📌 Case Study 1: Student Performance Prediction (Linear Regression)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load dataset
df = pd.read_csv("student_performance.csv")

# Check missing values
print(df.isnull().sum())

# Fill missing values
df = df.fillna(df.mean(numeric_only=True))

# Drop a column
df = df.drop(columns=['school'])

# One-hot encoding
df = pd.get_dummies(df, drop_first=True)

# NumPy functions
print("Mean:", np.mean(df['G3']))
print("Std:", np.std(df['G3']))
print("Shape:", np.shape(df))

# Heatmap
```

```
sns.heatmap(df.corr(), cmap='coolwarm')
plt.show()

# Features and target
X = df.drop('G3', axis=1)
y = df['G3']

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

# Model training
model = LinearRegression()
model.fit(X_train, y_train)

# Testing
y_pred = model.predict(X_test)
print("MSE:", mean_squared_error(y_test, y_pred))
print("R2 Score:", r2_score(y_test, y_pred))

# Visualization
plt.scatter(X_test.iloc[:, 0], y_test)
plt.plot(X_test.iloc[:, 0], y_pred, color='red')
plt.show()
```

📌 Case Study 2: House Price Prediction (Multiple Linear Regression)

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error

df = pd.read_csv("house_prices.csv")

# Select numeric columns
df = df.select_dtypes(include=['int64', 'float64'])

# Drop NA values
df = df.dropna()

# Correlation
print(df.corr())

# Describe dataset
print(df.describe())

# NumPy functions
df['Price_log'] = np.log(df['Price'])
df['Area_sqrt'] = np.sqrt(df['Area'])
df['Flag'] = np.where(df['Price'] > df['Price'].mean(), 1, 0)

# Pairplot
sns.pairplot(df)
plt.show()

# Histogram
plt.hist(df['Price'])
plt.show()

X = df.drop('Price', axis=1)
y = df['Price']

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

```
model = LinearRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("MSE:", mean_squared_error(y_test, y_pred))
```

📌 Case Study 3: Iris Flower Classification (KNN)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score

df = pd.read_csv("iris.csv")

# Head
print(df.head())

# Value counts
print(df['species'].value_counts())

# Drop column
df = df.drop(columns=['Id'])

# NumPy functions
print("Unique species:", np.unique(df['species']))
print("Argmax example:", np.argmax([5, 20, 10]))

# Scatterplot
sns.scatterplot(x='sepal_length', y='sepal_width', hue='species', data=df)
plt.legend()
plt.show()

X = df.drop('species', axis=1)
y = df['species']

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

model = KNeighborsClassifier(n_neighbors=5)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

📌 Case Study 4: Customer Churn Prediction (Decision Tree)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

df = pd.read_csv("telecom_churn.csv")

df = df.replace(" ", np.nan)
df['Churn'] = df['Churn'].map({'Yes': 1, 'No': 0})
df = pd.get_dummies(df, drop_first=True)
df = df.drop_duplicates()

# NumPy functions
print("Non-zero churn:", np.count_nonzero(df['Churn']))
```

```
df = np.nan_to_num(df)

# Countplot
sns.countplot(x=df[:, -1])
plt.show()

# Bar plot
plt.bar(['No Churn', 'Churn'], [300, 100])
plt.show()

X = df[:, :-1]
y = df[:, -1]

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

model = DecisionTreeClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

📌 Case Study 5: Credit Card Fraud Detection (Logistic Regression)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report

df = pd.read_csv("creditcard.csv")
```

```
df_sample = df.sample(5000)

summary = df_sample.groupby('Class').agg({'Amount': 'mean'})
print(summary)

print("95th percentile:", np.percentile(df_sample['Amount'], 95))
df_sample['Amount'] = np.clip(df_sample['Amount'], 0, 500)

sns.boxplot(x='Class', y='Amount', data=df_sample)
plt.show()

plt.scatter(df_sample['Time'], df_sample['Amount'])
plt.show()

X = df_sample.drop('Class', axis=1)
y = df_sample['Class']

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

model = LogisticRegression(max_iter=1000)
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print(classification_report(y_test, y_pred))
```

📌 Case Study 6: Sales Forecasting (Linear Regression)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

df = pd.read_csv("sales.csv")

df['Date'] = pd.to_datetime(df['Date'])
monthly = df.resample('M', on='Date').sum()

monthly['Rolling'] = monthly['Sales'].rolling(3).mean()
monthly['Lag'] = monthly['Sales'].shift(1)

print(np.diff(monthly['Sales'].dropna()))
print(np.cumsum(monthly['Sales'].fillna(0)))

plt.plot(monthly['Sales'])
sns.lineplot(x=monthly.index, y=monthly['Sales'])
plt.show()

monthly = monthly.dropna()
X = monthly[['Lag']]
y = monthly['Sales']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = LinearRegression()
model.fit(X_train, y_train)
print("Test R2:", model.score(X_test, y_test))
```

📌 Case Study 7: Spam Email Detection (Naive Bayes)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

df = pd.read_csv("spam.csv")

df['text'] = df['text'].apply(lambda x: x.lower())
df['text'] = df['text'].str.replace('[^a-z ]', '', regex=True)

arr = np.zeros(5)
print("Sum of zeros:", np.sum(arr))

sns.barplot(x=['spam', 'ham'], y=df['label'].value_counts().values)
plt.show()

plt.bar(['spam', 'ham'], df['label'].value_counts().values)
plt.show()

X = df['text']
y = df['label']

vec = CountVectorizer()
X_vec = vec.fit_transform(X)

X_train, X_test, y_train, y_test = train_test_split(X_vec, y, test_size=0.2)

model = MultinomialNB()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

📌 Case Study 8: Employee Attrition Prediction (Logistic Regression)

```python
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

df = pd.read_csv("hr.csv")

df = df.dropna()
df = df.rename(columns={'Attrition': 'attrition'})
print(df['attrition'].value_counts(normalize=True))

print("Rounded Mean Income:", np.round(df['MonthlyIncome'].mean()))
print("Mean Age:", np.mean(df['Age']))

sns.violinplot(x='attrition', y='Age', data=df)
plt.show()

df['attrition'].value_counts().plot.pie(autopct='%1.1f%%')
plt.show()

X = pd.get_dummies(df.drop('attrition', axis=1), drop_first=True)
y = df['attrition']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
model = LogisticRegression(max_iter=2000)
```

```
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```

📌 Case Study 9: Movie Recommendation System (KNN)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.neighbors import NearestNeighbors

df = pd.read_csv("ratings.csv")

df = df.dropna()
df = df.drop_duplicates()

movie_matrix = df.pivot(index='userId', columns='movieId', values='rating')
matrix_np = movie_matrix.fillna(0).to_numpy()

sns.boxplot(data=df['rating'])
plt.show()

plt.bar(df['movieId'].value_counts().head(5).index, df['movieId'].value_counts().head(5).values)
plt.show()

model = NearestNeighbors(metric='cosine')
model.fit(matrix_np)
```

📌 Case Study 10: Diabetes Prediction (SVM)

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score

df = pd.read_csv("diabetes.csv")

df = df.replace(0, np.nan)
df = df.fillna(df.median())

print("Median Glucose (NumPy):", np.median(df['Glucose']))

sns.boxplot(data=df)
plt.show()

plt.hist(df['Glucose'])
plt.show()

X = df.drop('Outcome', axis=1)
y = df['Outcome']

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

model = SVC()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Accuracy:", accuracy_score(y_test, y_pred))
```