```
import pandas as pd

df = pd.read_csv('/content/Salary_Data[1].csv')

# Identify independent and dependent variables
X = df['Age'].values.reshape(-1, 1)
y = df['Salary']

print("Dataset loaded successfully.")
print("Shape of independent variable X:", X.shape)
print("Shape of dependent variable y:", y.shape)
print("\nFirst 5 rows of X:\n", X[:5])
print("\nFirst 5 rows of y:\n", y[:5])
```

```
Dataset loaded successfully.
Shape of independent variable X: (6704, 1)
Shape of dependent variable y: (6704,)

First 5 rows of X:
 [[32.]
 [28.]
 [45.]
 [36.]
 [52.]]

First 5 rows of y:
 0     90000.0
1     65000.0
2    150000.0
3     60000.0
4    200000.0
Name: Salary, dtype: float64
```

```
display(df.head())

print(df.shape)
```

	Age	Gender	Education Level	Job Title	Years of Experience	Salary
0	32.0	Male	Bachelor's	Software Engineer	5.0	90000.0
1	28.0	Female	Master's	Data Analyst	3.0	65000.0
2	45.0	Male	PhD	Senior Manager	15.0	150000.0
3	36.0	Female	Bachelor's	Sales Associate	7.0	60000.0
4	52.0	Male	Master's	Director	20.0	200000.0

```
(6699, 6)
```

```
from sklearn.model_selection import train_test_split

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print("Shape of X_train:", X_train.shape)
print("Shape of X_test:", X_test.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train: (5359, 1)
Shape of X_test: (1340, 1)
Shape of y_train: (5359,)
Shape of y_test: (1340,)
```

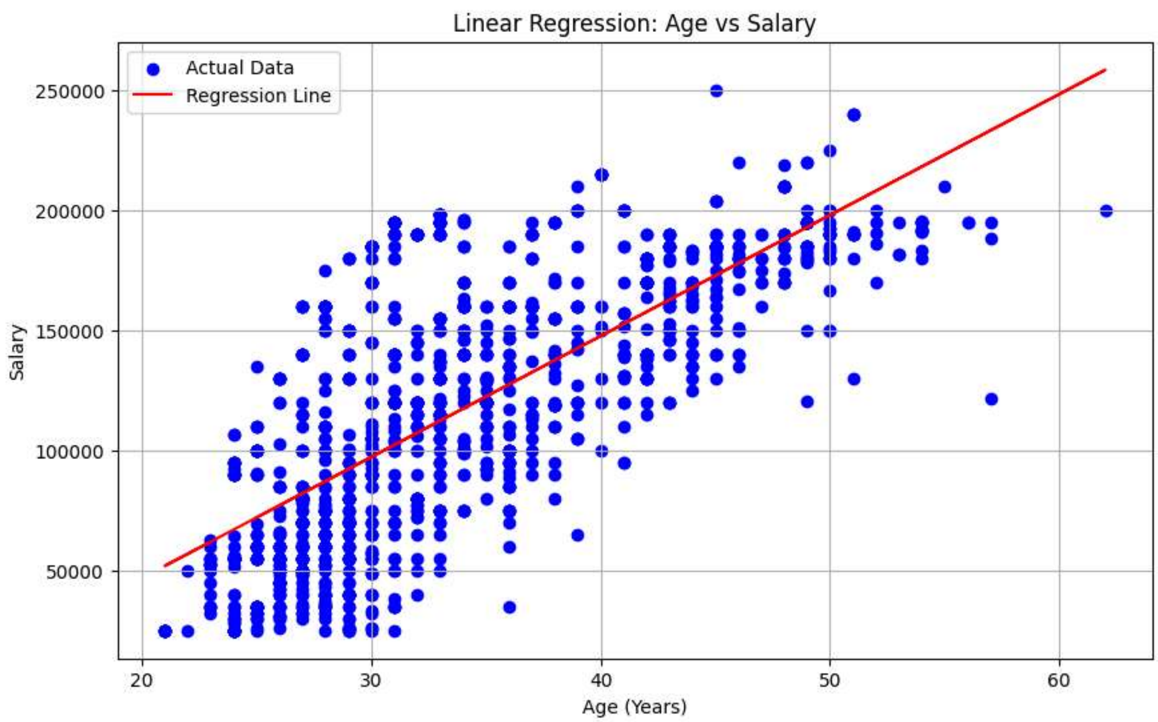
```
from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X_train, y_train)

print("Model coefficients: ", model.coef_)
print("Model intercept: ", model.intercept_)
```

```
Model coefficients:  [5038.01361961]
Model intercept:  -53978.89401482267
```

```
df.dropna(subset=['Age', 'Salary'], inplace=True)

X = df['Age'].values.reshape(-1, 1)  # Independent variable (feature)
y = df['Salary']  # Dependent variable (target)

# Train-Test Split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

model = LinearRegression()

model.fit(X_train, y_train)

print("Linear Regression model trained successfully.")
print("Model coefficients: ", model.coef_)
print("Model intercept: ", model.intercept_)
```

```
Linear Regression model trained successfully.
Model coefficients:  [5038.01361961]
Model intercept:  -53978.89401482267
```

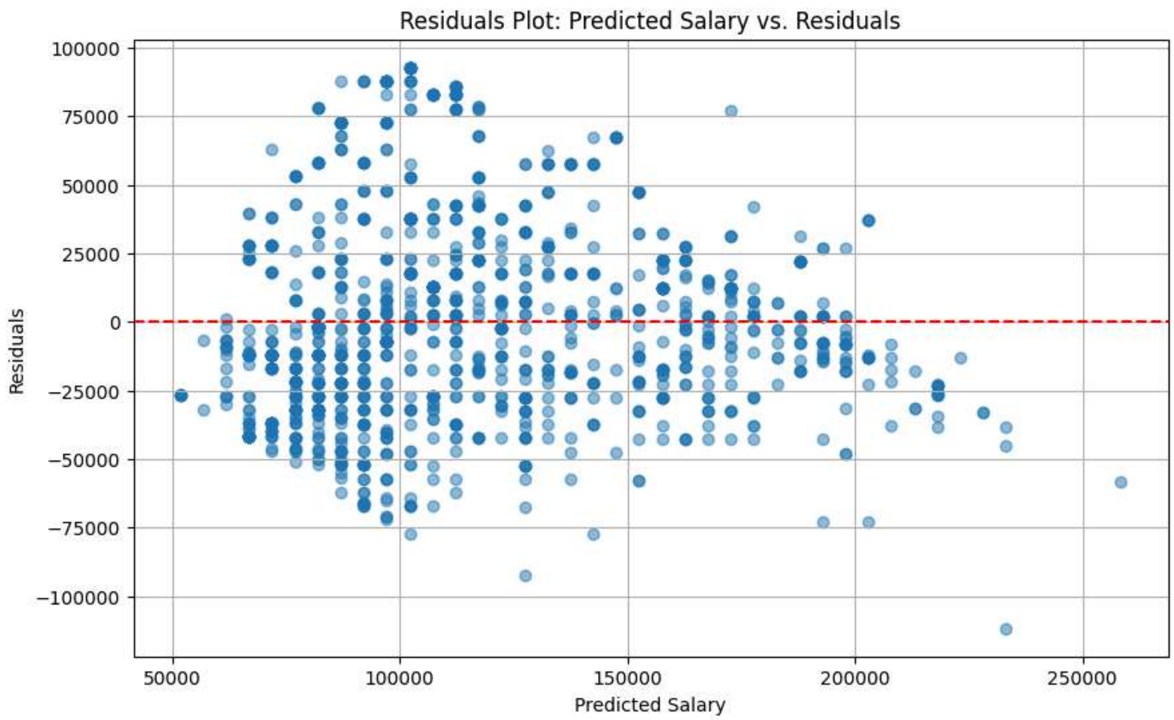
```
import matplotlib.pyplot as plt

y_pred = model.predict(X_test)

plt.figure(figsize=(10, 6))
plt.scatter(X_test, y_test, color='blue', label='Actual Data')
plt.plot(X_test, y_pred, color='red', label='Regression Line')

plt.title('Linear Regression: Age vs Salary')
plt.xlabel('Age (Years)')
plt.ylabel('Salary')
plt.legend()
plt.grid(True)
plt.show()
```



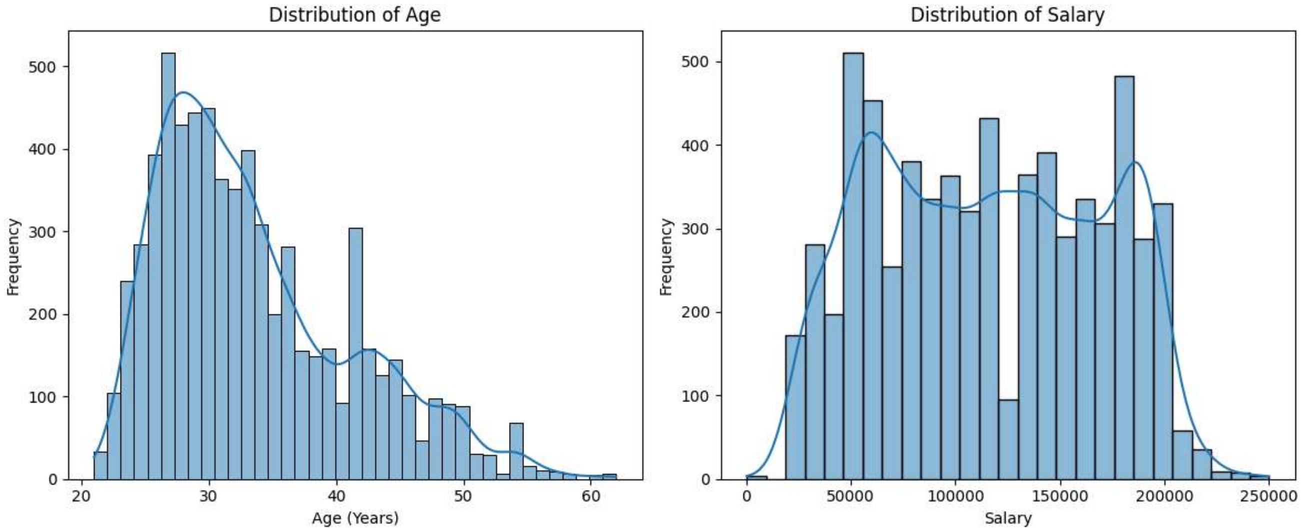

```
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

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

print(f"R-squared: {r2:.2f}")
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
```

```
R-squared: 0.54
Mean Absolute Error (MAE): 28720.92
Mean Squared Error (MSE): 1304133406.91
```

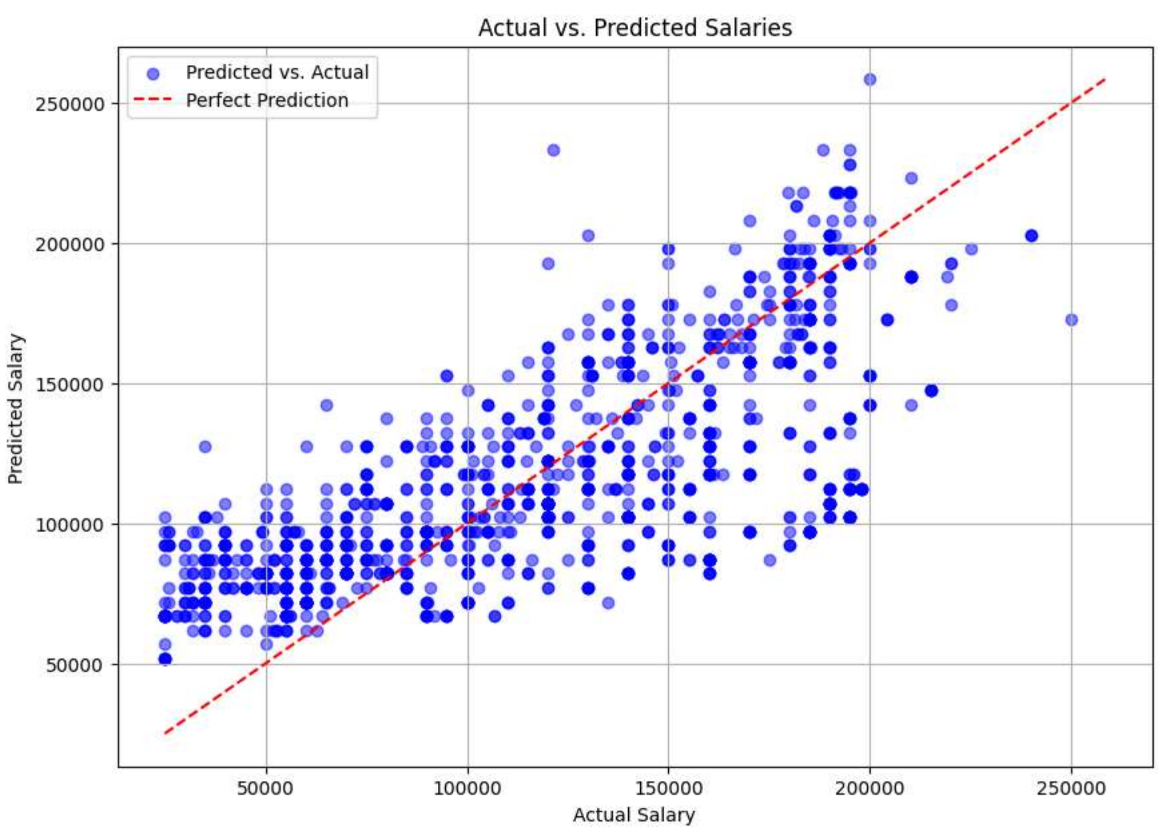
```
residuals = y_test - y_pred

plt.figure(figsize=(10, 6))
plt.scatter(y_pred, residuals, alpha=0.5)

plt.axhline(y=0, color='red', linestyle='--')

plt.xlabel('Predicted Salary')
plt.ylabel('Residuals')
plt.title('Residuals Plot: Predicted Salary vs. Residuals')

plt.grid(True)
plt.show()
```



