In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
df = pd.read_csv("/kaggle/input/insurance-dataset-simple-linear-regression/simplelinearregression.csv")
X = df.iloc[:,:-1].values
y = df.iloc[:,-1].values
df
```

Out[2]:

	Age	Premium
0	18	10000
1	22	15000
2	23	18000
3	26	21000
4	28	24000
5	31	26500
6	33	27000

In [3]:

```
print(X)
```

```
[[18]
 [22]
 [23]
 [26]
 [28]
 [31]
 [33]]
```

In [4]:

```
print(y)
```

```
[10000 15000 18000 21000 24000 26500 27000]
```

In [5]:

```
y = y.reshape(len(y),1)
print(y)
```

```
[[10000]
 [15000]
 [18000]
 [21000]
 [24000]
 [26500]
 [27000]]
```

In [6]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split (X,y, test_size=0.2, random_state =0)
```

In [7]:

```
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.linear_model import LinearRegression
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Create a list of regression models
models = [
    ("Linear Regression", LinearRegression()),
    ("Decision Tree Regression", DecisionTreeRegressor(random_state=42)),
    ("Random Forest Regression", RandomForestRegressor(random_state=42))
]
mae_values = []
mse_values = []
rmse_values = []
# Iterate through the models, train, and evaluate them
for model_name, model in models:
    # Train the model
    model.fit(X_train, y_train)

    # Make predictions
    y_pred = model.predict(X_test)

    # Calculate evaluation metrics
    mae = mean_absolute_error(y_test, y_pred)
    mse = mean_squared_error(y_test, y_pred)
    rmse = np.sqrt(mse)
    r2 = r2_score(y_test, y_pred)

    # Print the results
    print(f"Model: {model_name}")
    print(f"Mean Absolute Error (MAE): {mae:.2f}")
    print(f"Mean Squared Error (MSE): {mse:.2f}")
    print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
    print(f"R-squared (R2) Score: {r2:.2f}")
    print("-" * 40)
```

```
Model: Linear Regression
Mean Absolute Error (MAE): 2086.54
Mean Squared Error (MSE): 4919031.07
Root Mean Squared Error (RMSE): 2217.89
R-squared (R2) Score: 0.76
----------------------------------------
Model: Decision Tree Regression
Mean Absolute Error (MAE): 1750.00
Mean Squared Error (MSE): 4625000.00
Root Mean Squared Error (RMSE): 2150.58
R-squared (R2) Score: 0.77
----------------------------------------
Model: Random Forest Regression
Mean Absolute Error (MAE): 1870.00
Mean Squared Error (MSE): 3497300.00
Root Mean Squared Error (RMSE): 1870.11
R-squared (R2) Score: 0.83
----------------------------------------

/tmp/ipykernel_32/276533376.py:19: DataConversionWarning: A column-vec
tor y was passed when a 1d array was expected. Please change the shape
of y to (n_samples,), for example using ravel().
  model.fit(X_train, y_train)
```

In [ ]: