

# Data Apps Made Easy With



# Streamlit

**Women in Data Science - Mysuru, 26 Sept 2020**

# Before we get started....

---

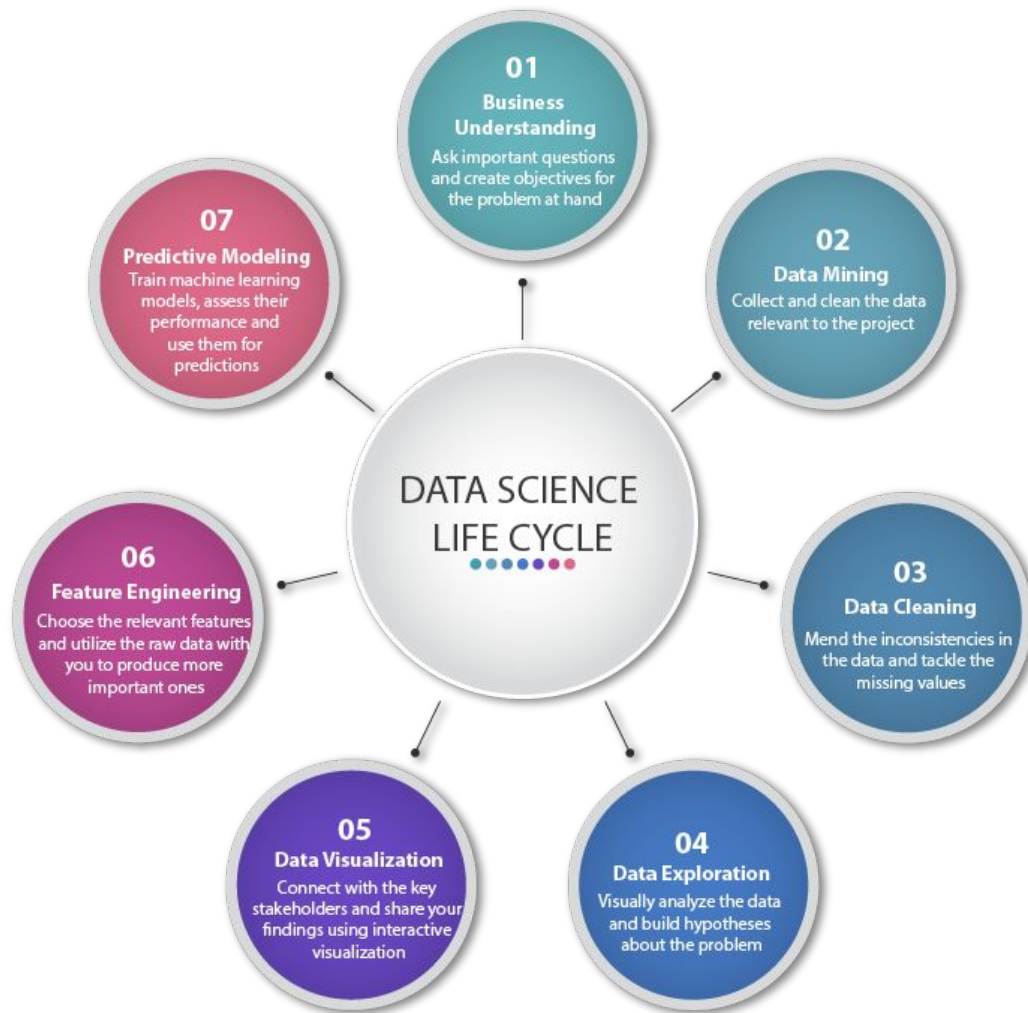
**NIHARIKA KRISHNAN**

“Leveraging AI to help people lead better lives!”

- Machine Learning Engineer, TCS
  - Building NLP based applications for Walmart, US
- Founder & Organizer - PyLadies Chennai
  - [chennai.pyladies.com](http://chennai.pyladies.com)
- Speaker: PyCon Canada'19, India'19, PyOhio'20(v)
- Mentor - Learn IT Girl



**Just another curious ML Engineer.. :D**



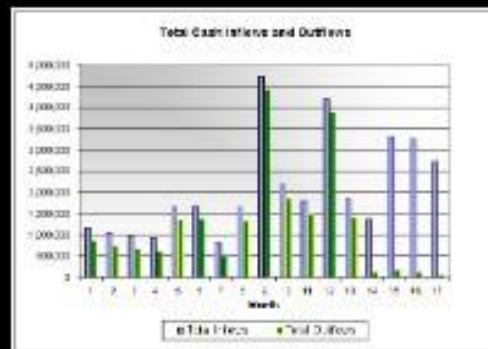


**WHEN DATA IS IN TABLE FORM**

ID	NAME	CLASS	MARK	SEX
1	John Doe	Four	75	female
2	Max Rahn	Three	85	male
3	Arnold	Three	90	male
4	Kish Star	Four	80	female
5	John Mico	Four	60	female
6	Alex John	Four	55	male
7	My John Rob	Fifth	78	male
8	Arnold	Five	85	male
9	Tes Coy	Six	78	male
10	Big John	Four	55	female



**WHEN DATA IS IN PLOT**



**How do you share your insights?**



Deploying it to production? Sharing notebooks?



