Descriptive Data Analysis:

Descriptive data analysis primarily involves summarizing and understanding the basic characteristics and features of the dataset without making in-depth inferences or predictions.

1. What is the age distribution in the dataset?
2. What is the distribution of estimated salaries?
3. How many people made a purchase (Purchased=1) and how many did not (Purchased=0)?
4. What is the overall percentage of people who made a purchase?
5. What is the average age and estimated salary of those who made a purchase compared to those who did not?
6. Are there any missing values in the dataset, and if so, in which columns?
7. What is the gender distribution in the dataset (if not included in the provided subset)?
8. What is the relationship between age and estimated salary?

Exploratory Data Analysis:

Exploratory data analysis goes a step further by investigating relationships, patterns, and trends within the data. It often includes visualization and statistical techniques to uncover insights and identify interesting patterns.

1. Is there a correlation between age and the likelihood of making a purchase?
2. Is there a correlation between estimated salary and the likelihood of making a purchase?
3. How does the distribution of estimated salary differ between those who made a purchase and those who did not?
4. Are there any outliers in the dataset, and how do they affect the analysis?

Segmentation and Customer Profiling:

Segmentation and customer profiling involve dividing your data into distinct groups or segments based on certain characteristics and understanding the unique traits and behaviors of these groups.

1. Can you segment the data into different customer groups based on age and estimated salary?
2. What are the characteristics of the customer segments, and how do they differ in terms of purchase behavior?

Feature Importance Analysis:

Feature importance analysis is often part of predictive modeling and understanding which features or variables have the most influence on the target variable (in your case, purchase decisions). It helps identify the key drivers of the outcome. The question related to feature importance is:

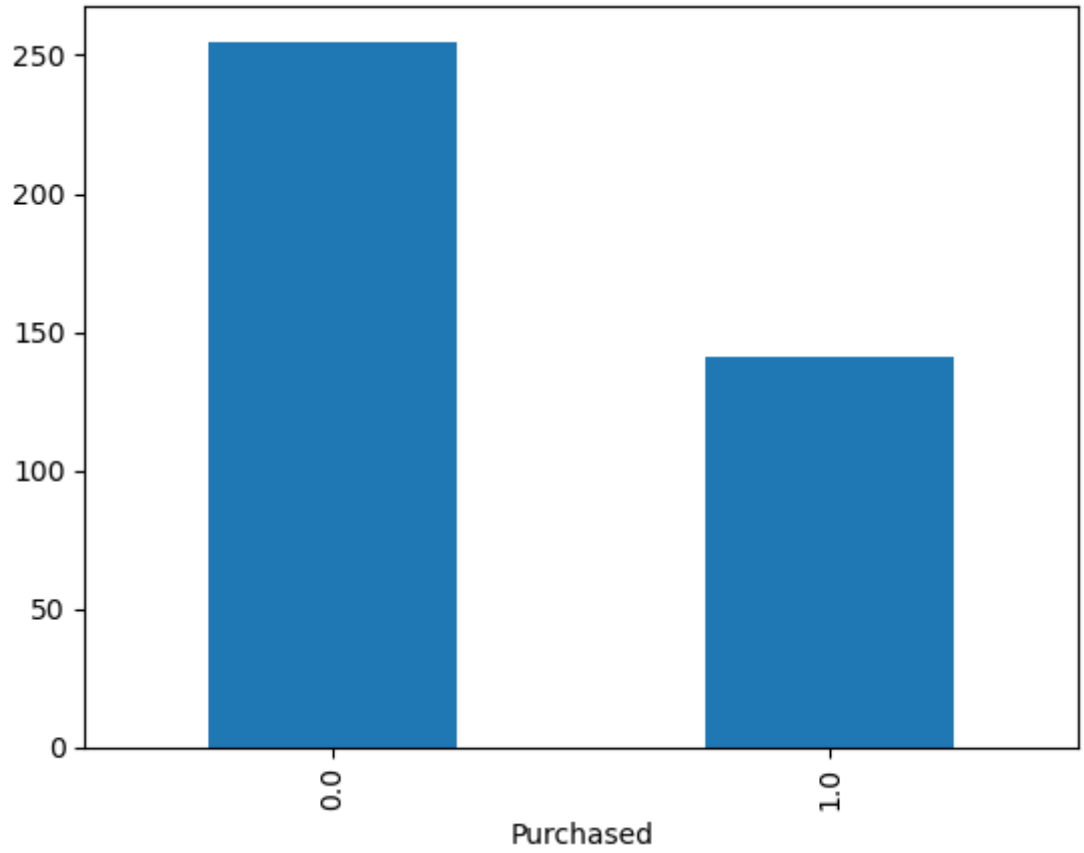
1. Which features (age, estimated salary, etc.) have the most significant impact on purchase decisions?

