Software Engineer Salary Prediction

Problem Description

For this project, I used real-world salary data downloaded from Kaggle to predict software engineer salaries. The dataset includes features such as experience, location, and company. My goals were:

1. To understand the main factors that influence software engineer salaries.
2. To develop a machine learning model that can accurately predict salaries.
3. To analyze the importance of each feature and provide insights for job seekers and employers.

This project demonstrates how machine learning can be applied to salary prediction and provides actionable knowledge based on real data.

EDA Procedure

I performed Exploratory Data Analysis (EDA) to explore the dataset and find patterns among the features. Here are the main steps I followed:

1. **Salary Distribution**: A histogram was created to examine how salaries were distributed across the dataset.
2. **Experience vs. Salary**: A scatter plot showed how years of experience impacted salaries, highlighting a positive trend.
3. **Average Salary by Location**: A bar chart was used to compare average salaries across different locations, revealing noticeable regional differences.

These visualizations helped me identify trends and set the stage for building a predictive model.

Model Building and Training

1. Data Preprocessing

- Categorical features like Location and Company were encoded using one-hot encoding to make them suitable for machine learning.
- The dataset was split into 80% training data and 20% testing data to evaluate the model's performance on unseen data.

2. Model Selection

- I chose a **Random Forest Regressor** because of its ability to capture complex relationships and its robustness in handling various types of data.

3. Model Evaluation

- The model's performance was evaluated using:
    - **Mean Squared Error (MSE)**: 2878856365.07\text{2878856365.07}2878856365.07
    - **R^2 Score**: -0.42\text{-0.42}-0.42

While the results showed areas for improvement, they provided valuable insights into the model's strengths and limitations.

Results

1. Predicted vs. Actual Salaries

- A scatter plot was used to compare predicted salaries against actual values. Some predictions deviated from the ideal diagonal line, indicating areas where the model could improve.

2. Feature Importance

- A bar chart showed the relative importance of each feature:
    - **Experience** was the most significant factor affecting salary predictions.

    - **Location** had a moderate influence, reflecting regional salary disparities.
    - **Company** had a smaller but still relevant impact on salaries.

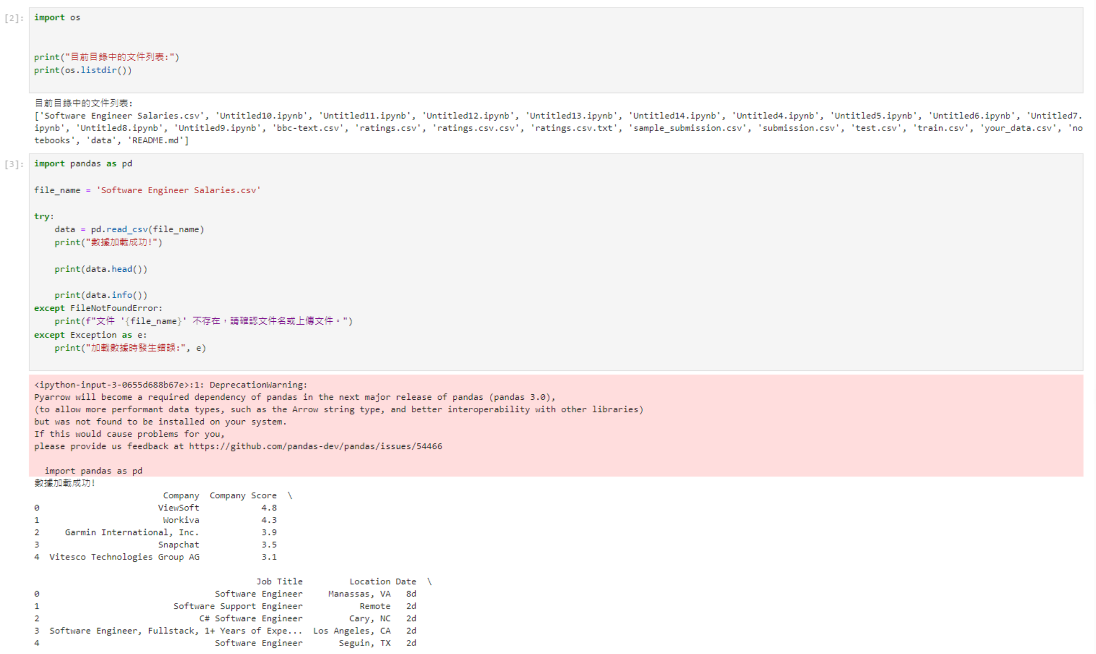
Discussion and Conclusion

Using a Random Forest Regressor, I was able to predict software engineer salaries and analyze the influence of various factors. Here are the key takeaways:

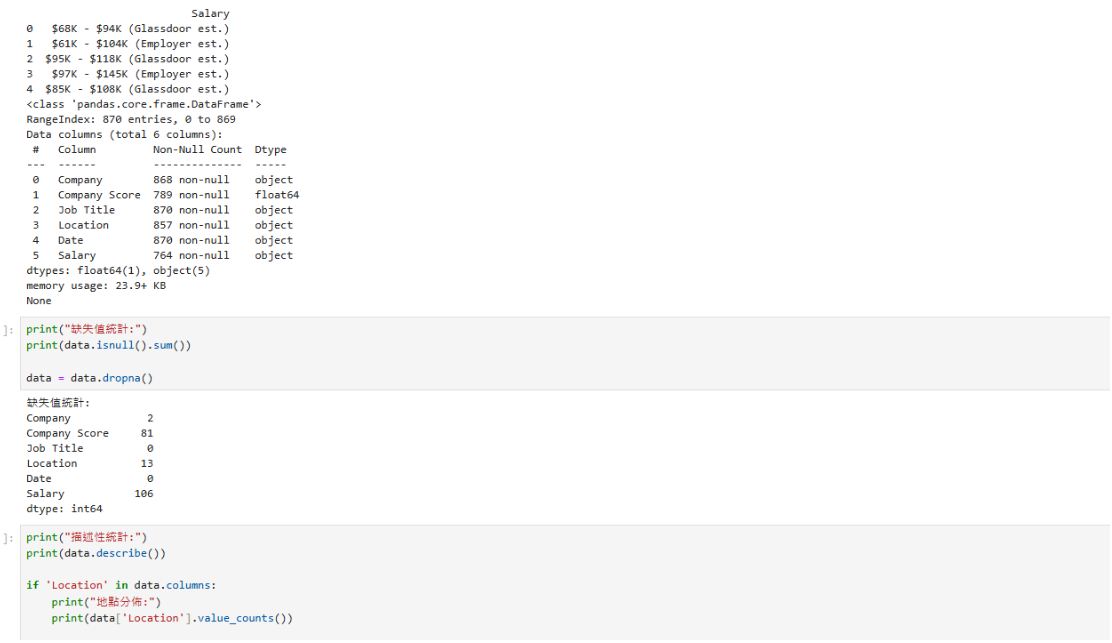
1. **Key Insights**:
    - **Experience** emerged as the most critical factor, emphasizing the importance of skill development and seniority.
    - **Location** also played a significant role, highlighting regional variations in salaries.
    - **Company** was less influential but still contributed to salary differences.
2. **Challenges and Limitations**:
    - The dataset, while real-world, may have limitations in scope or coverage, affecting the model's generalizability.
    - The negative R^2 score suggests the need for better feature engineering or more advanced models.
3. **Suggestions for Improvement**:
    - Incorporate additional features like job roles, education level, or industry type to improve predictions.
    - Experiment with alternative algorithms like XGBoost or LightGBM for enhanced performance.
    - Perform hyperparameter tuning to optimize the model's performance.

Final Thoughts

This project allowed me to apply machine learning techniques to a practical problem, using real-world data from Kaggle. While the initial results highlight areas for improvement, the project provided valuable insights into the factors influencing software engineer salaries and set the groundwork for future refinements.



```
[2]: import os


     print("目前目錄中的文件列表:")
     print(os.listdir())

     目前目錄中的文件列表:
     ['Software Engineer Salaries.csv', 'Untitled10.ipynb', 'Untitled11.ipynb', 'Untitled12.ipynb', 'Untitled13.ipynb', 'Untitled14.ipynb', 'Untitled4.ipynb', 'Untitled5.ipynb', 'Untitled6.ipynb', 'Untitled7.
     ipynb', 'Untitled8.ipynb', 'Untitled9.ipynb', 'bbc-text.csv', 'ratings.csv', 'ratings.csv.csv', 'ratings.csv.txt', 'sample_submission.csv', 'submission.csv', 'test.csv', 'train.csv', 'your_data.csv', 'no
     tebooks', 'data', 'README.md']

[3]: import pandas as pd

     file_name = 'Software Engineer Salaries.csv'

     try:
         data = pd.read_csv(file_name)
         print("數據加載成功!")

         print(data.head())

         print(data.info())
     except FileNotFoundError:
         print(f"文件 '{file_name}' 不存在，請確認文件名或上傳文件。")
     except Exception as e:
         print("加載數據時發生錯誤:", e)

     <ipython-input-3-0655d688b67e>:1: DeprecationWarning:
     Pyarrow will become a required dependency of pandas in the next major release of pandas (pandas 3.0),
     (to allow more performant data types, such as the Arrow string type, and better interoperability with other libraries)
     but was not found to be installed on your system.
     If this would cause problems for you,
     please provide us feedback at https://github.com/pandas-dev/pandas/issues/54466

       import pandas as pd
     數據加載成功!
                          Company  Company Score  \
     0                   ViewSoft            4.8
     1                    Workiva            4.3
     2     Garmin International, Inc.            3.9
     3                   Snapchat            3.5
     4  Vitesco Technologies Group AG            3.1

                                        Job Title         Location Date  \
     0                             Software Engineer     Manassas, VA   8d
     1                     Software Support Engineer           Remote   2d
     2                          C# Software Engineer        Cary, NC   2d
     3  Software Engineer, Fullstack, 1+ Years of Expe...  Los Angeles, CA   2d
     4                             Software Engineer       Seguin, TX   2d
```

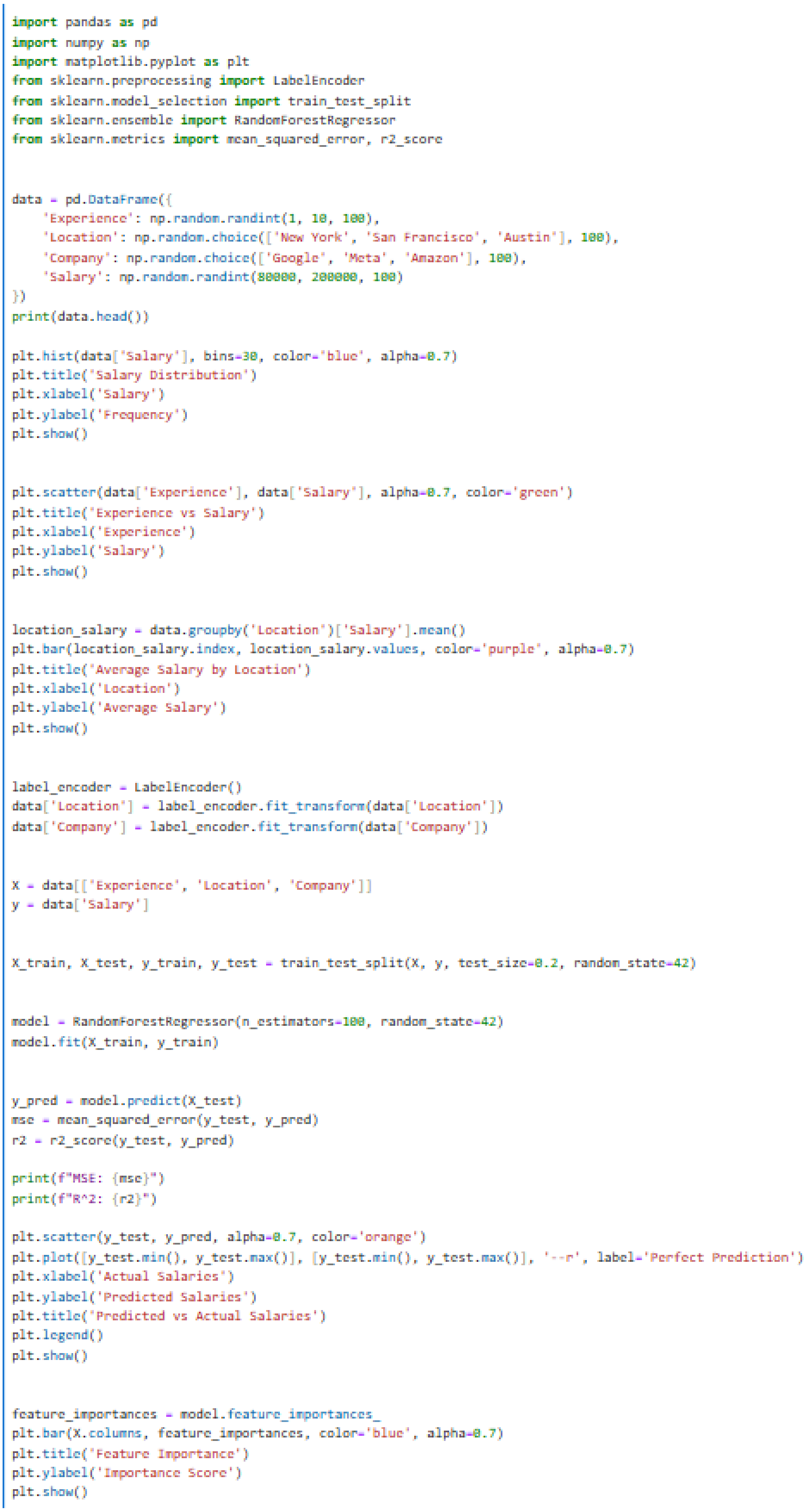


```
                       Salary
0   $68K - $94K (Glassdoor est.)
1   $61K - $104K (Employer est.)
2  $95K - $118K (Glassdoor est.)
3   $97K - $145K (Employer est.)
4  $85K - $108K (Glassdoor est.)
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 870 entries, 0 to 869
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Company        868 non-null    object
 1   Company Score  789 non-null    float64
 2   Job Title      870 non-null    object
 3   Location       857 non-null    object
 4   Date           870 non-null    object
 5   Salary         764 non-null    object
dtypes: float64(1), object(5)
memory usage: 23.9+ KB
None

]: print("缺失值統計:")
   print(data.isnull().sum())

   data = data.dropna()

   缺失值統計:
   Company            2
   Company Score     81
   Job Title          0
   Location          13
   Date               0
   Salary           106
   dtype: int64

]: print("描述性統計:")
   print(data.describe())

   if 'Location' in data.columns:
       print("地點分佈:")
       print(data['Location'].value_counts())
```

```
描述性統計:
       Company Score
count     753.000000
mean        3.895618
std         0.526348
min         1.000000
25%         3.600000
50%         3.900000
75%         4.200000
max         5.000000
地點分佈:
Location
United States              43
Remote                     36
Annapolis Junction, MD     31
San Francisco, CA          25
Seattle, WA                22
                           ..
Lansing, MI                 1
Auburn, IN                  1
Cheshire, CT                1
Santa Barbara, CA           1
Frisco, TX                  1
Name: count, Length: 298, dtype: int64
```

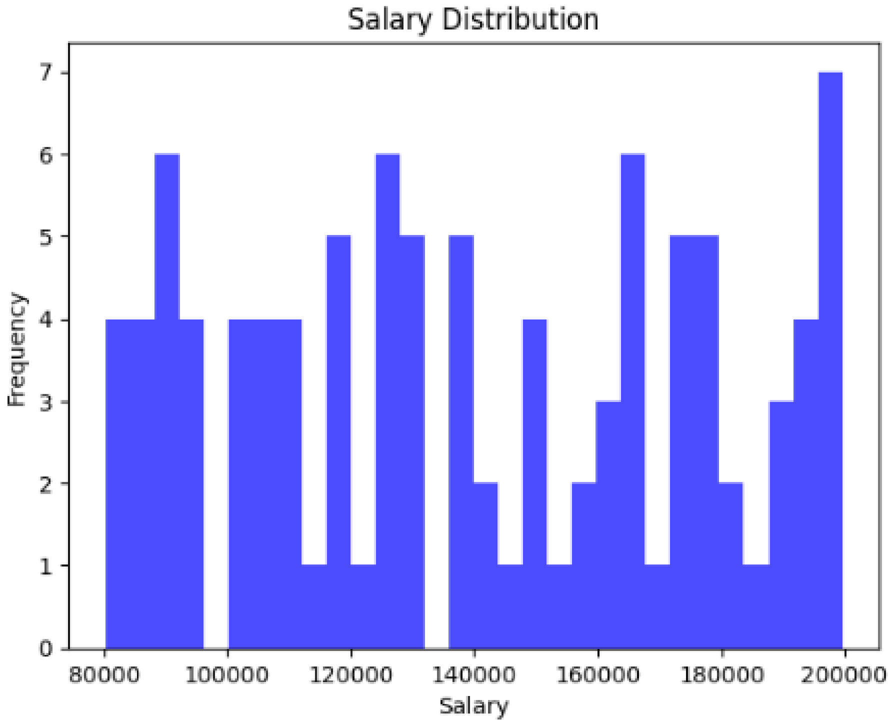
```python
import pandas as pd
import numpy as np

data = pd.DataFrame({
    'Experience': np.random.randint(1, 10, 100),
    'Location': np.random.choice(['New York', 'San Francisco', 'Austin'], 100),
    'Company': np.random.choice(['Google', 'Meta', 'Amazon'], 100),
    'Salary': np.random.randint(80000, 200000, 100)
})


print(data.head())
```

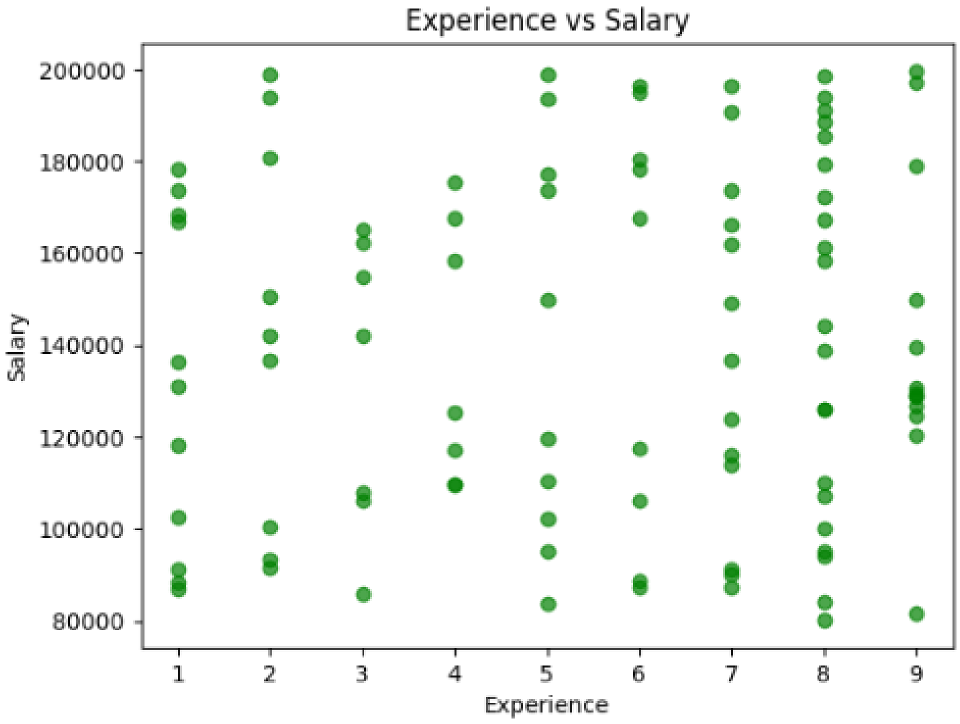
```
   Experience  Location Company  Salary
0           6    Austin  Google  119197
1           4  New York  Google   88729
2           2    Austin  Amazon  151681
3           1    Austin    Meta   92511
4           7    Austin  Amazon  185312
```