Requires Web Development Knowledge

Comparatively more complex and time consuming

Although, definitely more customizable and robust



More suitable for Plotly graphs


**Building data apps in Python?**



# Getting started with Streamlit?

— — —

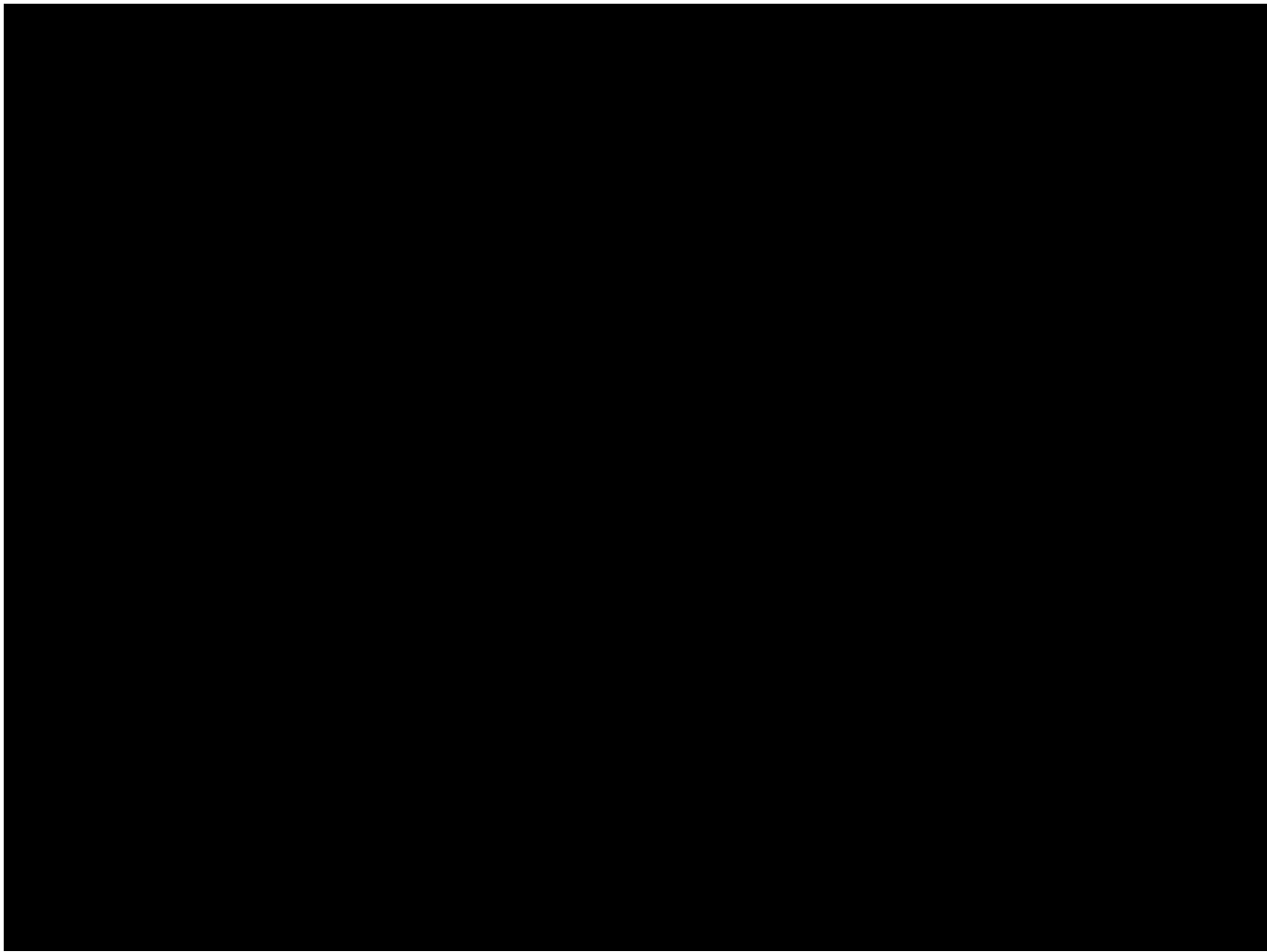
- Lets data scientists create their own applications
- Turn your Python scripts → Web Applications
- Create rich, interactive experiences using widgets
- Compatible with major ML/DL libraries & frameworks
- Fast interactive loop; Visualise → mutate → share



```
pip install streamlit
```

```
#Demo  
streamlit hello
```

```
#Custom Python Script  
streamlit run filename.py
```



