Customer Churn MLOps User Interface

**Graphical user interface, application

Description automatically generated**

Follow this guide to create a user interface as show above for the customer churn MLOps (machine learning applications)

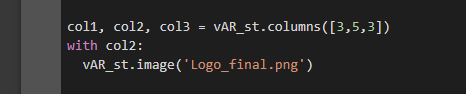
**STREAMLIT:**

Streamlit is a Framework for Machine Learning, Data Science and it is open source, most of the Developers use python for ML and Data science and one of the libraries in python used for this technology is Streamlit. Beautiful UIs can easily be designed through numerous components from the library.

**EXERCISE 1:** Display media

St.columns helps in Side-by-side columns where you can insert Streamlit elements like images or text or any type of elements. We can also use st.columns for aligning the elements in a specified manner.

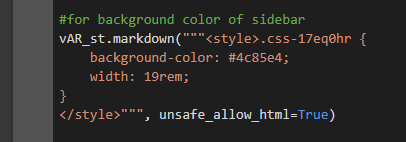
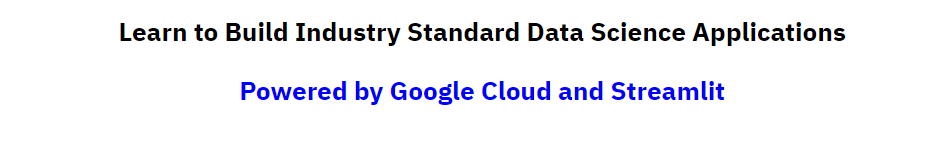
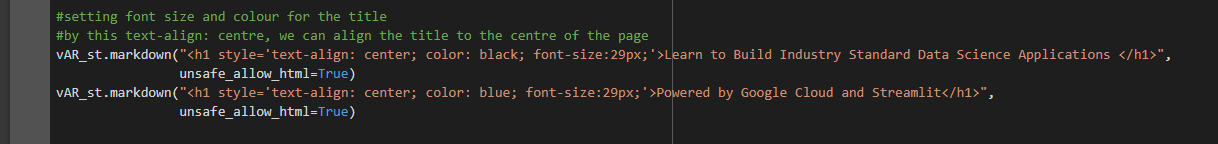
We can display images, audio and video files by simple functions like st.image(), st.audio(), st.video()respectively. We can also set a caption and width to the image.



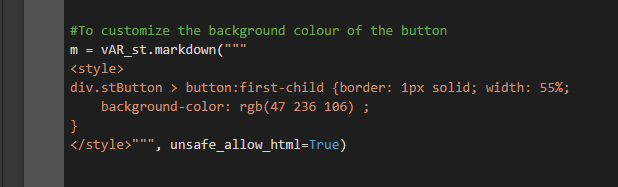


**EXERCISE 2:** st.markdown

Markdown has several uses within a streamlit application. As the only tool for custom HTML within a streamlit app, you can use it to flexibly insert rich content into your application like changing the color for a button, changing background colors for sidebar or you can make custom titles for streamlit app by changing the font size and color.