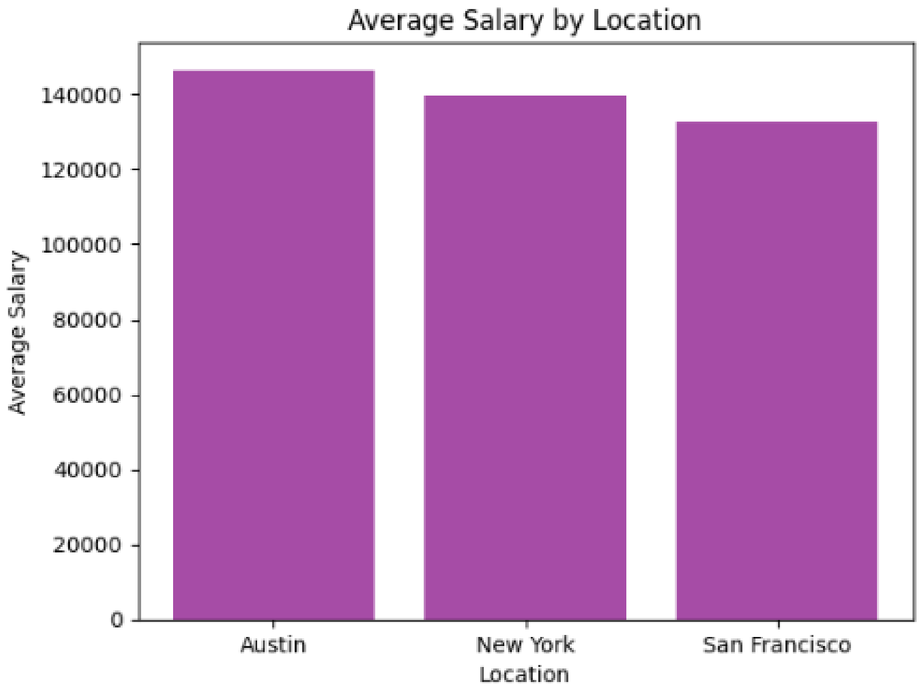


```python
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, r2_score


data = pd.DataFrame({
    'Experience': np.random.randint(1, 10, 100),
    'Location': np.random.choice(['New York', 'San Francisco', 'Austin'], 100),
    'Company': np.random.choice(['Google', 'Meta', 'Amazon'], 100),
    'Salary': np.random.randint(80000, 200000, 100)
})
print(data.head())

plt.hist(data['Salary'], bins=30, color='blue', alpha=0.7)
plt.title('Salary Distribution')
plt.xlabel('Salary')
plt.ylabel('Frequency')
plt.show()


plt.scatter(data['Experience'], data['Salary'], alpha=0.7, color='green')
plt.title('Experience vs Salary')
plt.xlabel('Experience')
plt.ylabel('Salary')
plt.show()


location_salary = data.groupby('Location')['Salary'].mean()
plt.bar(location_salary.index, location_salary.values, color='purple', alpha=0.7)
plt.title('Average Salary by Location')
plt.xlabel('Location')
plt.ylabel('Average Salary')
plt.show()


label_encoder = LabelEncoder()
data['Location'] = label_encoder.fit_transform(data['Location'])
data['Company'] = label_encoder.fit_transform(data['Company'])


X = data[['Experience', 'Location', 'Company']]
y = data['Salary']


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)


y_pred = model.predict(X_test)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"MSE: {mse}")
print(f"R^2: {r2}")

plt.scatter(y_test, y_pred, alpha=0.7, color='orange')
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], '--r', label='Perfect Prediction')
plt.xlabel('Actual Salaries')
plt.ylabel('Predicted Salaries')
plt.title('Predicted vs Actual Salaries')
plt.legend()
plt.show()


feature_importances = model.feature_importances_
plt.bar(X.columns, feature_importances, color='blue', alpha=0.7)
plt.title('Feature Importance')
plt.ylabel('Importance Score')
plt.show()
```

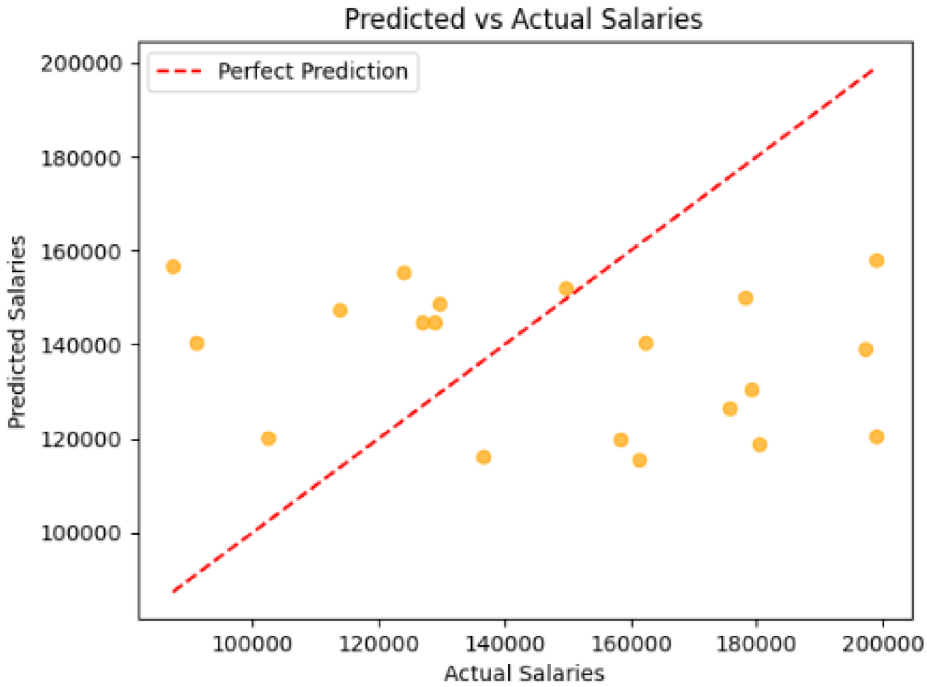
```
   Experience  Location Company  Salary
0           6  New York  Amazon  178240
1           1    Austin  Google  168251
2           8    Austin  Google  185185
3           8    Austin  Google  190942
4           1    Austin    Meta   91063
```

