**AgGrid**

**Streamlit library in python used for wht all other alternate in market and give 5 different code example of different uses in streamlit**

**Streamlit** is a popular Python library for building interactive and data-driven web applications with minimal code. It’s widely used in data science for creating dashboards, visualizations, and prototypes due to its simplicity and the ability to create rich UIs without needing HTML or JavaScript. Here’s a look at **alternatives to Streamlit** and **five Streamlit code examples** showcasing different uses.

**Alternatives to Streamlit**

1. **Dash (by Plotly)**: A powerful library for building web applications with Python. It’s well-suited for complex dashboards and supports a wide variety of components, though it has a steeper learning curve than Streamlit.
2. **Panel (by HoloViz)**: Used for creating interactive dashboards and data applications with more flexibility and support for multiple libraries like Matplotlib, Bokeh, and Plotly.
3. **Voila**: Converts Jupyter notebooks into interactive dashboards. It’s great for people who want to leverage Jupyter while adding interactivity.
4. **Flask** (with frontend frameworks): While Flask isn’t a direct competitor, it’s a robust option for developers who want full control over their app’s frontend and backend. Flask can be combined with tools like React, Vue, or vanilla HTML/CSS/JavaScript for customized interfaces.
5. **Gradio**: Designed for quickly building simple machine learning model interfaces. Gradio provides a drag-and-drop interface but is more focused on ML models compared to Streamlit.

**Streamlit Code Examples**

**Example 1: Displaying Data with a Simple Data Table**

Streamlit can be used to display data tables directly from a Pandas DataFrame.

python

Copy code

import streamlit as st

import pandas as pd

# Create a sample dataframe

data = {'Name': ['Alice', 'Bob', 'Charlie'], 'Score': [85, 92, 78]}

df = pd.DataFrame(data)

st.title("Simple Data Table Example")

st.write("This is an example of displaying a data table.")

st.dataframe(df)

**Example 2: Creating a Line Chart**

Streamlit makes it easy to plot data using its built-in charting functions.

python

Copy code

import streamlit as st

import pandas as pd

import numpy as np

# Generate sample data

data = pd.DataFrame({

'X': np.linspace(1, 100, 100),

'Y': np.random.randn(100).cumsum()

})

st.title("Line Chart Example")

st.line\_chart(data)

**Example 3: Interactive Widgets - Slider and User Input**

Streamlit’s widgets like sliders and input fields allow for interactivity in applications.

python

Copy code

import streamlit as st

st.title("Interactive Widgets Example")

# Get user input

age = st.slider("Select your age:", 1, 100)

name = st.text\_input("Enter your name:")

# Display result based on input

if st.button("Submit"):

st.write(f"Hello, {name}! You are {age} years old.")

**Example 4: File Upload and Image Display**

Streamlit allows users to upload files directly to the app, such as images, CSVs, or other data files.

python

Copy code

import streamlit as st

from PIL import Image

st.title("File Upload Example")

uploaded\_file = st.file\_uploader("Choose an image file", type=["jpg", "png"])

if uploaded\_file:

image = Image.open(uploaded\_file)

st.image(image, caption="Uploaded Image", use\_column\_width=True)

**Example 5: Building a Machine Learning Model Interface**

Create a simple interface to input data, pass it to a machine learning model, and display the result.

python

Copy code

import streamlit as st

import numpy as np

from sklearn.ensemble import RandomForestClassifier

from sklearn.datasets import load\_iris

st.title("Iris Flower Prediction App")

# Load Iris dataset and model

iris = load\_iris()

clf = RandomForestClassifier()

clf.fit(iris.data, iris.target)

# User inputs for the model

sepal\_length = st.slider("Sepal length", float(iris.data[:, 0].min()), float(iris.data[:, 0].max()))

sepal\_width = st.slider("Sepal width", float(iris.data[:, 1].min()), float(iris.data[:, 1].max()))

petal\_length = st.slider("Petal length", float(iris.data[:, 2].min()), float(iris.data[:, 2].max()))

petal\_width = st.slider("Petal width", float(iris.data[:, 3].min()), float(iris.data[:, 3].max()))

# Prediction

input\_data = np.array([[sepal\_length, sepal\_width, petal\_length, petal\_width]])

prediction = clf.predict(input\_data)

st.write("Predicted class:", iris.target\_names[prediction[0]])

These examples demonstrate how versatile Streamlit is, covering data display, visualization, interactivity, file handling, and even simple machine learning model interfaces.