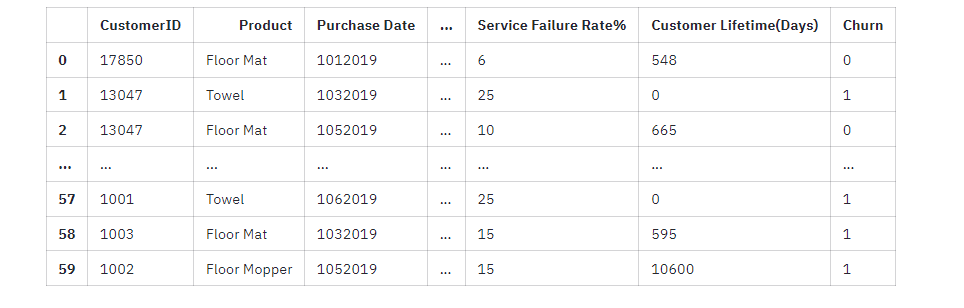






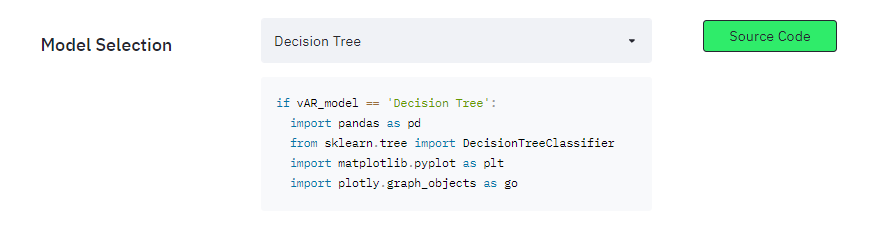
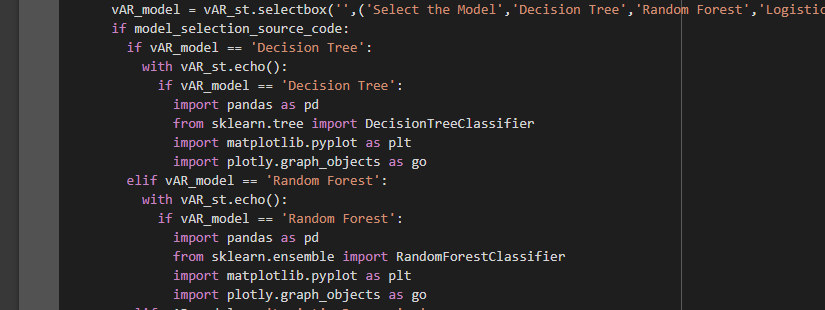
**EXERCISE 3:** Display text/tables

This is a Streamlit command which does different things depending on what you specify. Unlike other Streamlit commands, write() has some unique properties: You can pass in multiple arguments like write(data\_frame), write(function), write(dict), write(plotly\_fig), all of which will be written. Or simply we can use it to display a text when needed.



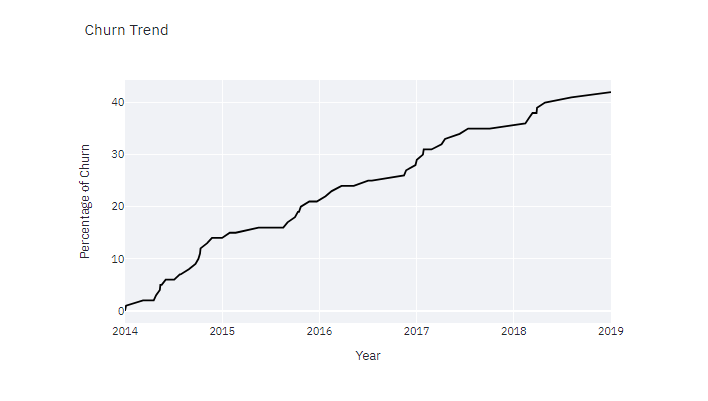
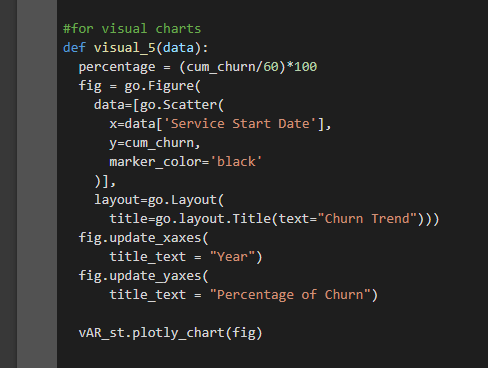
**EXERCISE 4:** Display code

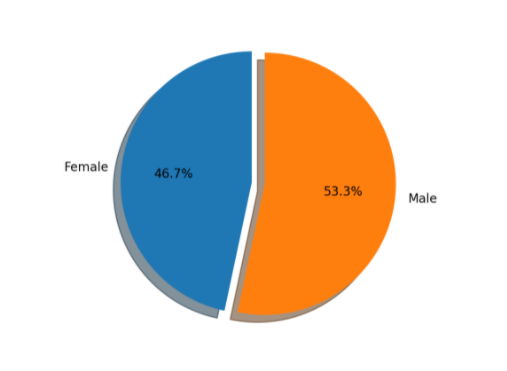
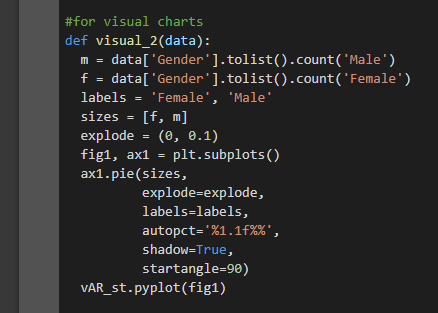
With the help of st.echo() we can make the middle section of the code visible.Whatever code is present inside this function will be visible on the screen, and it will also be executed.