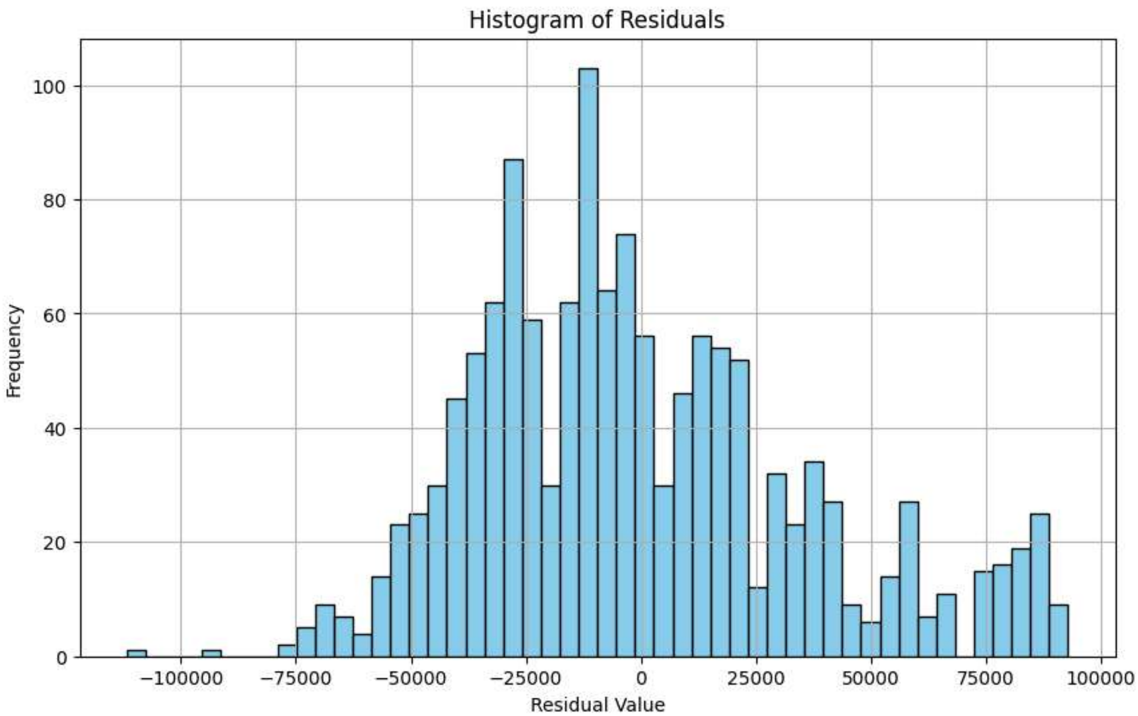

```
plt.figure(figsize=(12, 5))

plt.subplot(1, 2, 1)
sns.histplot(df['Age'], kde=True)
plt.title('Distribution of Age')
plt.xlabel('Age (Years)')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
sns.histplot(df['Salary'], kde=True)
plt.title('Distribution of Salary')
plt.xlabel('Salary')
plt.ylabel('Frequency')

plt.tight_layout()
plt.show()
```




```
import numpy as np

plt.figure(figsize=(10, 7))
plt.scatter(y_test, y_pred, color='blue', alpha=0.5, label='Predicted vs. Actual')

min_val = min(y_test.min(), y_pred.min())
max_val = max(y_test.max(), y_pred.max())
plt.plot([min_val, max_val], [min_val, max_val], color='red', linestyle='--', label='Perfect Prediction')

plt.xlabel('Actual Salary')
plt.ylabel('Predicted Salary')
plt.title('Actual vs. Predicted Salaries')
plt.legend()
plt.grid(True)
plt.show()
```

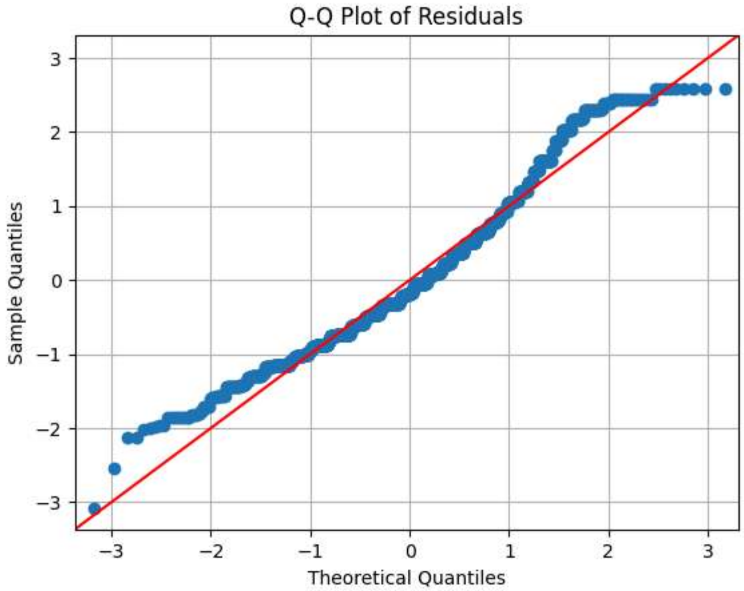


```
plt.figure(figsize=(10, 6))
plt.hist(residuals, bins=50, color='skyblue', edgecolor='black')

plt.title('Histogram of Residuals')
plt.xlabel('Residual Value')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```

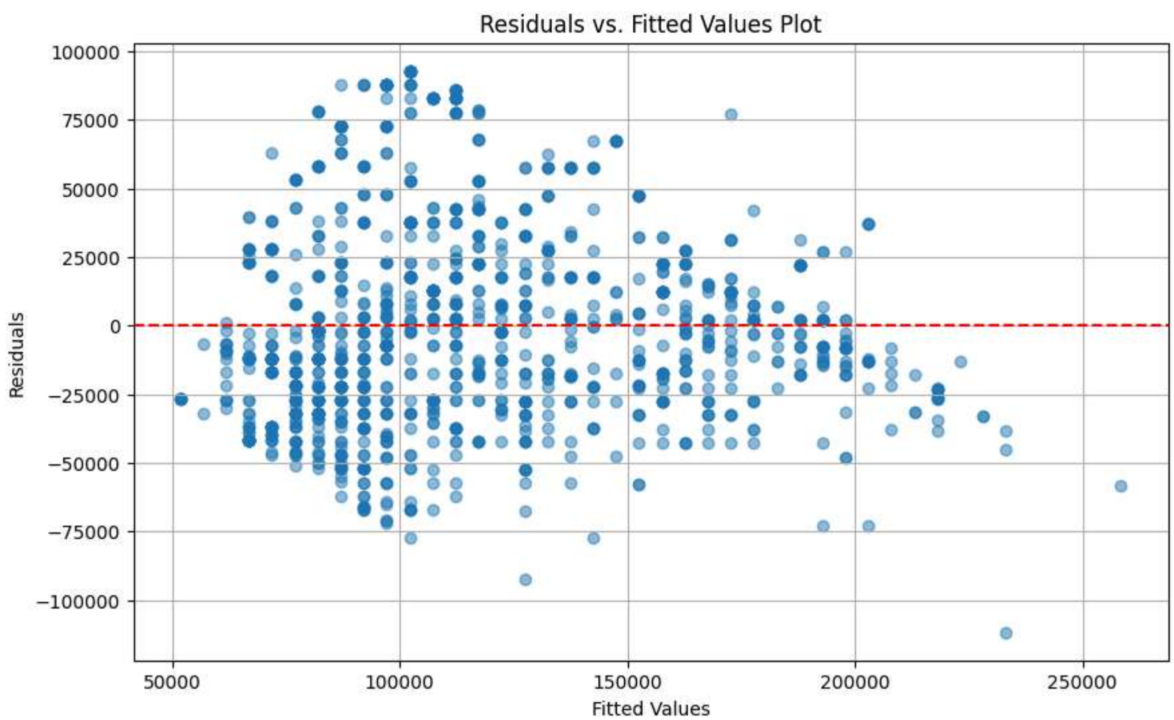