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Load your dataset
data = pd.read_csv('Social_Network_Ads.csv')
```

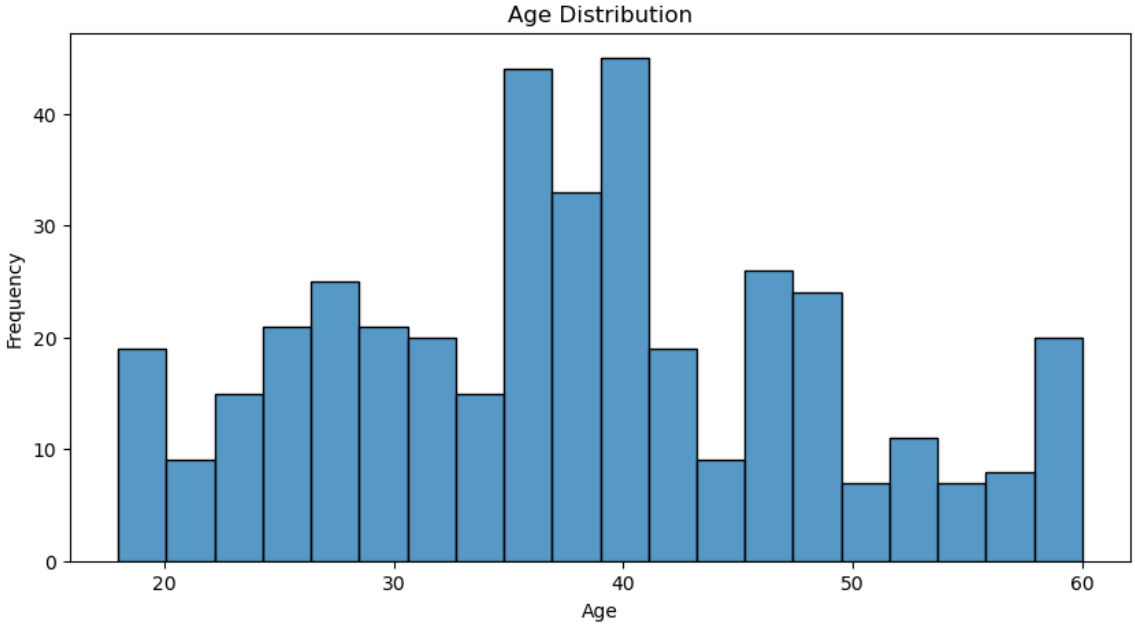
In [12]:

