KDnuggets

Visit KDnuggets.com for more cheatsheets and additional learning resources.

Streamlit CheatSheet

Streamlit is an open-source Python-based web framework. Turn your machine-learning model into a web app and deploy it without worrying about the frontend and backend.

write

Display pandas DataFrames, errors, dictionaries, interactive visualization, and Keras models.

```
st.write('Welcome to KDnuggets.')
```

markdown

Use markdown to display images, text, headings, and everything else.

```
st.markdown('KDnuggets is **_really_ cool**.')
```

metric

Display displaying KPIs and model performance metrics.

```
st.metric("Accuracy", "90%", "+2.5")
```

chart

Display a line, bar, and area chart.

```
st.line_chart(df)
st.bar_chart(df)
st.area_chart(df)
```

map

Display a map with points on it.

```
st.map(df)
```

plotly_chart

Display an interactive Plotly chart.

```
st.plotly_chart(fig, use_container_width=True)
```

button

Add the button widget.

```
if st.button('Predict'):
   st.write(result)
```

selectbox

Add dropdown selectbox widget.

```
option = st.selectbox('Category',('Car', 'Bus', 'Train'))
st.write('You selected:', option)
```

multiselect

Add multiselect option widget.

```
options = st.multiselect('Category',['Car', 'Bus',
'Train', 'Airplane'],['Car', 'Bus'])
st.write('You selected:', options)
```

input_text

Add single-line text input widget for NLP and ML models.

```
tweet = st.text_input('Paste the tweets', 'Love')
```

file_uploader

Upload any type of file using uploader widget.

```
uploaded_file = st.file_uploader("Choose a file")
```

image

Display an image or list of images.

```
st.image(image, caption='bus ride')
```

audio

Add an audio player.

```
st.audio(audio_bytes, format='mp3')
```

sidebar

Organize your widgets into a collapsible sidebar.

```
with st.sidebar:
   st.[element_name]
```

tabs

Organize your widgets into multiple tabs.

```
tab1, tab2 = st.tabs(["Bus", "Train"])
```

progress

Add progress bar.

```
p_bar = st.progress(0)

for percent in range(100):
   time.sleep(0.1)
   p_bar.progress(percent + 1)
```

cache

A function decorator to memorize function executions.

```
@st.cache(persist=True)
def fetch_and_clean_data(url):
   return data
```

Learn more on kdnuggets.com

Abid Ali Awan | 2023