```
import statsmodels.api as sm
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
sm.qqplot(residuals, line='45', fit=True)

plt.title('Q-Q Plot of Residuals')
plt.xlabel('Theoretical Quantiles')
plt.ylabel('Sample Quantiles')
plt.grid(True)
plt.show()
```

```
<Figure size 1000x600 with 0 Axes>
```

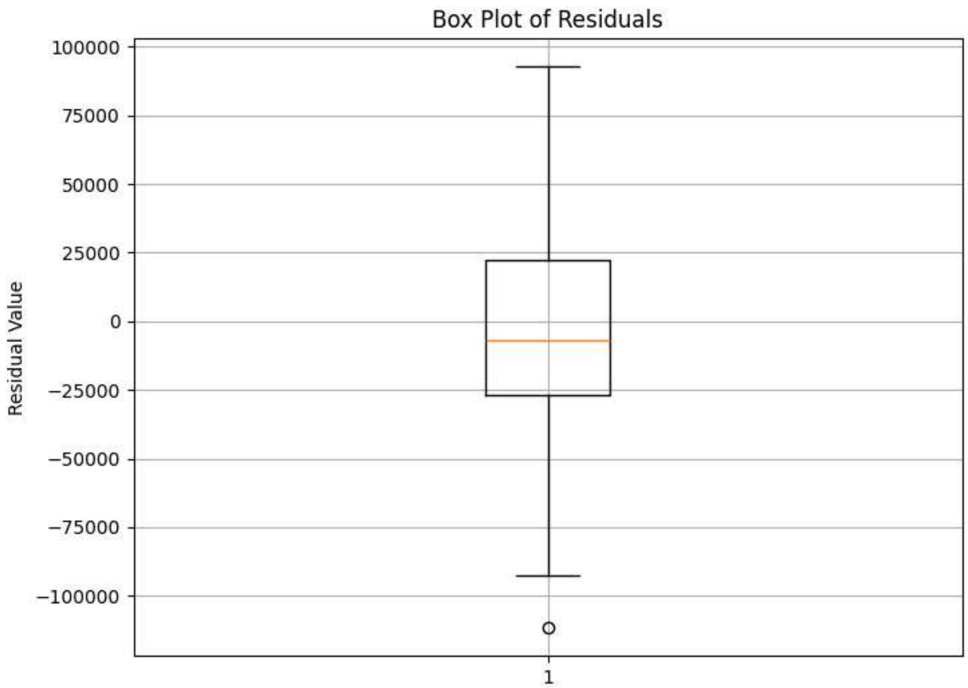



```
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.scatter(y_pred, residuals, alpha=0.5)

plt.axhline(y=0, color='red', linestyle='--')

plt.xlabel('Fitted Values')
plt.ylabel('Residuals')
plt.title('Residuals vs. Fitted Values Plot')

plt.grid(True)
plt.show()
```




```
import matplotlib.pyplot as plt

plt.figure(figsize=(8, 6))
plt.boxplot(residuals)

plt.title('Box Plot of Residuals')
```

```
plt.ylabel('Residual Value')

plt.grid(True)
plt.show()
```




```
categorical_features = ['Gender', 'Education Level', 'Job Title']

X_categorical = pd.get_dummies(df[categorical_features], drop_first=True)

X_new = pd.concat([df[['Age']], X_categorical], axis=1)
y = df['Salary']

print("Shape of new feature set X_new:", X_new.shape)
print("Shape of target variable y:", y.shape)
print("\nFirst 5 rows of X_new:\n", X_new.head())
print("\nFirst 5 rows of y:\n", y.head())
```

```
   Education Level_High School  Education Level_Master's  \
0                        False                     False
1                        False                      True
2                        False                     False
3                        False                     False
4                        False                      True

   Education Level_Master's Degree  Education Level_PhD  Education Level_phD  \
0                            False                False                False
1                            False                False                False
2                            False                 True                False
3                            False                False                False
4                            False                False                False

   Job Title_Accountant  ...  Job Title_Supply Chain Manager  \
0                 False  ...                           False
1                 False  ...                           False
2                 False  ...                           False
3                 False  ...                           False
4                 False  ...                           False

   Job Title_Technical Recruiter  Job Title_Technical Support Specialist  \
```

```
4                        False                          False

   Job Title_UX Designer  Job Title_UX Researcher  Job Title_VP of Finance  \
0                  False                    False                    False
1                  False                    False                    False
2                  False                    False                    False
3                  False                    False                    False
4                  False                    False                    False

   Job Title_VP of Operations  Job Title_Web Developer
0                       False                    False
1                       False                    False
2                       False                    False
3                       False                    False
4                       False                    False

[5 rows x 200 columns]

First 5 rows of y:
 0     90000.0
1     65000.0
2    150000.0
3     60000.0
4    200000.0
Name: Salary, dtype: float64
```

```
from sklearn.model_selection import train_test_split

X_train_new, X_test_new, y_train, y_test = train_test_split(X_new, y, test_size=0.2, random_state=42)

print("Shape of X_train_new:", X_train_new.shape)
print("Shape of X_test_new:", X_test_new.shape)
print("Shape of y_train:", y_train.shape)
print("Shape of y_test:", y_test.shape)
```

