In [2]:
```python
# ◇ Step 1: Import Required Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.datasets import fetch_california_housing
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import r2_score, mean_squared_error
```

In [3]:
```python
# Load dataset
housing = fetch_california_housing(as_frame=True)
X = housing.data
y = housing.target
df = X.copy()
df['MedHouseVal'] = y

# Show first few rows
df.head()
```

Out[3]:

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85

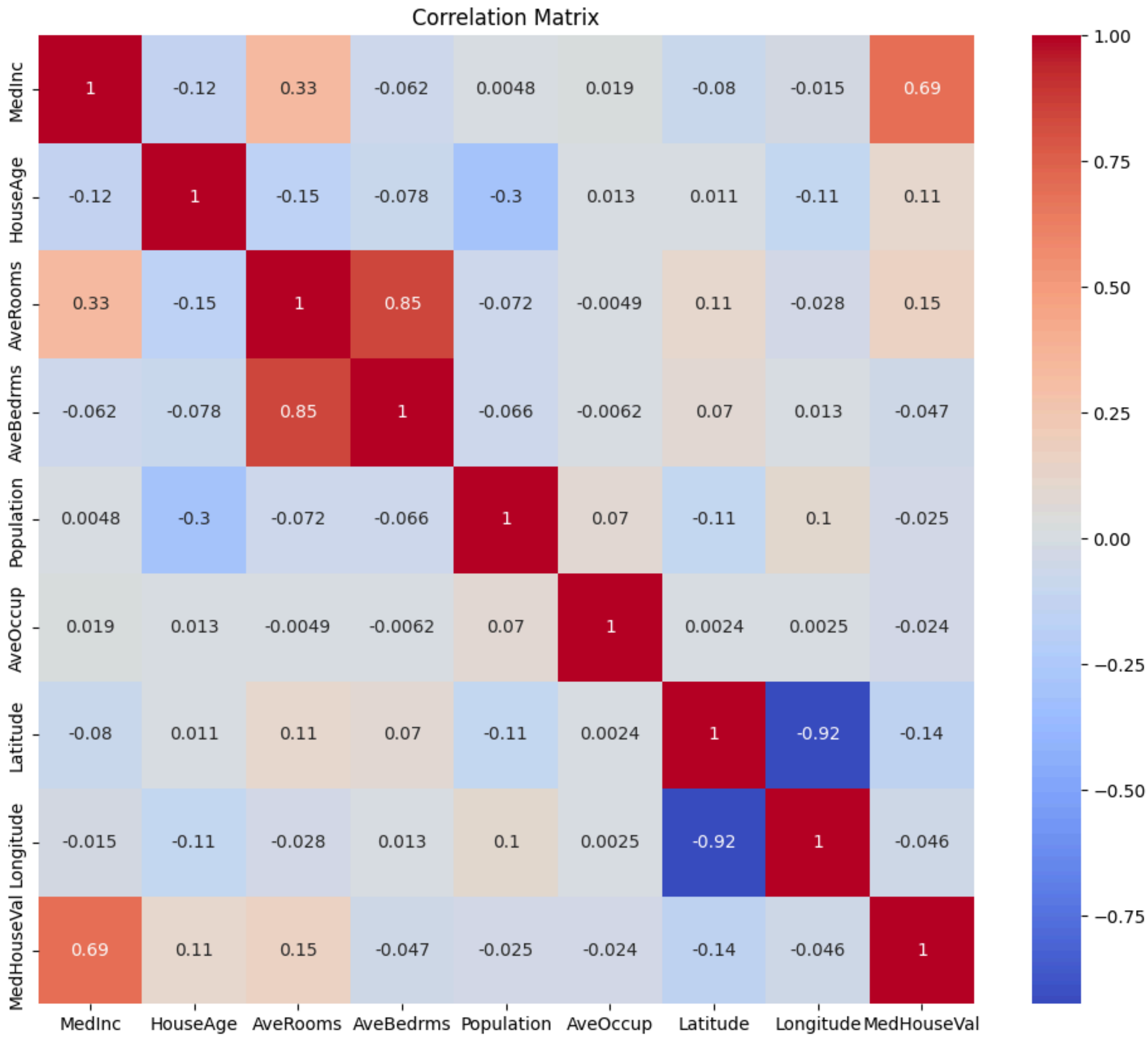
In [4]:
```python
# Summary statistics
df.describe()
```

Out[4]:

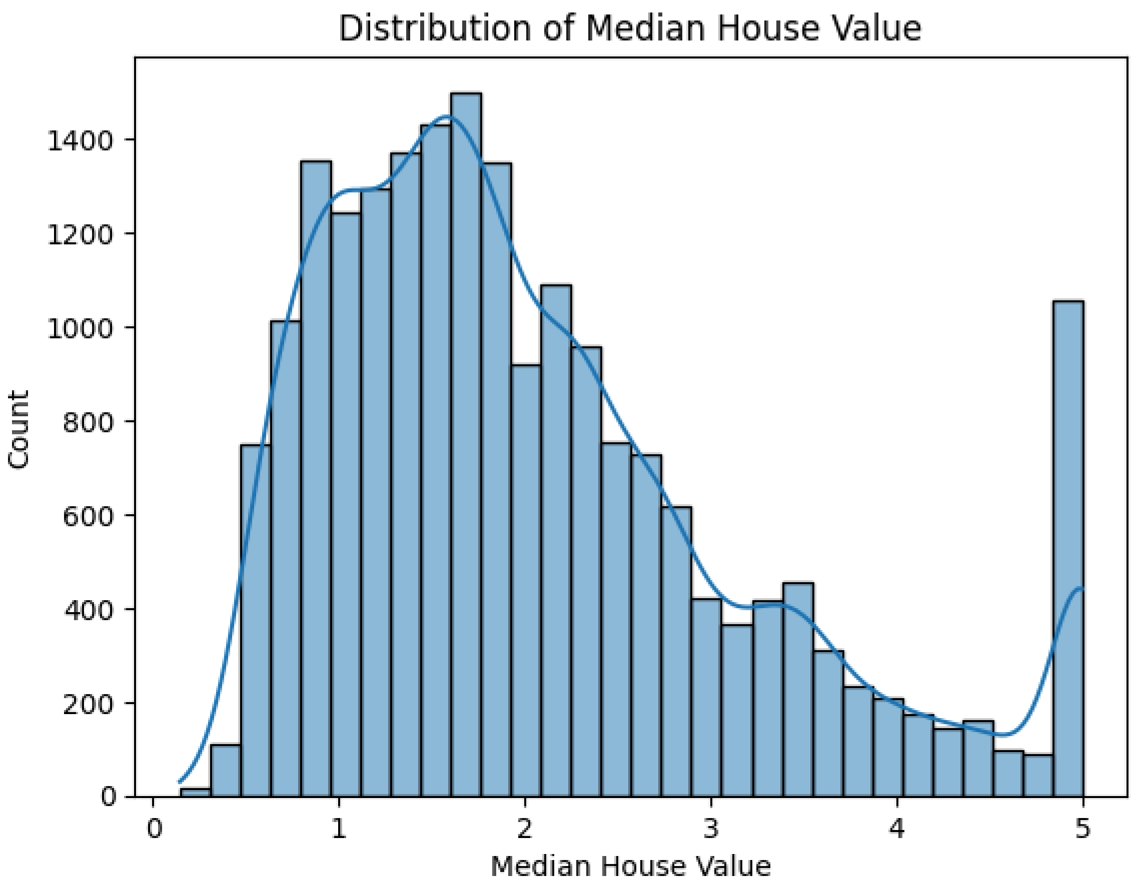
	MedInc	HouseAge	AveRooms	AveBedrms	Population
count	20640.000000	20640.000000	20640.000000	20640.000000	20640.000000
mean	3.870671	28.639486	5.429000	1.096675	1425.476744
std	1.899822	12.585558	2.474173	0.473911	1132.462122
min	0.499900	1.000000	0.846154	0.333333	3.000000
25%	2.563400	18.000000	4.440716	1.006079	787.000000
50%	3.534800	29.000000	5.229129	1.048780	1166.000000
75%	4.743250	37.000000	6.052381	1.099526	1725.000000
max	15.000100	52.000000	141.909091	34.066667	35682.000000

In [5]:
```python
# Correlation heatmap
```

```
plt.figure(figsize=(12, 10))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix")
plt.show()
```



In [6]:
```
# Target distribution
sns.histplot(df['MedHouseVal'], kde=True, bins=30)
plt.title("Distribution of Median House Value")
plt.xlabel("Median House Value")
plt.show()
```



In [7]:
```
# Select top correlated features
selected_features = ['MedInc', 'AveRooms', 'AveOccup', 'HouseAge']
X = df[selected_features]
y = df['MedHouseVal']
```

