a Model**: Use a basic machine learning model, such as linear regression or decision trees, to forecast future premiums based on the provided features.
4. **Make Predictions**: Use the trained model to predict future premiums for new users.
5. **Evaluate the Model**: Check how well the model performs using metrics like Mean Absolute Error (MAE) or Mean Squared Error (MSE).

Implementing a model using linear regression with the popular scikit-learn library.

Code:

```
# Importing libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error

# Sample Data (You would replace this with your actual dataset)
# Columns: Age, Annual Income, Claims History, Type of Insurance (Motor = 1, Health = 2, Property = 3, Travel = 4), Premium
data = {
    'Age': [25, 40, 35, 50, 23, 30, 60, 45, 33, 38],
    'Annual_Income': [50000, 80000, 60000, 120000, 40000, 55000, 90000, 85000, 75000, 95000],
```

```
    'Claims_History': [1, 0, 0, 1, 1, 0, 1, 0, 1, 0],
    'Insurance_Type': [1, 2, 3, 1, 4, 2, 3, 1, 4, 3],
    'Premium': [1200, 2500, 1800, 3000, 1000, 2200, 2800, 2900, 1300, 2700]
}

# Convert to DataFrame
df = pd.DataFrame(data)

# Define Features and Target
X = df[['Age', 'Annual_Income', 'Claims_History', 'Insurance_Type']]  # Features
y = df['Premium']  # Target

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Linear Regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)

# Output the results
```

```
print(f'Mean Absolute Error (MAE): {mae}')
print(f'Mean Squared Error (MSE): {mse}')

# Forecast for a new user (Example)
new_user = np.array([[32, 65000, 0, 2]])  # Age: 32, Income: 65000, Claims: 0, Health Insurance
predicted_premium = model.predict(new_user)
print(f'Predicted Premium for new user: ${predicted_premium[0]:.2f}')
```

Explanation:

1. **Data Preparation**: We use a sample dataset with columns like age, income, claims history, and type of insurance.
2. **Model**: A simple linear regression model is used to fit the data.
3. **Prediction**: The model predicts premiums based on the given features.
4. **Evaluation**: We calculate error metrics like MAE and MSE to evaluate model performance.

Output:

```
Mean Absolute Error (MAE): 246.89764276642757
Mean Squared Error (MSE): 85456.83679718585
Predicted Premium for new user: $2147.17
```