```
Shape of X_train_new: (5359, 200)
Shape of X_test_new: (1340, 200)
Shape of y_train: (5359,)
Shape of y_test: (1340,)
```

```
from sklearn.linear_model import LinearRegression

model_new = LinearRegression()
model_new.fit(X_train_new, y_train)

print("New Linear Regression model trained successfully.")
print("Model coefficients (new): ", model_new.coef_)
print("Model intercept (new): ", model_new.intercept_)
```

```
New Linear Regression model trained successfully.
Model coefficients (new):  [ 3.96912909e+03  2.25422283e+02 -1.17886661e+04 -5.52219067e+04
 -6.55469832e+04 -6.72122356e+03 -4.64694888e+04 -4.25873027e+04
  6.91190345e+03 -1.01863407e-09 -4.66047877e+04  8.47433523e+04
  1.96292912e+01  1.37829654e+04 -2.91038305e-11  1.03555676e+05
  1.39957754e+05  1.38204721e+05  7.11834050e+04  2.02271622e-09
  1.22553550e+04 -1.88147745e+04 -1.94995664e-09  3.15086543e+04
 -1.91234836e+04 -1.09598032e+04  7.47568156e+04 -1.19906741e+04
  1.11165630e+05  5.03000159e+04  4.48456454e+04 -1.68363195e+04
  7.00595041e+04  5.44540447e+04  5.23386418e+04  4.21842872e+04
  1.23426732e+05  6.01225454e+04  4.13433048e+04  6.07507492e+04
  6.69388101e-10  7.73992723e+04  8.52616047e+04  4.54310832e+04
  4.32151582e+04  8.73114914e-10  7.63684014e+04 -3.61665295e+04
  1.90083877e+04  7.49420112e+04  1.07758491e+05  7.09105981e+04
  7.54944196e+04  9.32199195e+04  4.80695490e+04 -2.66047877e+04
 -1.24041268e+04 -1.22160964e+04  3.88531278e+04  4.84405804e+04
  6.35346120e+04 -1.05678072e+04 -1.70617418e+04 -3.08765164e+04
 -1.49644661e+04 -1.61337740e+04 -2.20617418e+04 -2.01693179e+04
 -2.17511852e+04 -6.27426127e+04 -2.18363195e+04 -1.61852255e+04
  4.53691602e+04  7.61420372e+02 -3.90514288e-11 -3.24696730e+03
 -8.73114914e-11 -1.89078695e+04  3.10144173e+04  3.81412547e+04
 -8.38262591e+03 -2.35781270e+04  4.51663528e+04 -1.00673626e+04
 -1.87322331e+04  1.45519152e-11 -1.38564179e+04 -1.57731532e+04
 -1.81990307e+04 -1.78671905e+04 -7.27595761e-12  3.83560344e+04
  3.64384079e+04 -2.18363195e+04 -2.58054486e+04  3.55880163e+04
  5.18874718e+04 -3.20770872e+03 -1.80926127e+04  3.64772472e+04
  5.25408737e+04  4.95362107e+04  5.87649928e+04  5.79337420e+04
  1.36053929e+05  7.86626861e+04  2.00088834e-11 -7.06174182e+03
 -4.15276104e+04 -1.65986781e+04  7.82165444e-11  6.66098966e+04
  7.40812373e+04  7.17107863e+04  5.56861846e+04  9.92954044e+04
 -6.54836185e-11  4.72937245e-11  9.35052097e+04 -1.86812868e+03
  4.91130899e+04 -3.25584813e+04  1.22550006e+05  1.21489100e+05
  4.65106305e+04  9.55837498e+04  4.99690309e+04  7.55360034e+04
 -1.65986781e+04  4.35982944e+04 -7.55848133e+03  3.47238958e+03
```

```
 -9.73293430e+03  3.22711127e+04  3.04995309e+04  6.41523050e+04
  6.70194955e+04  7.30859002e+04  1.03596498e+05  6.39268827e+04
  1.73417210e+04  1.53649783e+04  8.20532388e+03  2.25535497e+03
  4.60901124e+04  2.03479677e+04  6.36646291e-11 -2.72848411e-11
  5.72127799e+04  2.01936131e+04  1.79064490e+04  1.13395800e+04
 -1.26295491e+04  7.40040600e+04 -8.45788178e+02 -6.65420270e+03
  9.20905632e+04  2.01279207e+04  5.22949532e+03 -1.90738726e+03
  1.18447105e+04  5.62457048e+03  6.76831759e+04 -9.67759206e-12
  2.59942310e+04  7.38735385e+04  1.15989235e+02  9.57245561e+04
  1.76349285e+04  1.70617418e+04  8.50781282e+04 -1.23691279e-10
  3.91675732e+03 -9.54304588e+03  5.87725282e+04  1.08047971e-09
  2.44527554e+04  9.11375578e+04  1.09139364e-11  7.62140468e+04
  1.04262103e+05  5.27660986e+04 -1.38980614e+04  6.41092831e+04
  8.27986198e+04  8.71971092e+04  3.34694050e-10  1.54354561e+02
  3.63797881e-12 -1.62189013e+04 -1.36604200e+04  1.47117873e-11
 -7.27595761e-12 -1.01543546e+04 -3.75584813e+04  4.00838767e+03
  1.67922913e+04  7.21842872e+04  6.21842872e+04  4.89041177e+04]
Model intercept (new):  -52237.55309050085
```

```
from sklearn.metrics import r2_score, mean_absolute_error, mean_squared_error

y_pred_new = model_new.predict(X_test_new)

r2_new = r2_score(y_test, y_pred_new)
mae_new = mean_absolute_error(y_test, y_pred_new)
mse_new = mean_squared_error(y_test, y_pred_new)

print(f"New Model R-squared: {r2_new:.2f}")
print(f"New Model Mean Absolute Error (MAE): {mae_new:.2f}")
print(f"New Model Mean Squared Error (MSE): {mse_new:.2f}")
```

```
New Model R-squared: 0.86
New Model Mean Absolute Error (MAE): 14075.46
New Model Mean Squared Error (MSE): 396646386.14
```