# Streamlit - Text

— — —

```
import streamlit as st

st.title("Women in Data Science - Mysuru")
st.write("Reverse Image recommendation system")
st.markdown("# Streamlit Demo")
```

## Women in Data Science - Mysuru

Reverse Image recommendation system

## Streamlit Demo

# Streamlit- Widgets

— — —

```
genre = st.radio(
    "Gender",
    ('Male', 'Female'))

size = st.multiselect(
    'Size', ['Extra Small (XS)', 'Small (S)', 'Medium (M)', 'Large (L)', 'Extra Large (XL)'])

material = st.multiselect('Material', ['Cotton', 'Nylon', 'Syntnthetic', 'Woolen'])

price = st.slider("Price:", value=(0, 3000))
```

Gender

☒ Male

☐ Female

Size

Choose an option ▼

Material

Choose an option ▼

Price:

0 3000

0 3000

# Streamlit - SideBar Widgets

## Women in Data Science - Mysuru Demo

Reverse Image recommendation system



Gender



Male



Female

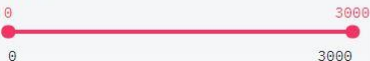
Size

Choose an option

Material

Choose an option

Price:



# Streamlit - Placeholders, Progress

— — —



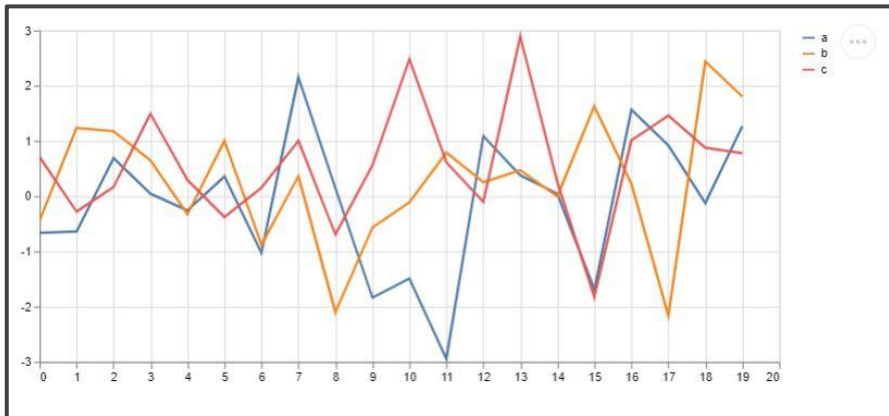
```
placeholder = st.empty()

placeholder.image('cloth1.jpg', width = 300)

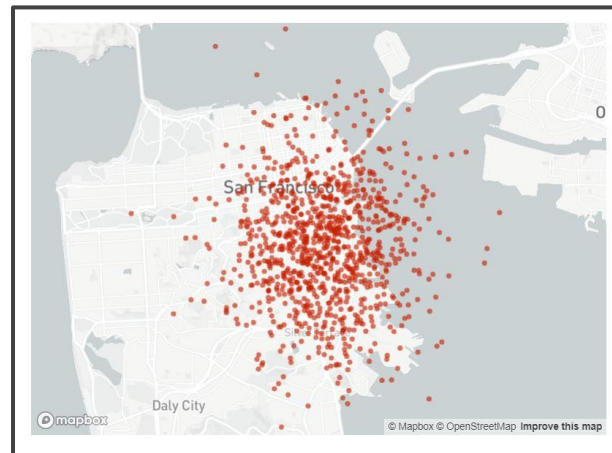
if st.button('GET RECOMMENDATIONS'):
    with st.spinner('Generating recommendations'):
        time.sleep(2)
        st.success('Results:')
```

