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

# Load the dataset
dataset = pd.read_csv("Documents/heart.csv")

# Data exploration
dataset.info()
print("Size of dataset is:", dataset.size)
print("Shape of dataset is:", dataset.shape)
print(dataset.describe())
print(dataset.head())

# Data type of each column
print("Data Type for Each Column:\n", dataset.dtypes.value_counts())

# Display missing values
print("Missing Values:\n", dataset[dataset.columns[dataset.dtypes != 'object']].isnull())

# Finding mean age of Patients
average = s.mean(dataset['age'])
print("Average age:", average)
```

```
# Extracting Particular Columns
selected_columns = ['age', 'sex', 'trestbps', 'chol']
print(dataset[selected_columns])

# Displaying Confusion Matrix
corr_matrix = dataset.corr()
sns.heatmap(corr_matrix, annot=True, linewidths=0.5, fmt=".2f", cmap="YlGnBu");

# Splitting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(dataset.drop('target', axis=1), dataset.target,
test_size=0.2, random_state=56)

# Classification using Logistic Regression
lr_clf = LogisticRegression(solver='liblinear')
lr_clf.fit(X_train, y_train)

# Displaying Training and Testing Accuracy for Logistic Regression
test_score = accuracy_score(y_test, lr_clf.predict(X_test)) * 100
train_score = accuracy_score(y_train, lr_clf.predict(X_train)) * 100

results_df = pd.DataFrame(data=[["Logistic Regression", train_score, test_score]],
                columns=['Model', 'Training Accuracy %', 'Testing Accuracy %'])
results_df
```