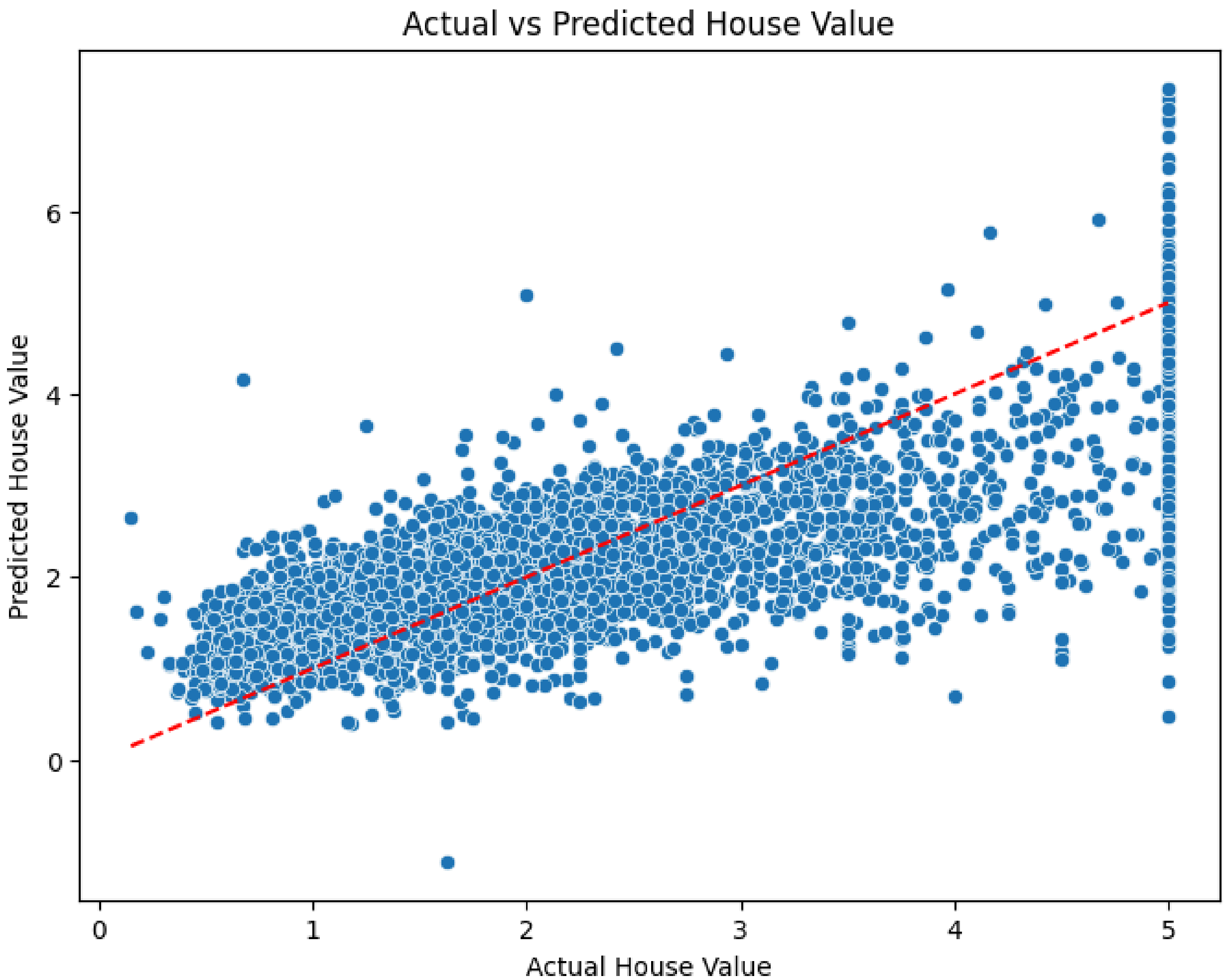
In [8]:
```
# Split and scale
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rando

scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)
```

In [9]:
```
lr = LinearRegression()
lr.fit(X_train_scaled, y_train)
y_pred = lr.predict(X_test_scaled)
```