```
data["Purchased"].value_counts().plot(kind="bar");
```



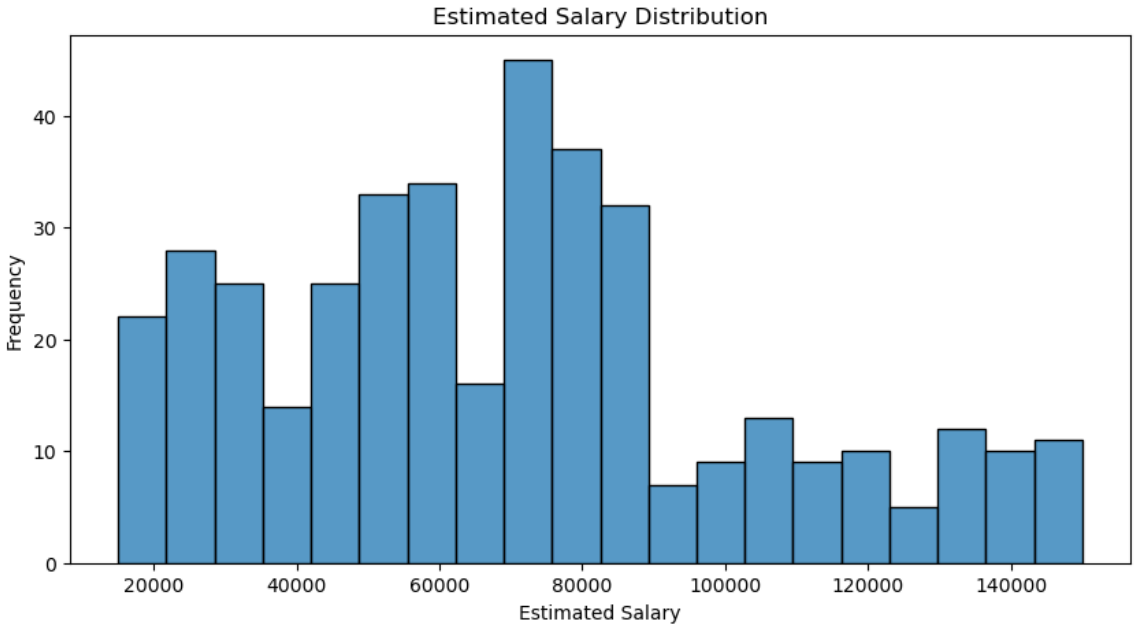
In [3]:

```
# Descriptive Questions:

# What is the age distribution in the dataset?
plt.figure(figsize=(10, 5))
sns.histplot(data['Age'], bins=20)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()

```



In [4]:

```
# What is the distribution of estimated salaries?
plt.figure(figsize=(10, 5))
sns.histplot(data['EstimatedSalary'], bins=20)
plt.title('Estimated Salary Distribution')
plt.xlabel('Estimated Salary')
plt.ylabel('Frequency')
plt.show()
```



In [5]:

```
# How many people made a purchase (Purchased=1) and how many did not (Purchase
purchase_counts = data['Purchased'].value_counts()
print('Purchase Counts:\n', purchase_counts)


```

```
Purchase Counts:
 Purchased
0.0    255
1.0    141
Name: count, dtype: int64
```

In [6]:

```
# What is the overall percentage of people who made a purchase?
percentage_purchase = (purchase_counts[1] / purchase_counts.sum()) * 100
print(f'Percentage of people who made a purchase: {percentage_purchase:.2f}%')

```

```
Percentage of people who made a purchase: 35.61%
```

In [7]:

```
1
2  # What is the average age and estimated salary of those who made a purchase c
3  average_age_purchase = data[data['Purchased'] == 1]['Age'].mean()
4  average_salary_purchase = data[data['Purchased'] == 1]['EstimatedSalary'].mean
5  average_age_no_purchase = data[data['Purchased'] == 0]['Age'].mean()
6  average_salary_no_purchase = data[data['Purchased'] == 0]['EstimatedSalary'].m
7
8  print('Average Age of Purchase:', average_age_purchase)
9  print('Average Estimated Salary of Purchase:', average_salary_purchase)
10 print('Average Age of No Purchase:', average_age_no_purchase)
11 print('Average Estimated Salary of No Purchase:', average_salary_no_purchase)
12
```