```
feature_importances = pd.Series(model_new.coef_, index=X_new.columns)
sorted_importances = feature_importances.sort_values(ascending=False)

print("Top 10 Feature Importances (Coefficients):")
print(sorted_importances.head(10))

print("\nBottom 10 Feature Importances (Coefficients):")
print(sorted_importances.tail(10))
```

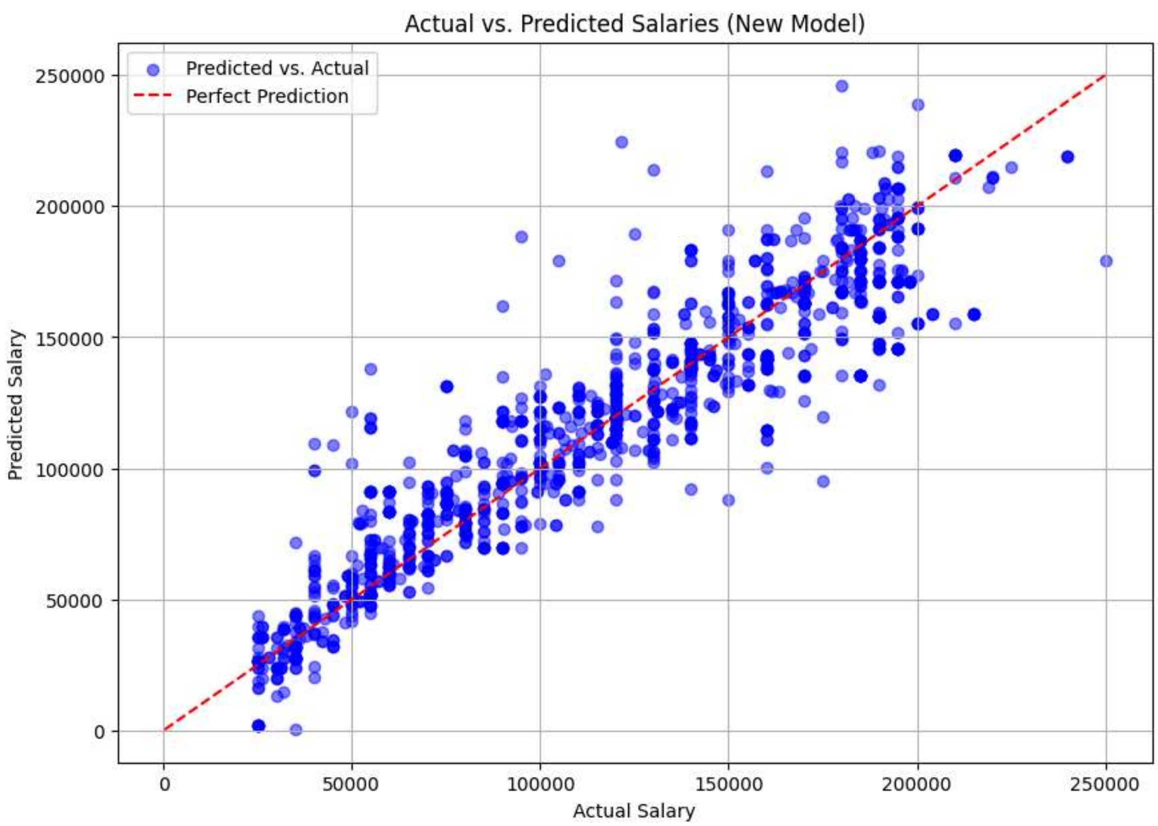
```
Top 10 Feature Importances (Coefficients):
Job Title_Chief Data Officer          139957.753643
Job Title_Chief Technology Officer    138204.720941
Job Title_Marketing Director          136053.929037
Job Title_Director of Data Science    123426.731799
Job Title_Research Director           122550.006127
Job Title_Research Scientist          121489.099574
Job Title_Data Scientist              111165.630143
Job Title_Financial Manager           107758.490656
Job Title_Social Media Man            104262.103498
Job Title_Senior Data Scientist       103596.498197
dtype: float64

Bottom 10 Feature Importances (Coefficients):
Job Title_Recruiter                           -32558.481332
Job Title_Event Coordinator                   -36166.529524
Job Title_Training Specialist                 -37558.481332
Job Title_Office Manager                      -41527.610419
Education Level_PhD                           -42587.302696
Education Level_Master's Degree               -46469.488815
Job Title_Administrative Assistant            -46604.787700
Education Level_Bachelor's Degree             -55221.906688
Job Title_Junior Business Operations Analyst  -62742.612737
Education Level_High School                   -65546.983243
dtype: float64
```

```
residuals_new = y_test - y_pred_new

print("Residuals:\n", residuals_new.head())
```

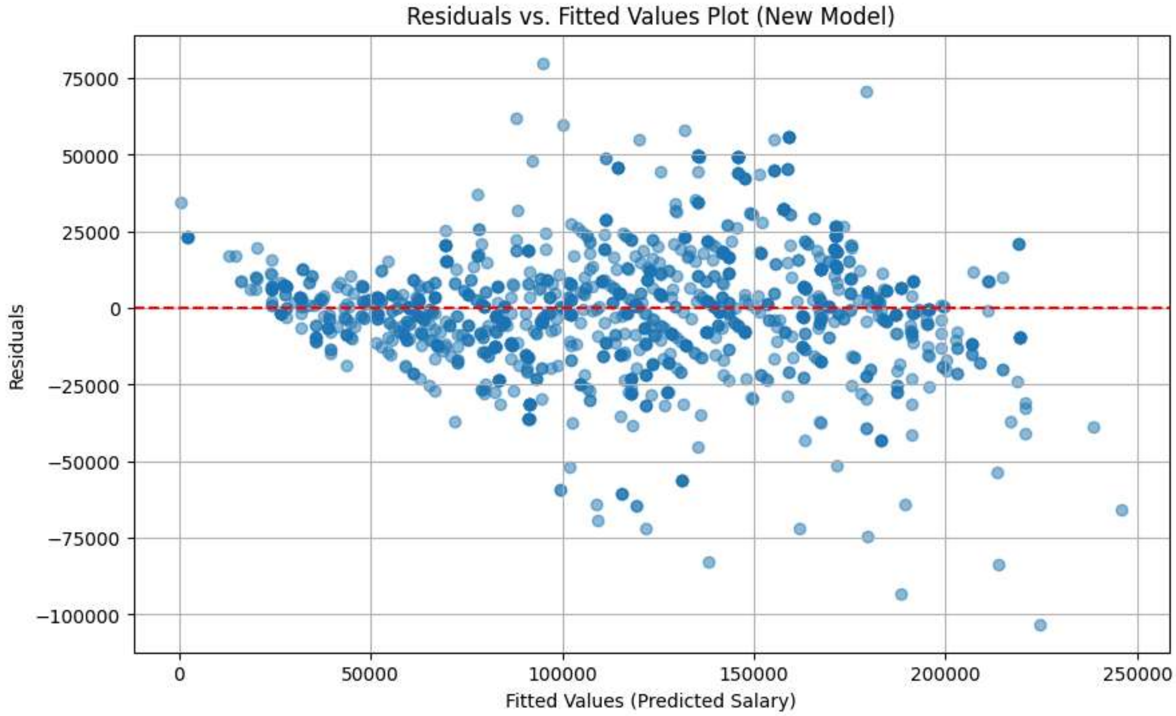
```
Residuals:
 2460    13511.458045
2230    12148.281129
5559    -2731.067714
3080    -8586.412867
265    -59305.598380
```

