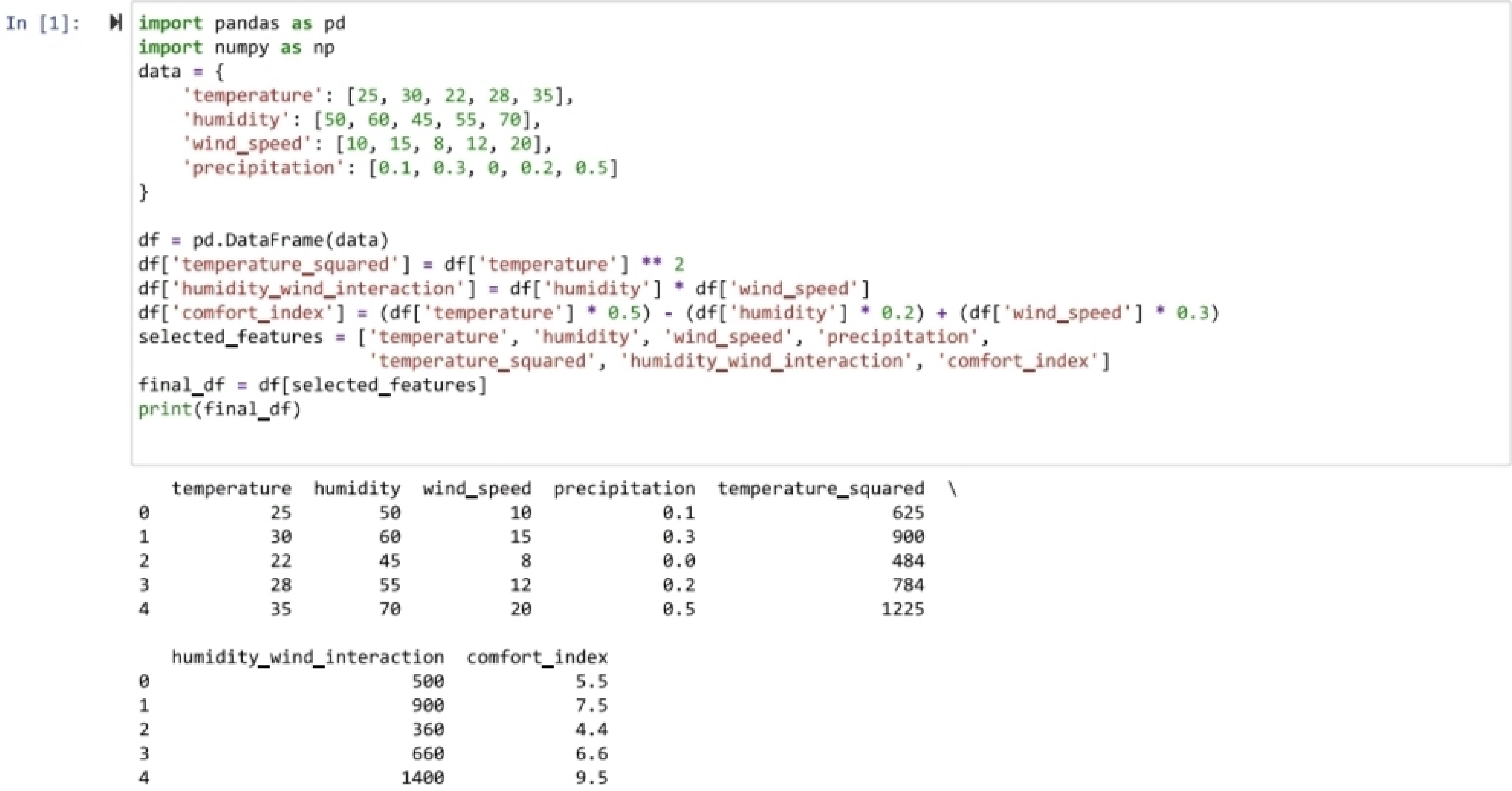
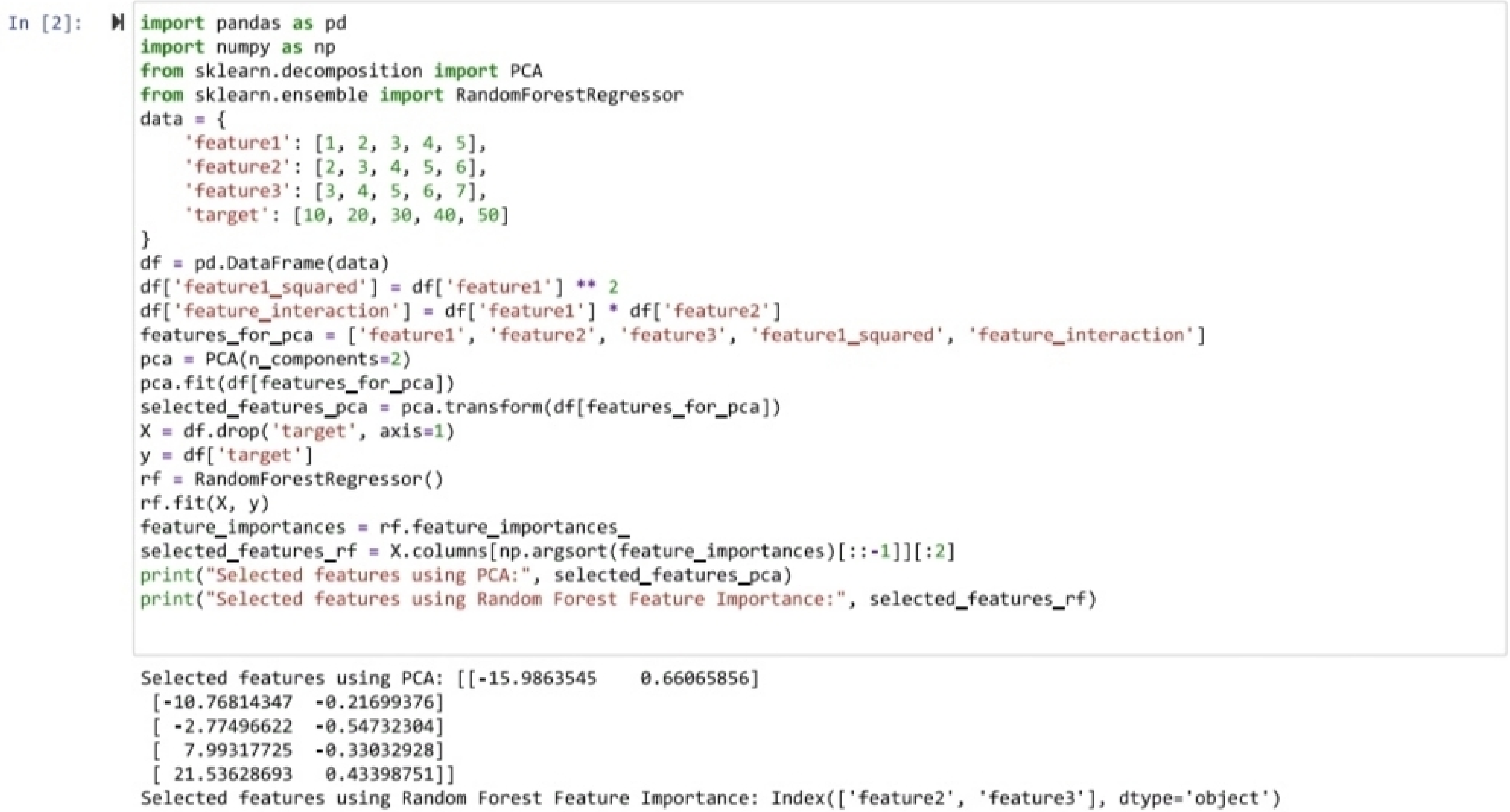


```
import pandas as pd
import numpy as np
data = {
    'temperature': [25, 30, 22, 28, 35],
    'humidity': [50, 60, 45, 55, 70],
    'wind_speed': [10, 15, 8, 12, 20],
    'precipitation': [0.1, 0.3, 0, 0.2, 0.5]
}

df = pd.DataFrame(data)
df['temperature_squared'] = df['temperature'] ** 2
df['humidity_wind_interaction'] = df['humidity'] * df['wind_speed']
df['comfort_index'] = (df['temperature'] * 0.5) - (df['humidity'] * 0.2) + (df['wind_speed'] * 0.3)
selected_features = ['temperature', 'humidity', 'wind_speed', 'precipitation',
                     'temperature_squared', 'humidity_wind_interaction', 'comfort_index']
final_df = df[selected_features]
print(final_df)
```

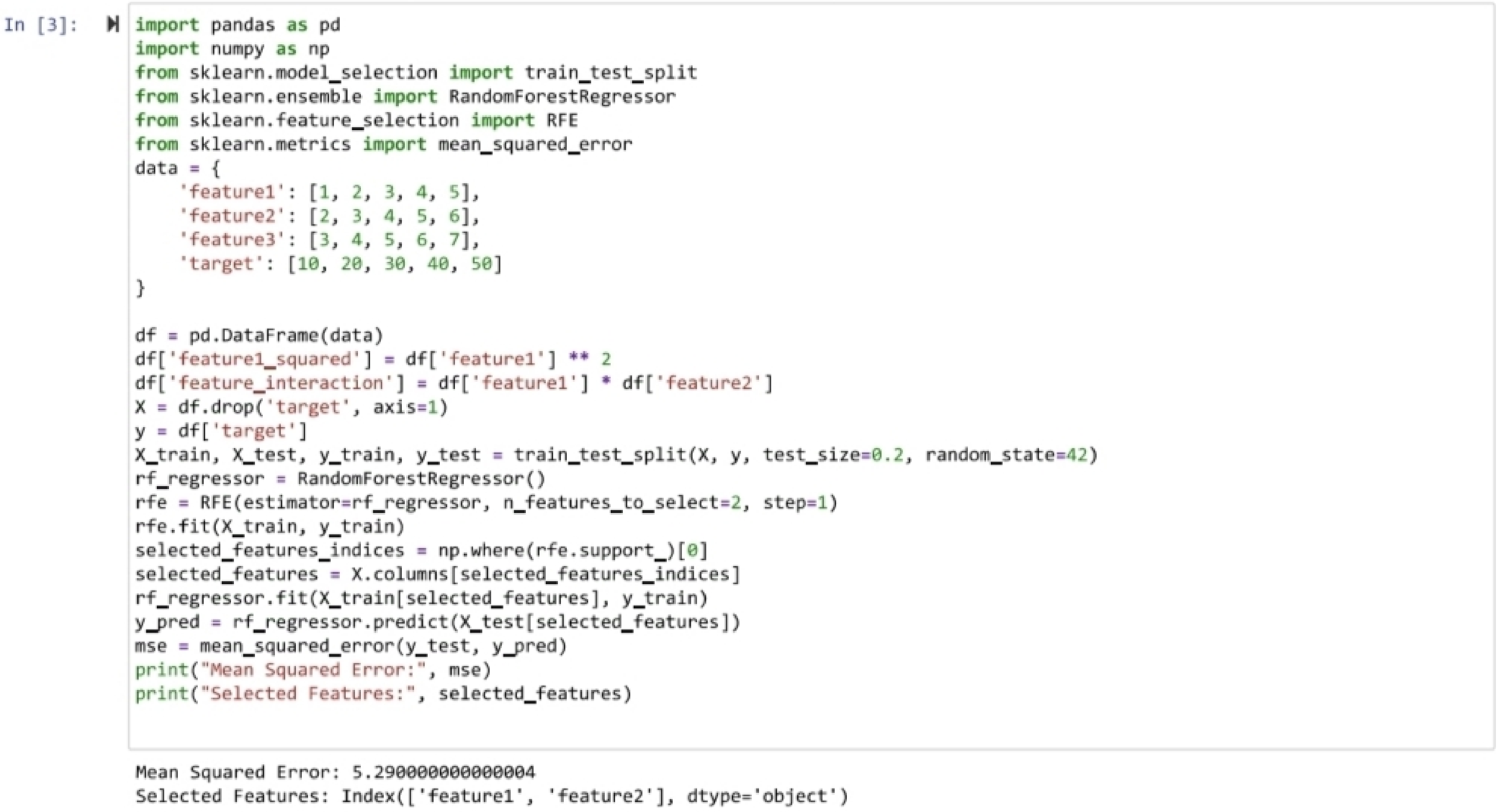
```
   temperature  humidity  wind_speed  precipitation  temperature_squared  \
0           25        50          10            0.1                  625