**EXERCISE 4:** Display charts

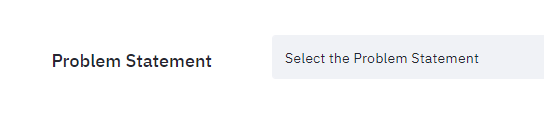
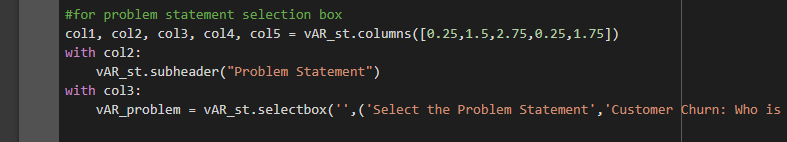
Plotly is a charting library for Python, Display a matplotlib.pyplot figure. The arguments to this function closely follow the ones for Plotly’s plot() function.





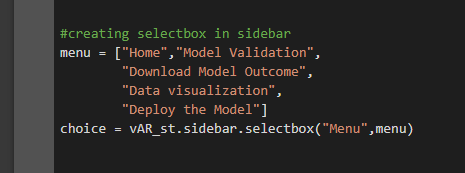
**EXERCISE 5:** Text formats

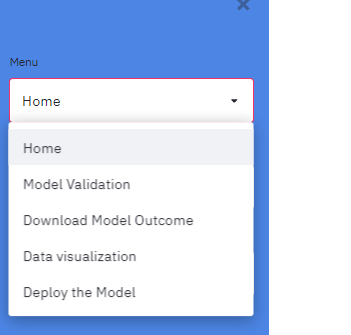
Streamlit apps usually start with a call to st.title to set the app’s title. After that, there are 2 heading levels you can use: st.header and st.subheader. Pure text is entered with st.text, and Markdown with st.markdown.



**EXERCISE 6:** Sidebar and selectbox

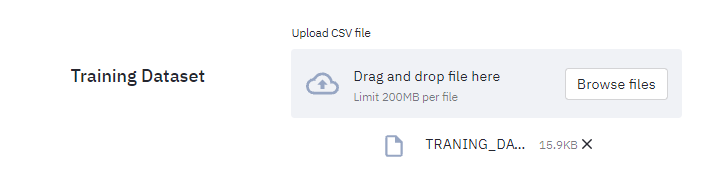
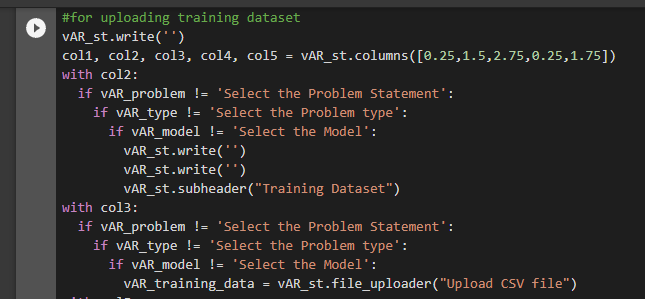
We can add interactivity to our report with widgets, we can organize them into a sidebar with st.sidebar.selectbox. Each element that’s passed to st.sidebar is pinned to the left, allowing users to focus on the content in your app. The only elements that aren’t supported are st.echo and st.spinner.