```
Name: Salary, dtype: float64
```

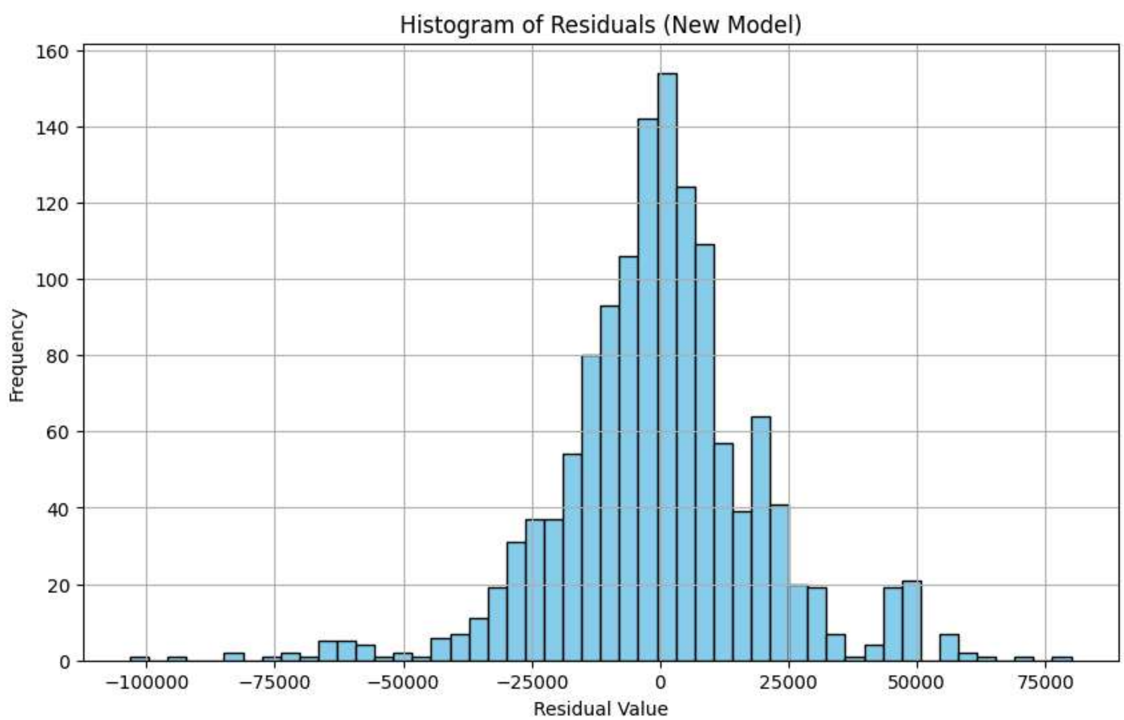
```
plt.figure(figsize=(10, 7))
plt.scatter(y_test, y_pred_new, color='blue', alpha=0.5, label='Predicted vs. Actual')

min_val_new = min(y_test.min(), y_pred_new.min())
max_val_new = max(y_test.max(), y_pred_new.max())
plt.plot([min_val_new, max_val_new], [min_val_new, max_val_new], color='red', linestyle='--', label='Perfect Prediction')

plt.xlabel('Actual Salary')
plt.ylabel('Predicted Salary')
plt.title('Actual vs. Predicted Salaries (New Model)')
plt.legend()
plt.grid(True)
plt.show()
```




```
plt.figure(figsize=(10, 6))
plt.scatter(y_pred_new, residuals_new, alpha=0.5)

plt.axhline(y=0, color='red', linestyle='--')

plt.xlabel('Fitted Values (Predicted Salary)')
plt.ylabel('Residuals')
plt.title('Residuals vs. Fitted Values Plot (New Model)')

plt.grid(True)
plt.show()
```



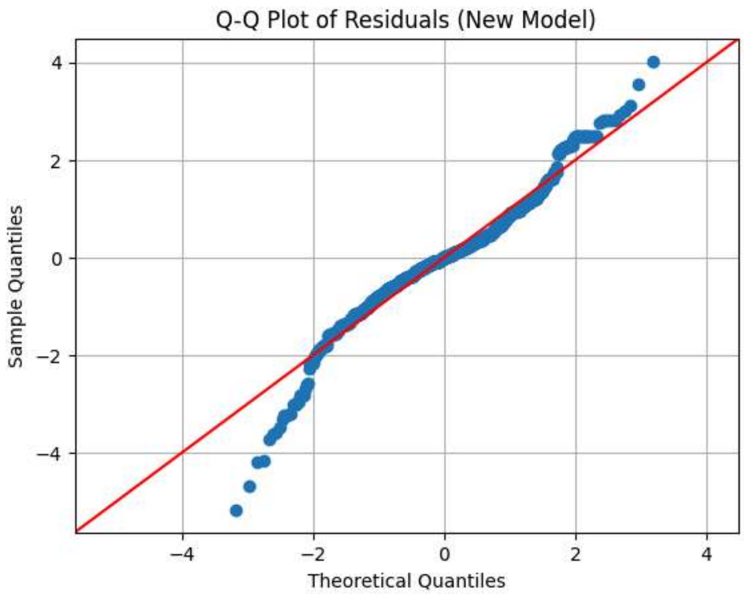

```
plt.figure(figsize=(10, 6))
plt.hist(residuals_new, bins=50, color='skyblue', edgecolor='black')

plt.title('Histogram of Residuals (New Model)')
plt.xlabel('Residual Value')
plt.ylabel('Frequency')
plt.grid(True)
plt.show()
```



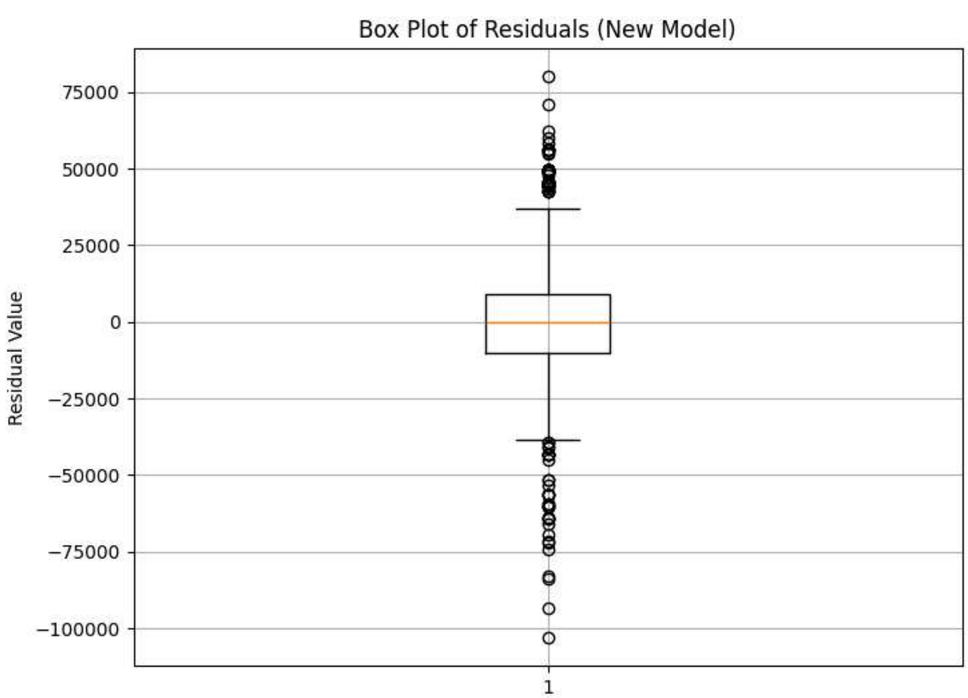