# Streamlit - Charts

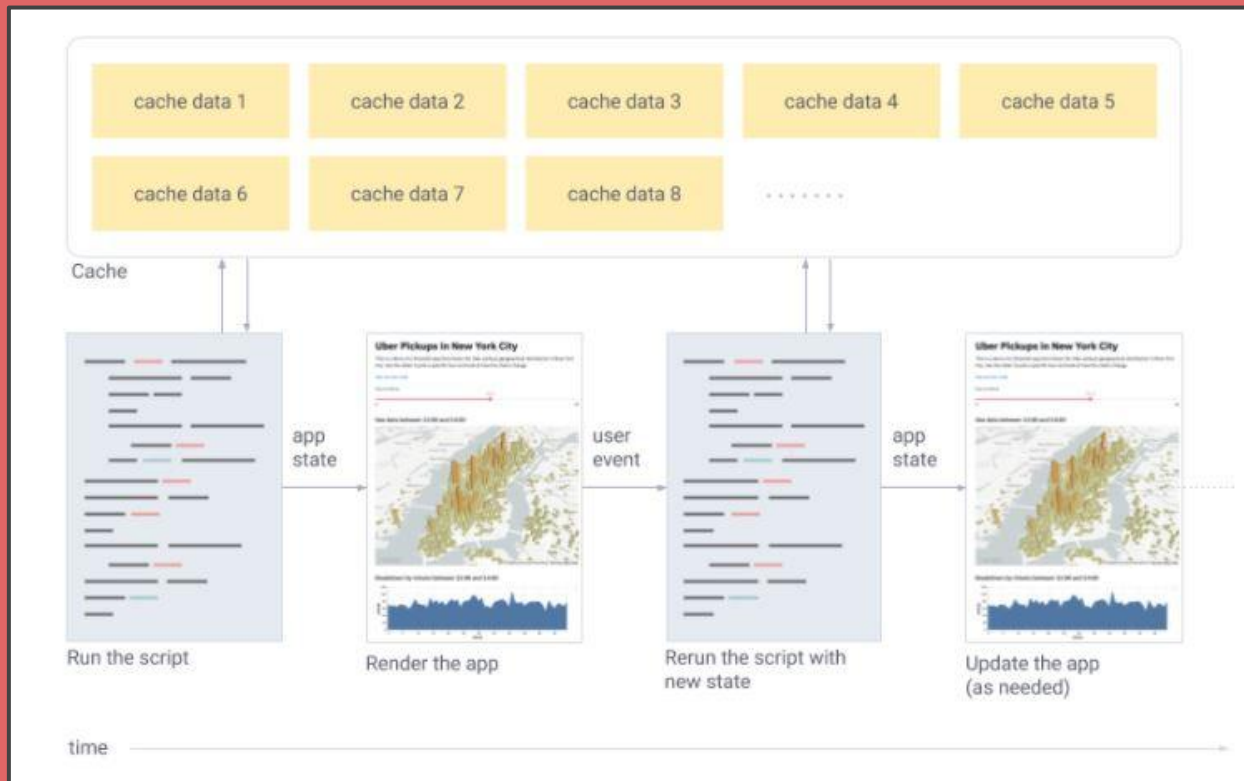
```
chart_data = pd.DataFrame(  
    np.random.randn(20, 3),  
    columns=['a', 'b', 'c'])  
  
st.line_chart(chart_data)
```



```
map_data = pd.DataFrame(  
    np.random.randn(1000, 2) / [50, 50] + [37.76, -122.4],  
    columns=['lat', 'lon'])  
  
st.map(map_data)
```



# Bringing it all together...





# Streamlit - Caching

— — —

- Linear sequence of actions from top to bottom
- Every code change, re-runs the source file but **what if you have expensive computation / manipulating large datasets?**
- `st.cache()` checks for:
  - The input parameters that you called the function with
  - The value of any external variable used in the function
  - The body of the function
  - The body of any function used inside the cached function

```
import streamlit as st
import time

@st.cache #Decorator
def expensive_computation(a, b):
    time.sleep(2)
    return a * b

a = 2
b = 21
res = expensive_computation(a, b)

st.write("Result:", res)
```

# Other Functionalities you can explore...

— — —

**Animations**

**Components**

**Grids**

**ScreenCast**

**Latex**

**File Upload**

**Advanced Caching**

# Summary

— — —

- Streamlit python script draws output live in a browser
- Reduces the learning curve associated with deployment and web-app building
- Open Source - Build data apps within minutes
- **“Go to Hackathon buddy” ~ “Showcase to shareholders” ~ “DIY Projects”**

**Go build your own data app!**

# QUESTIONS

**Want to explore further? Let's connect!**



**[linkedin.com/in/niharikakrishnan](https://www.linkedin.com/in/niharikakrishnan)**



**[@Nihaaarika](https://twitter.com/Nihaaarika)**



**[github.com/niharikakrishnan](https://github.com/niharikakrishnan)**

**Slide Deck: <https://github.com/niharikakrishnan/Talks>**