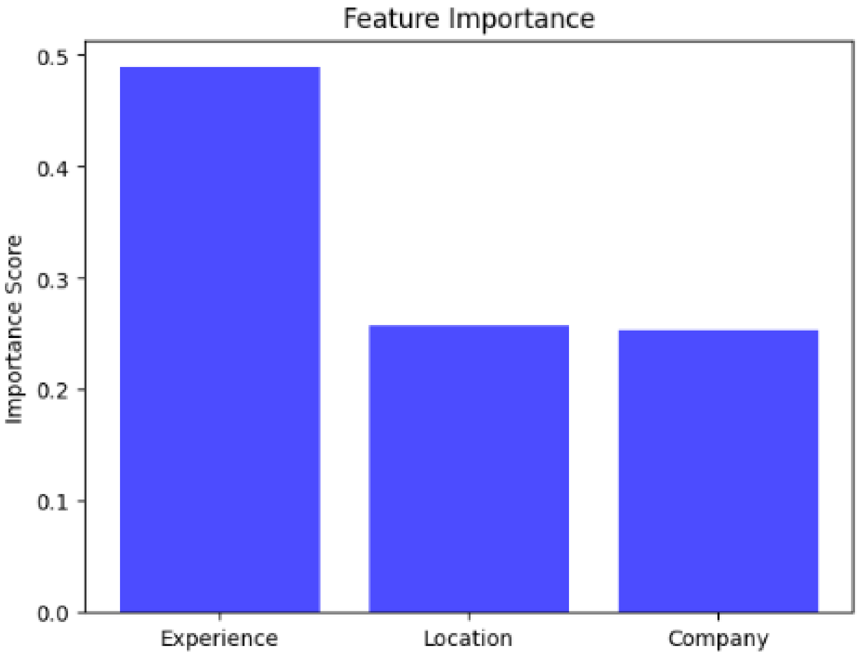


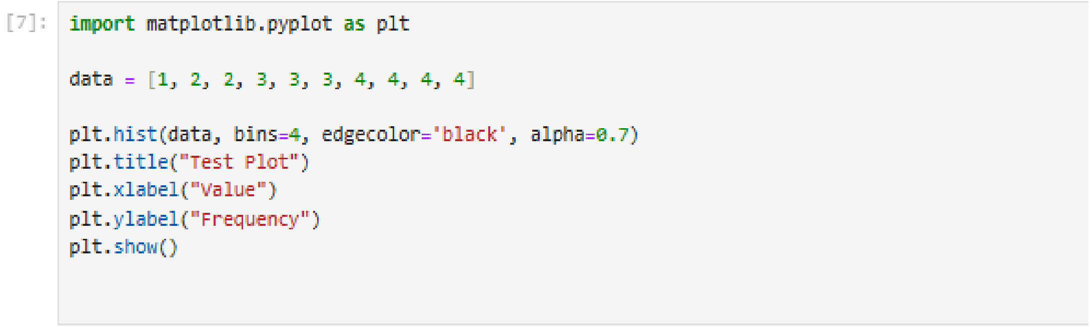




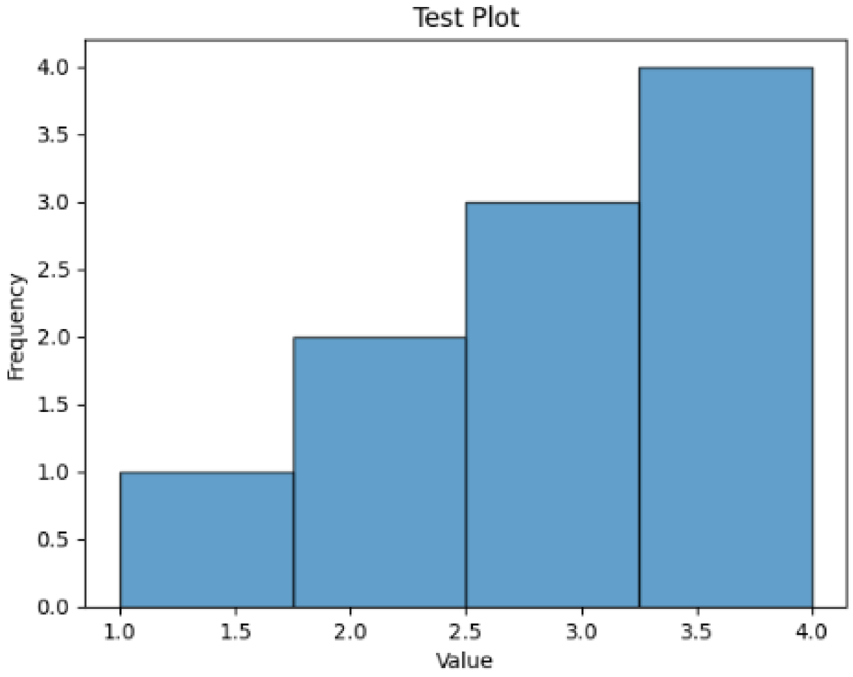
```
MSE: 1784730354.495608
R^2: -0.4984816494826516
```