```
Average Age of Purchase: 46.51063829787234
Average Estimated Salary of Purchase: 86338.12949640288
Average Age of No Purchase: 32.77865612648221
Average Estimated Salary of No Purchase: 60444.88188976378
```

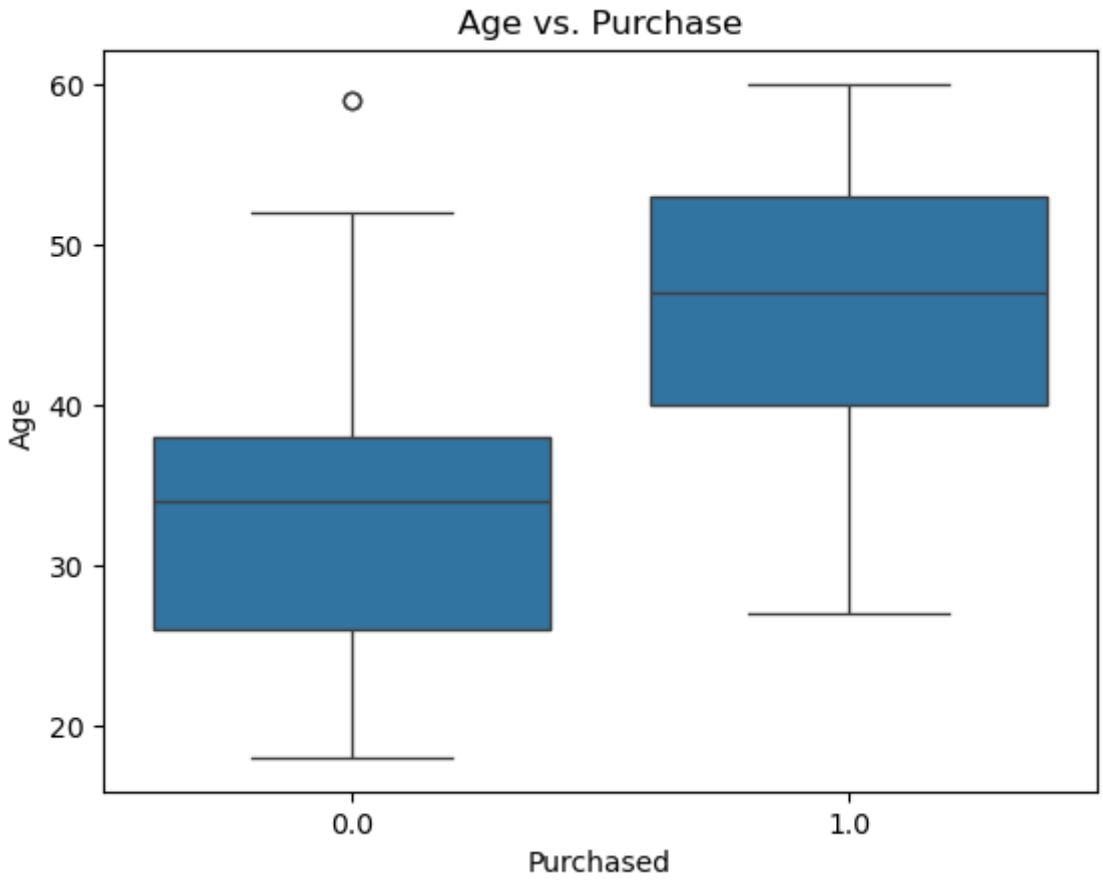
In [8]:

```
1  # Are there any missing values in the dataset, and if so, in which columns?
2  missing_values = data.isnull().sum()
3  print('Missing Values:\n', missing_values)
4
5
```