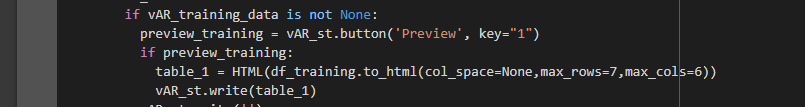
**EXERCISE 7:** File Uploader

Upload files to streamlit by the function st.file\_uploader()



**EXERCISE 8:** Action button

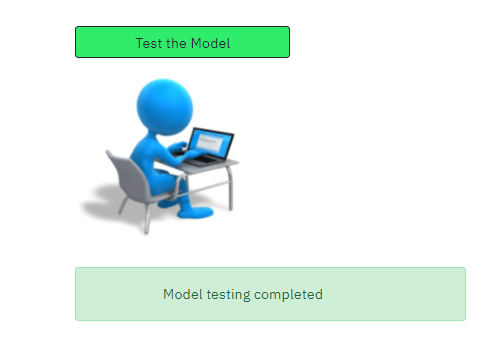
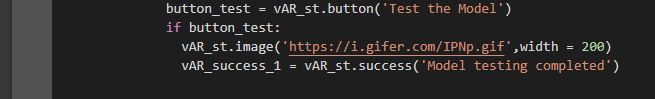
We can add a action button to perform some actions when clicked



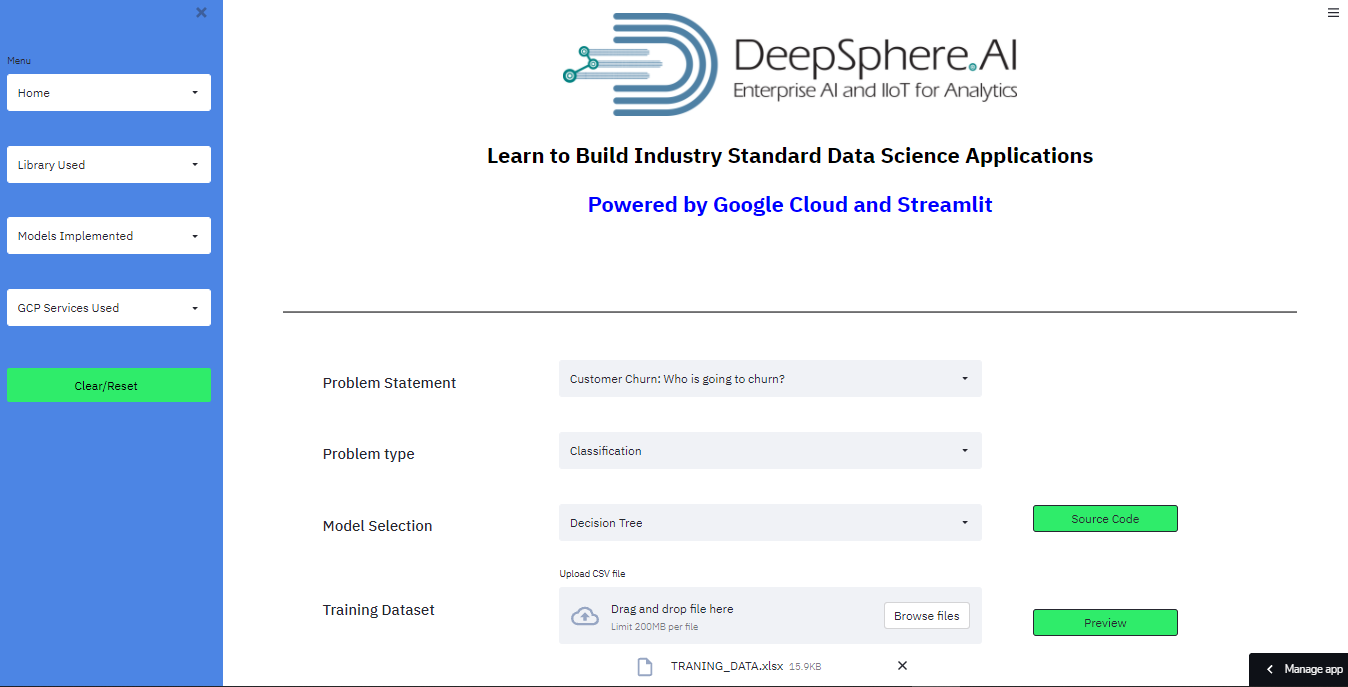


**EXERCISE 8:** Display status

Display a success message after the execution.



**CHURN PREDICTION DATA SCIENCE APPLICATION USING STREAMLIT**

****