1           30        60          15            0.3                  900
2           22        45           8            0.0                  484
3           28        55          12            0.2                  784
4           35        70          20            0.5                 1225

   humidity_wind_interaction  comfort_index
0                        500            5.5
1                        900            7.5
2                        360            4.4
3                        660            6.6
4                       1400            9.5
```



```python
import pandas as pd
import numpy as np
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestRegressor
data = {
    'feature1': [1, 2, 3, 4, 5],
    'feature2': [2, 3, 4, 5, 6],
    'feature3': [3, 4, 5, 6, 7],
    'target': [10, 20, 30, 40, 50]
}
df = pd.DataFrame(data)
df['feature1_squared'] = df['feature1'] ** 2
df['feature_interaction'] = df['feature1'] * df['feature2']
features_for_pca = ['feature1', 'feature2', 'feature3', 'feature1_squared', 'feature_interaction']
pca = PCA(n_components=2)
pca.fit(df[features_for_pca])
selected_features_pca = pca.transform(df[features_for_pca])
X = df.drop('target', axis=1)
y = df['target']
rf = RandomForestRegressor()
rf.fit(X, y)
feature_importances = rf.feature_importances_
selected_features_rf = X.columns[np.argsort(feature_importances)[::-1]][:2]
print("Selected features using PCA:", selected_features_pca)
print("Selected features using Random Forest Feature Importance:", selected_features_rf)
```

```
Selected features using PCA: [[-15.9863545    0.66065856]
 [-10.76814347  -0.21699376]
 [ -2.77496622  -0.54732304]
 [  7.99317725  -0.33032928]
 [ 21.53628693   0.43398751]]
Selected features using Random Forest Feature Importance: Index(['feature2', 'feature3'], dtype='object')
```



```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.feature_selection import RFE
from sklearn.metrics import mean_squared_error
data = {
    'feature1': [1, 2, 3, 4, 5],
    'feature2': [2, 3, 4, 5, 6],
    'feature3': [3, 4, 5, 6, 7],
    'target': [10, 20, 30, 40, 50]
}

df = pd.DataFrame(data)
df['feature1_squared'] = df['feature1'] ** 2
df['feature_interaction'] = df['feature1'] * df['feature2']
X = df.drop('target', axis=1)
y = df['target']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
rf_regressor = RandomForestRegressor()
rfe = RFE(estimator=rf_regressor, n_features_to_select=2, step=1)
rfe.fit(X_train, y_train)
selected_features_indices = np.where(rfe.support_)[0]
selected_features = X.columns[selected_features_indices]
rf_regressor.fit(X_train[selected_features], y_train)
y_pred = rf_regressor.predict(X_test[selected_features])
mse = mean_squared_error(y_test, y_pred)
print("Mean Squared Error:", mse)
print("Selected Features:", selected_features)
```

```
Mean Squared Error: 5.290000000000004
Selected Features: Index(['feature1', 'feature2'], dtype='object')
```