```
Missing Values:
 Age               2
EstimatedSalary    3
Purchased          4
dtype: int64
```

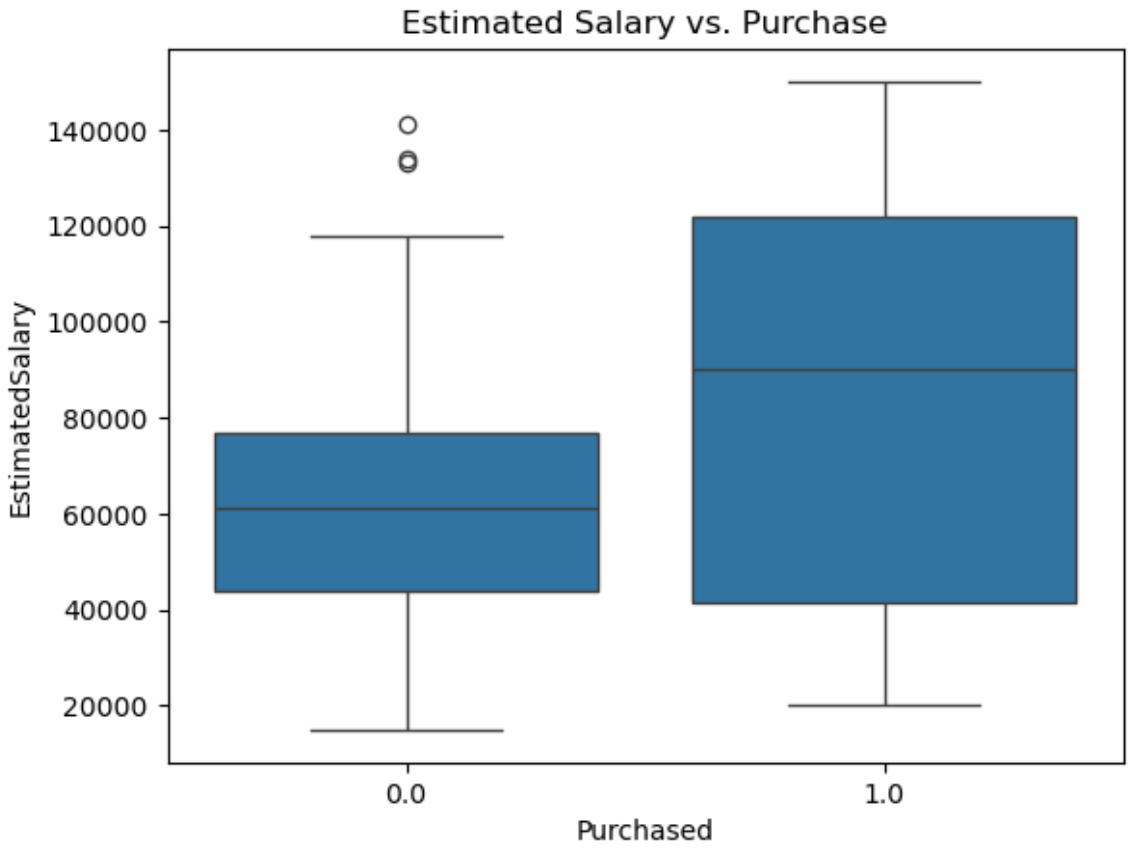
In [9]:

```
# Exploratory Questions:

# Is there a correlation between age and the likelihood of making a purchase?
sns.boxplot(x='Purchased', y='Age', data=data)
plt.title('Age vs. Purchase')
plt.show()


```



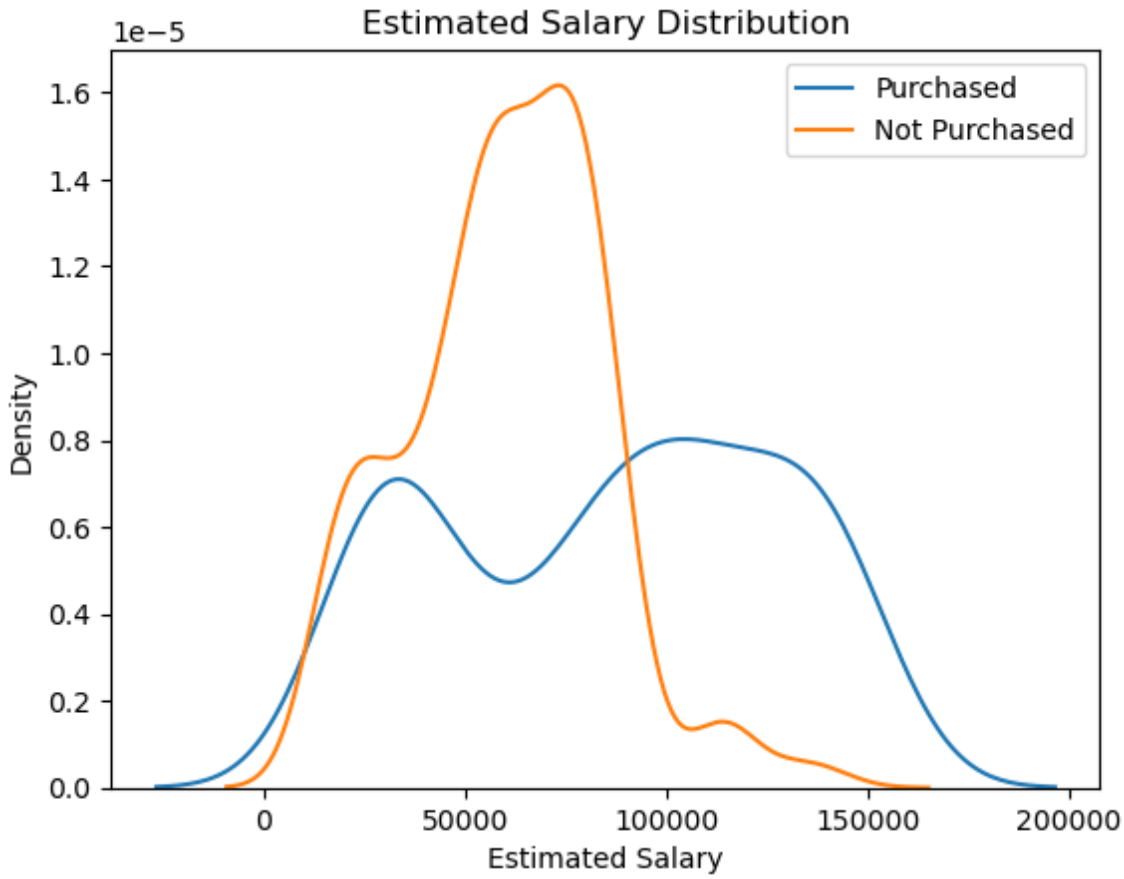
In [10]:

```
# Is there a correlation between estimated salary and the likelihood of making
sns.boxplot(x='Purchased', y='EstimatedSalary', data=data)
plt.title('Estimated Salary vs. Purchase')
plt.show()


```



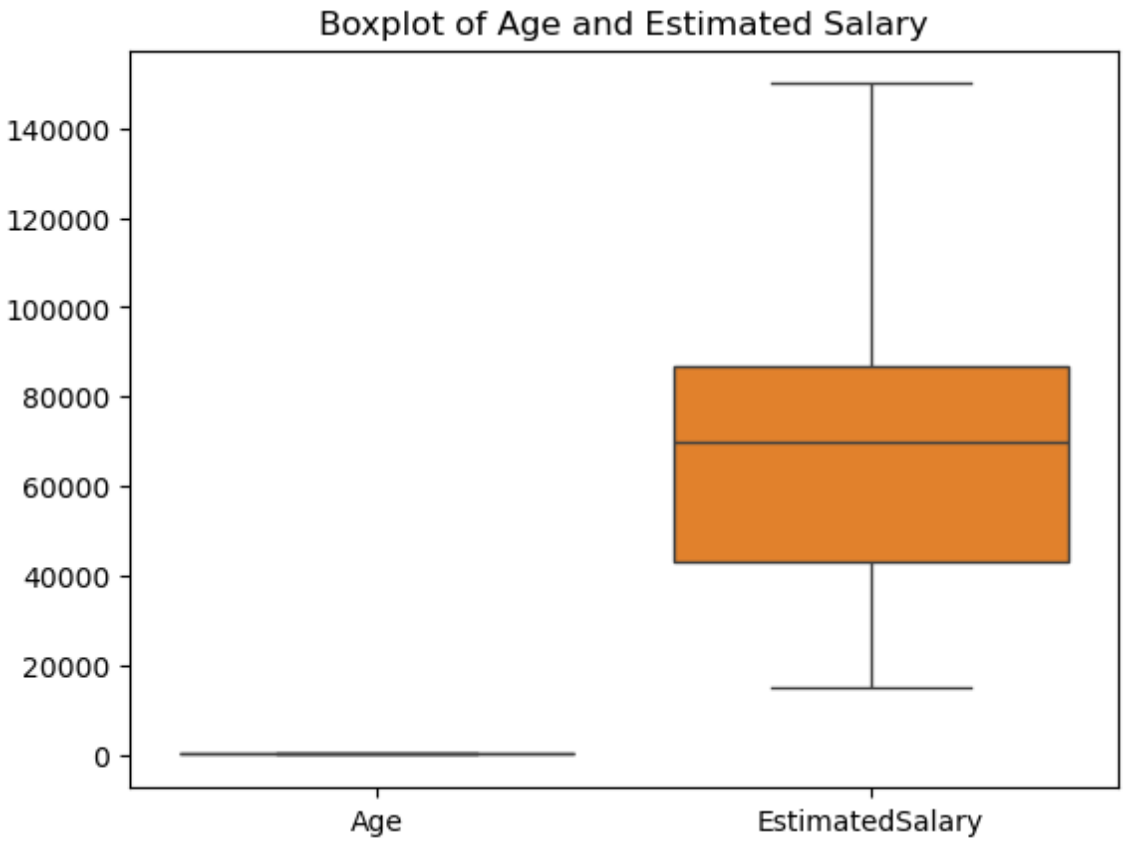
In [11]:

```
# How does the distribution of estimated salary differ between those who made
sns.kdeplot(data[data['Purchased'] == 1]['EstimatedSalary'], label='Purchased
sns.kdeplot(data[data['Purchased'] == 0]['EstimatedSalary'], label='Not Purcha
plt.title('Estimated Salary Distribution')
plt.xlabel('Estimated Salary')
plt.ylabel('Density')
plt.legend()
plt.show()


```



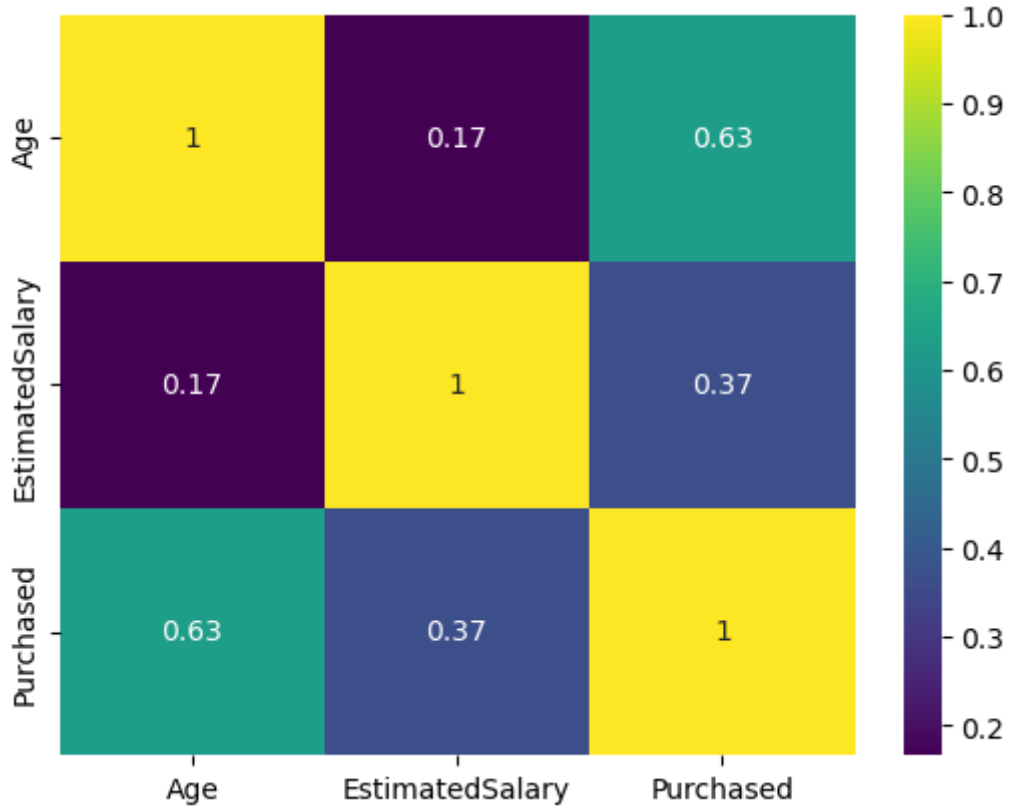
In [12]:

```
# Are there any outliers in the dataset, and how do they affect the analysis?
sns.boxplot(data=data[['Age', 'EstimatedSalary']])
plt.title('Boxplot of Age and Estimated Salary')
plt.show()
```



In [13]:

```
1 sns.heatmap(data.corr(), annot = True, cmap='viridis');
```



- the Age appears to have more impart in the purchased column when compared to Estimated salary with 63% positive relationship/correlations

In [ ]:

```
1
```