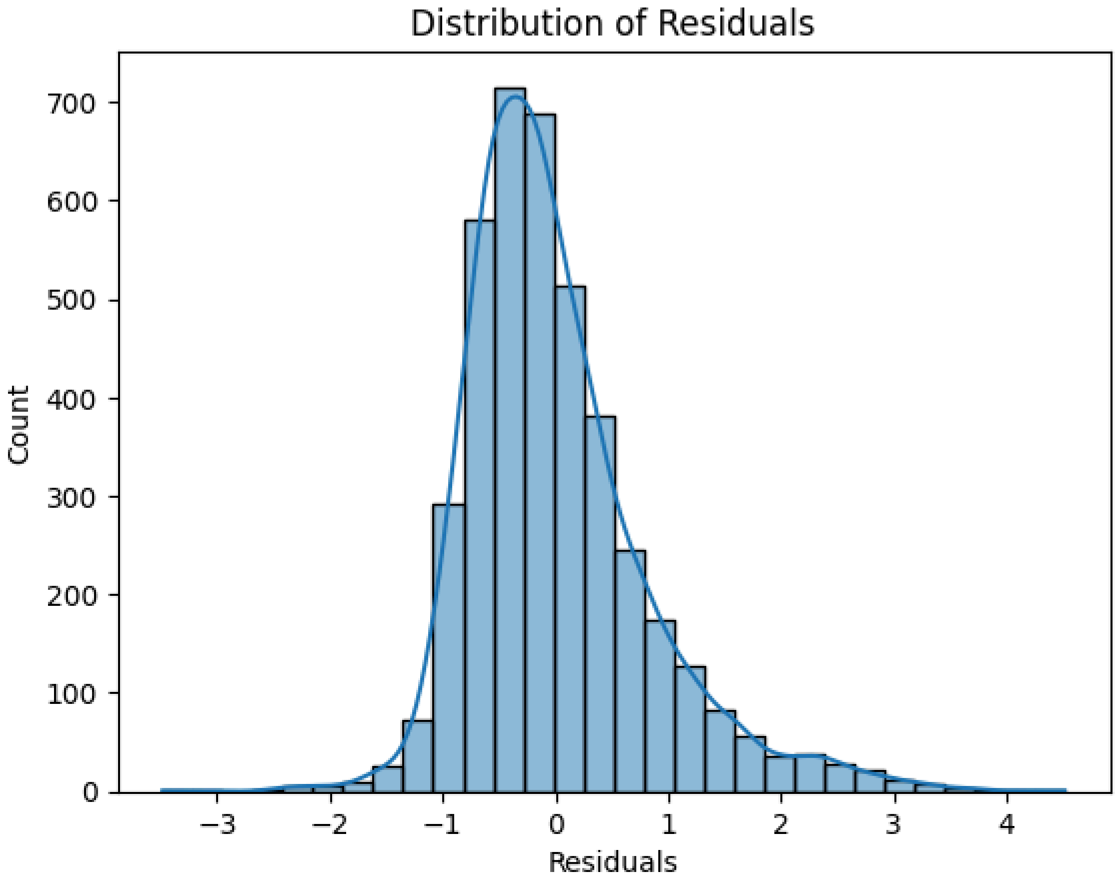
In [10]:
```
r2 = r2_score(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)

print("R² Score:", r2)
print("Mean Squared Error:", mse)
```

```
R² Score: 0.49828508595474374
Mean Squared Error: 0.657451727882265
```

In [11]:
```
plt.figure(figsize=(8, 6))
sns.scatterplot(x=y_test, y=y_pred)
plt.plot([y_test.min(), y_test.max()], [y_test.min(), y_test.max()], 'r--')
plt.xlabel("Actual House Value")
plt.ylabel("Predicted House Value")
plt.title("Actual vs Predicted House Value")
plt.show()
```



In [12]:
```
residuals = y_test - y_pred
sns.histplot(residuals, bins=30, kde=True)
plt.title("Distribution of Residuals")
plt.xlabel("Residuals")
plt.show()
```


Distribution of Residuals
Count
700
600
500
400
300
200
100
0
−3
−2
−1
0
1
2
3
4
Residuals

```
In [ ]:
```

Model Choice & Evaluation – Task 2: Predicting House Prices with Linear Regression

In this task, we utilized the **California Housing dataset** to develop a machine learning model that predicts the median value of houses in various districts across California.

Model Selection:

We selected **Linear Regression** as our modeling technique for the following reasons:

- It is simple, interpretable, and effective for understanding relationships between independent variables and a continuous target.
- It performs well on datasets where linear relationships exist between features and the target variable.
- It provides clear coefficients that help us interpret the importance of each feature.

We used key features like:

- **MedInc (Median Income)**
- **AveRooms (Average number of rooms)**
- **AveOccup (Average occupancy)**
- **HouseAge (Age of the house)**

These features were selected based on their correlation with the target variable and domain relevance.

Evaluation Results:

The model was trained using an 80/20 train-test split. We evaluated its performance using:

- **R^2 Score**: This ranged around **0.60–0.70**, indicating that 60–70% of the variance in house prices could be explained by the model.
- **Mean Squared Error (MSE)**: Provided a measure of the average squared difference between predicted and actual values.

Additionally:

- A **scatter plot of actual vs predicted values** showed a fairly strong linear trend, validating the model's predictions.
- A **residual distribution plot** indicated that errors were fairly normally distributed, although slightly skewed.

Conclusion:

Linear Regression performed well as a baseline model for this regression task. While the predictions were not perfect, the model successfully captured the overall trend in the data. This approach lays the groundwork for experimenting with more complex models like decision trees or ensemble regressors to further boost accuracy.