Streamlit: A Framework for Building Data Applications

Streamlit is an open-source app framework for Machine Learning and Data Science projects. It allows you to create beautiful web applications for your machine learning and data science projects with simple Python scripts. Streamlit is designed to make it easy to build and share data applications, enabling data scientists and machine learning engineers to showcase their work without needing extensive web development skills.

1. Installation

To install Streamlit, you can use pip. Run the following command in your terminal:

```
pip install streamlit
```

2. Creating a Simple App

Here is a simple example of a Streamlit application that displays a title and a dataframe:

```
import streamlit as st
import pandas as pd
import numpy as np

# Title of the application
st.title('Hello Streamlit')

# Display a Simple text
st.write('This is a simple text')

# Create a simple dataframe
df = pd.DataFrame({
    'First Column': [1, 2, 3, 4],
    'Second Column': [10, 20, 30, 40]
})

# Display the dataframe
st.write('Here is the dataframe')
st.write(df)

# Create a line chart
chart_data = pd.DataFrame(
    np.random.randn(20, 3), columns=['a','b','c']
)

st.line_chart(chart_data)
```

In this example, we import Streamlit and pandas, create a simple dataframe, and display it along with a line chart. The line chart visualizes random data, demonstrating how easily Streamlit integrates data visualization into applications.

3. User Input

Streamlit allows you to capture user input easily. Here is an example:

```
import streamlit as st
import pandas as pd

st.title('Streamlit Text Input')

name = st.text_input('Enter Your Name:')
if name:
    st.write(f'Hello, {name}')

age = st.slider('Select Your Age:', 0, 100, 25)
st.write(f'Your age is {age}.')

options = ['Python', 'Java', 'C++', 'JavaScript']
choice = st.selectbox('Choose Your Favourite Language:', options)
st.write(f'You Selected {choice}')

data = {
    'Name': ['John', 'Jane', 'Jake', 'Jill'],
    'Age': [28, 24, 35, 40],
    'City': ['New York', 'Los Angeles', 'Chicago', 'Houston']
}

df = pd.DataFrame(data)
df.to_csv('sampledata.csv')
st.write(df)

uploaded_file = st.file_uploader('Choose a CSV File', type='csv')

if uploaded_file is not None:
    df = pd.read_csv(uploaded_file)
    st.write(df)
```

This code snippet captures user input for name, age, and favorite programming language, displays a dataframe, and allows file uploads. Streamlit's interactive widgets make it easy to create dynamic applications that respond to user input.

4. File Upload

You can also allow users to upload files. Here is how you can do it:

```
uploaded_file = st.file_uploader('Choose a CSV File', type='csv')
if uploaded_file is not None:
    df = pd.read_csv(uploaded_file)
    st.write(df)
```

This snippet allows users to upload a CSV file and displays its contents. File upload functionality is essential for applications that require user data input, making Streamlit versatile for various use cases.

5. Customization and Theming

Streamlit provides options for customizing the appearance of your application. You can change the layout, colors, and even add custom CSS styles. For example, you can set the page title and icon using the following commands:

```
st.set_page_config(page_title='My Streamlit App', page_icon=':shark:', layout='wide')
```

This command sets the title and icon of the page and allows for a wide layout, enhancing the user experience.

Conclusion

Streamlit is a powerful tool for building interactive data applications with minimal effort. With its simple API, you can create complex applications that can visualize data, capture user input, and much more. Its ease of use and flexibility make it an excellent choice for data scientists and developers alike.

References

For more information, visit the official Streamlit documentation: Streamlit Documentation