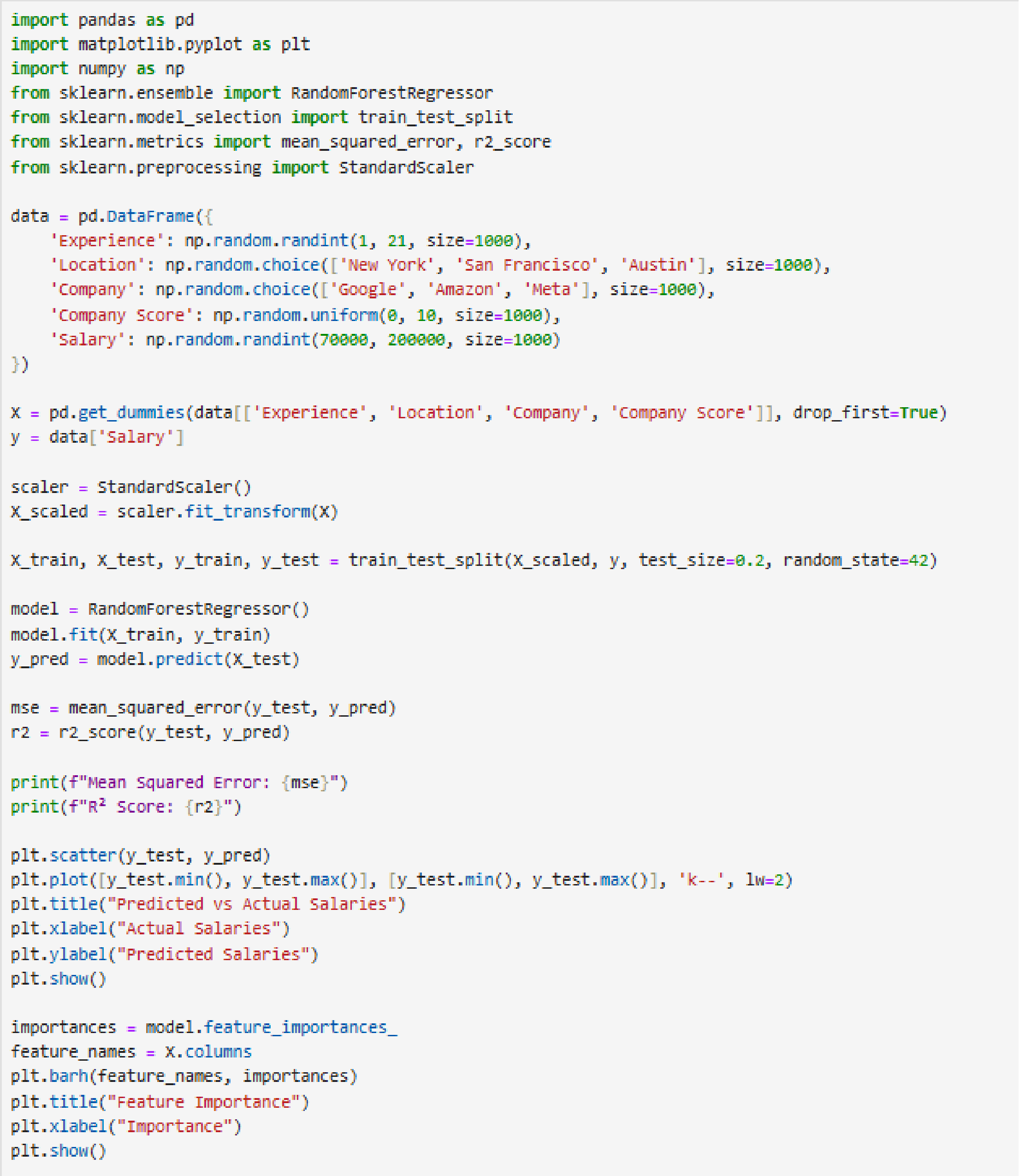




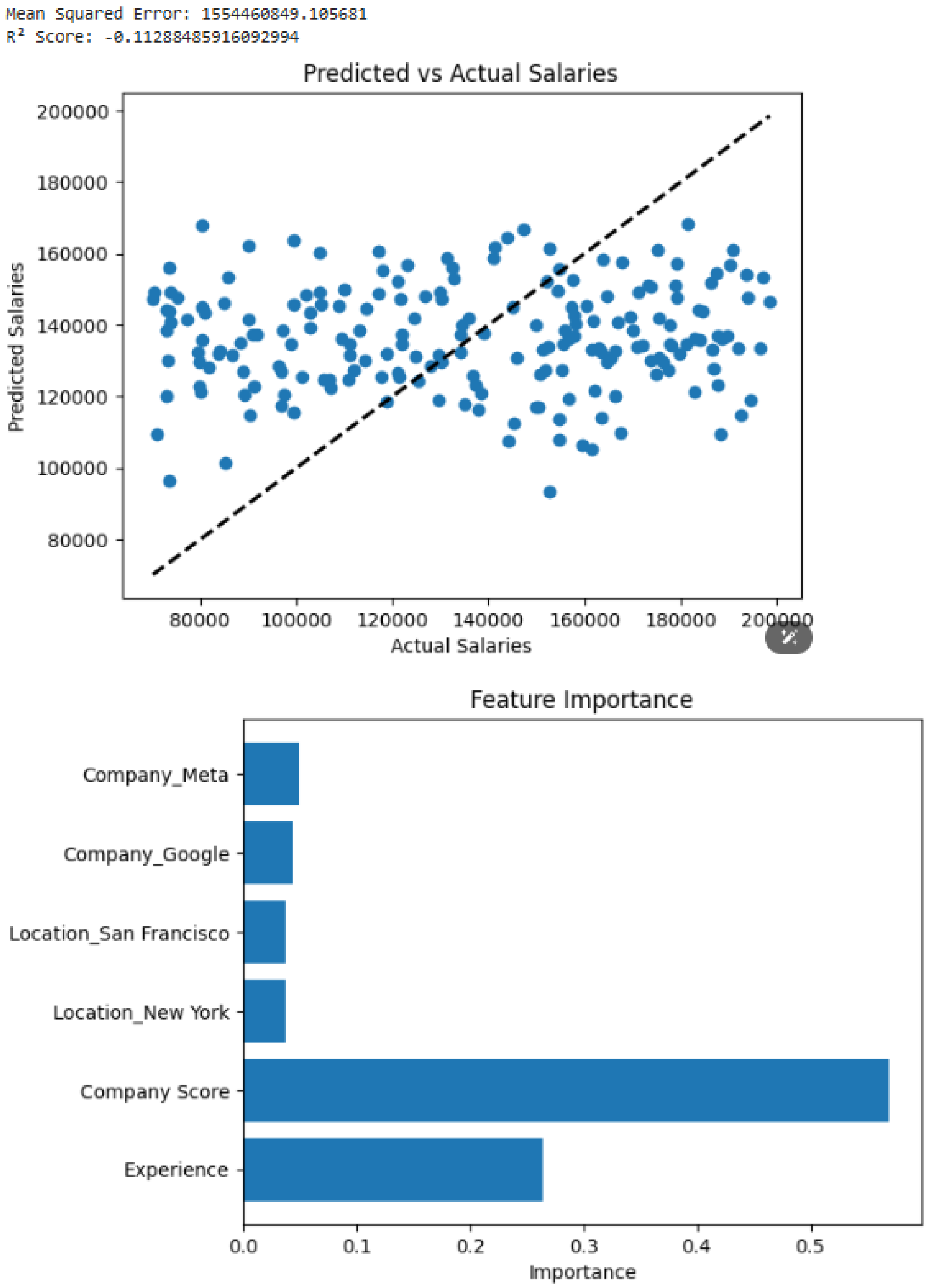

[7]:

```
import matplotlib.pyplot as plt

data = [1, 2, 2, 3, 3, 3, 4, 4, 4, 4]

plt.hist(data, bins=4, edgecolor='black', alpha=0.7)
plt.title("Test Plot")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```





```python
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score
from sklearn.preprocessing import StandardScaler

data = pd.DataFrame({
    'Experience': np.random.randint(1, 21, size=1000),
    'Location': np.random.choice(['New York', 'San Francisco', 'Austin'], size=1000),
    'Company': np.random.choice(['Google', 'Amazon', 'Meta'], size=1000),
    'Company Score': np.random.uniform(0, 10, size=1000),
    'Salary': np.random.randint(70000, 200000, size=1000)
})

X = pd.get_dummies(data[['Experience', 'Location', 'Company', 'Company Score']], drop_first=True)
y = data['Salary']

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2, random_state=42)

model = RandomForestRegressor()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"Mean Squared Error: {mse}")
print(f"R² Score: {r2}")

plt.scatter(y_test, y_pred)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'k--', lw=2)
plt.title("Predicted vs Actual Salaries")
plt.xlabel("Actual Salaries")
plt.ylabel("Predicted Salaries")
plt.show()

importances = model.feature_importances_
feature_names = X.columns
plt.barh(feature_names, importances)
plt.title("Feature Importance")
plt.xlabel("Importance")
plt.show()
```



```
Mean Squared Error: 1554460849.105681
R² Score: -0.11288485916092994
```