```
plt.figure(figsize=(10, 6))
sm.qqplot(residuals_new, line='45', fit=True)

plt.title('Q-Q Plot of Residuals (New Model)')
plt.xlabel('Theoretical Quantiles')
plt.ylabel('Sample Quantiles')
```

```
plt.grid(True)
plt.show()
```

<Figure size 1000x600 with 0 Axes>

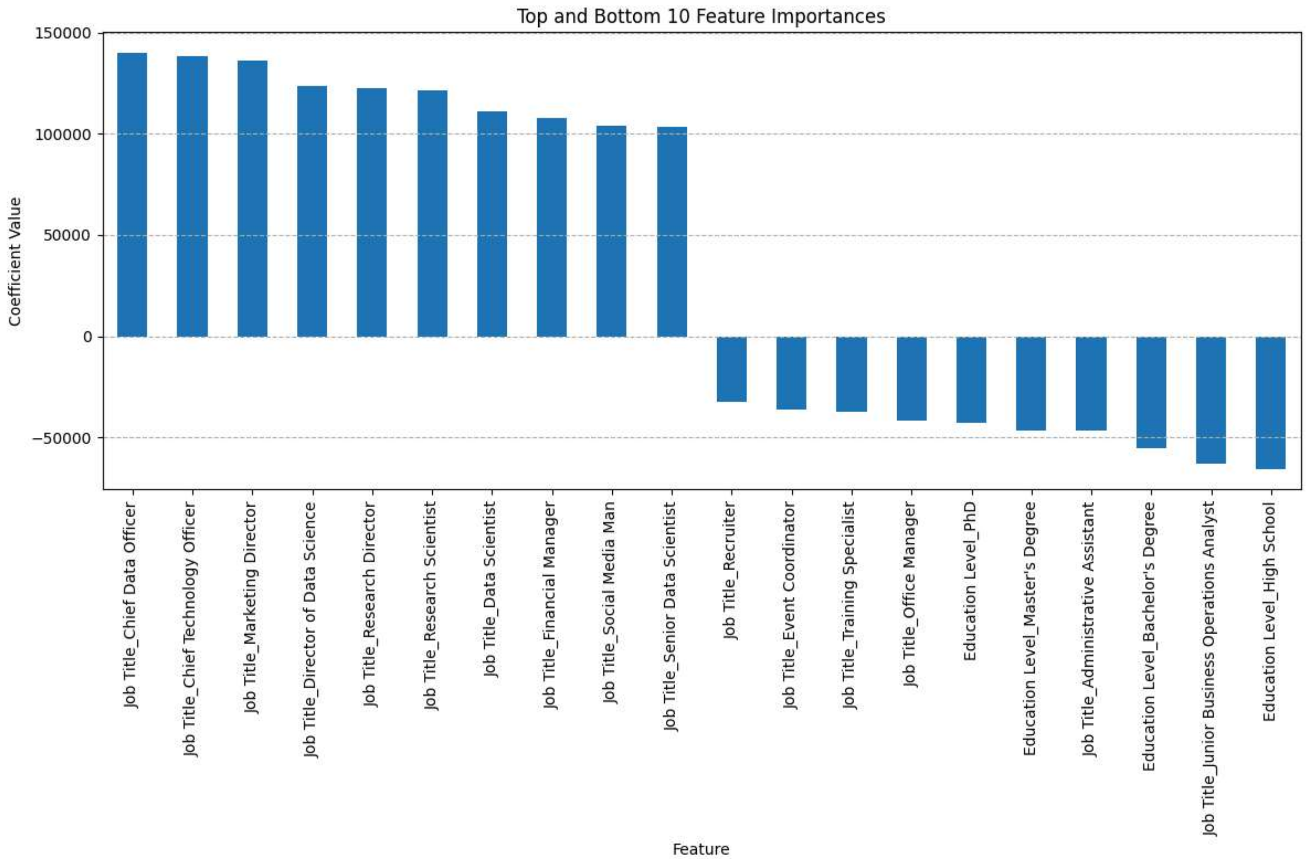



```
plt.figure(figsize=(8, 6))
plt.boxplot(residuals_new)

plt.title('Box Plot of Residuals (New Model)')
plt.ylabel('Residual Value')

plt.grid(True)
plt.show()
```




```
top_10_features = sorted_importances.head(10)
bottom_10_features = sorted_importances.tail(10)

selected_features = pd.concat([top_10_features, bottom_10_features])

plt.figure(figsize=(12, 8))
selected_features.plot(kind='bar')
plt.title('Top and Bottom 10 Feature Importances')
plt.xlabel('Feature')
```

```
plt.ylabel('Coefficient Value')
plt.xticks(rotation=90)
plt.grid(axis='y', linestyle='--')
plt.tight_layout()
plt.show()
```




```
correlation_matrix = X_new.corr()

print("Correlation Matrix (first 5x5 elements):")
display(correlation_matrix.head())

plt.figure(figsize=(15, 12))
sns.heatmap(correlation_matrix, annot=False, cmap='coolwarm', fmt=".2f")
plt.title('Correlation Matrix of Features in X_new')
plt.show()
```

```
Correlation Matrix (first 5x5 elements):
```

	Age	Gender_Male	Gender_Other	Education Level_Bachelor's Degree	Education Level_High School	Education Level_Master's	Education Level_Master's Degree	Education Level_PhD	Ed Le
Age	1.000000	0.114892	0.035753	-0.306717	-0.237900	0.007621	0.129663	0.501786	-
Gender_Male	0.114892	1.000000	-0.050403	0.069402	-0.072724	-0.101833	-0.113724	0.090881	
Gender_Other	0.035753	-0.050403	1.000000	-0.032708	0.144772	-0.009699	-0.009914	-0.023182	-
Education Level_Bachelor's Degree	-0.306717	0.069402	-0.032708	1.000000	-0.191338	-0.151485	-0.395759	-0.362055	-
Education Level_High School	-0.237900	-0.072724	0.144772	-0.191338	1.000000	-0.056741	-0.148238	-0.135614	-

5 rows × 200 columns