```
! pip install streamlit -q

          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 9.7/9.7 MB 32.3 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 164.8/164.8 KB 10.8 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.7/4.7 MB 48.1 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 82.1/82.1 KB 5.0 MB/s eta 0:00:00
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 184.3/184.3 KB 11.6 MB/s eta 0:00:00
       Preparing metadata (setup.py) ... done
          ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 62.7/62.7 KB 4.8 MB/s eta 0:00:00
       Building wheel for validators (setup.py) ... done
```

Write the cell python code into an app.py file

```
%%writefile app.py
import streamlit as st
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn import metrics
from xgboost import XGBClassifier
from sklearn.linear_model import SGDClassifier
PAGE_CONFIG = {"page_title":"StColab.io","page_icon":":smiley:","layout":"centered"}


def main():
  st.title("Telcom Churn Prediction")
  st.subheader("Enter the details and predict if the customer will churn or not")
  gender=("Male","Female","Prefer not to say")
  gender = st.selectbox("Gender", gender)
  partner = st.radio("Do you have a partner like husband/wife?",('Yes','No'))
  phone = st.radio("Do you have phone service plan?",('Yes','No'))
  Tenure = st.slider("What is the tenure like in years?", 0, 50, 1)
  charge = st.number_input('What are the total charges you are paying for the company?',step=0.01)
  st.write('The current number is ', charge)
  security= st.radio("Do you have a online security like number protection, etc?",('Yes','No'))
  contract=st.radio("What type of contract you are in?",('Contract_Month-to-month','Contract_One year','Contract_Two year'))
  tech_support=st.radio("Are you satisfied or Is tech support provided?",('Yes','No'))
  payment_meth=st.radio("What type of payment method you bhave used?",('PaymentMethod_Bank transfer','PaymentMethod_Credit card','PaymentMeth
  st.write("the selected are",gender,partner,phone,Tenure,charge,security,contract,tech_support,payment_meth)
  gender_Male=0
  gender_Female=0
  partner_Yes=0
  partner_No=0
  phone_Yes=0
  phone_No=0
  tech_Yes=0
  tech_No=0
  pay_bank=0
  pay_credit=0
  pay_elec=0
  pay_mail=0
  contract_month=0
  contract_one_year=0
  contract_two_year=0
  security_Yes=0
  security_No=0
  tech_No_internet_service=0
  sec_No_internet_service=0

  ok = st.button("Predict Chur")

  if ok:
    #st.write("churn")
    if security=='Yes':
      sec_Yes=1
    if security=='No':
      sec_No=1
    if gender=='Female':
      gender_Female=0
    if gender=='Male':
      gender_Male=1
    if partner=='Yes':
      partner_Yes=1
    if partner=='No':
      partner_No=1
    if tech_support=='Yes':
      tech_Yes=1
    if tech_support=='No':
      tech_No=1
    if contract=='Contract_One year':
      contract_one_year=1
    if contract=='Contract_Two year':
      contract_two_year=1
    if contract=='Contract_Month-to-month':
      contract_month=1
    if payment_meth=='PaymentMethod_Bank transfer':
      pay_bank=1
    if payment_meth=='PaymentMethod_Credit card':
      pay_credit=1
    if payment_meth=='PaymentMethod_Electronic check':
      pay_elec=1
    if payment_meth=='PaymentMethod_Mailed check':
      pay_mail=1
    y=[[Tenure,gender_Female,gender_Male,partner_No,partner_Yes,phone_No,phone_Yes,tech_No,tech_No_internet_service,tech_Yes,contract_month,c
    df = pd.DataFrame(y, columns=['Tenure','gender_Female','gender_Male','partner_No','partner_Yes','phone_No','phone_Yes','tech_No','tech_No
    df2 = pd.read_csv("churn.csv")
    st.write("The values you have selected are:")
    st.write(df.head())
    df=df.drop(['charge'],axis=1)
    df2.dropna(inplace = True)
    df2['Churn'].replace(to_replace='Yes', value=1, inplace=True)
    df2['Churn'].replace(to_replace='No',  value=0, inplace=True)
    df3=df2['charge']
    df2=df2.drop(['charge'],axis=1)
    df2=pd.get_dummies(df2)
    #df2['charge']=df3
    #st.write(df2.head())
    y = df2['Churn'].values
    X = df2.drop(columns = ['Churn'])
    #st.write(X)
    #X['charge']=df3
    model = SGDClassifier()
    model.fit(X, y)
    preds = model.predict(df)
    if preds==0:
      st.write("The predicted is Not Churn")
    if preds==1:
      st.write("The predicted is Churn")
    st.write(preds)
```

```
if __name__ == '__main__':
  main()
```

```
Overwriting app.py
```

```
! pip install pyngrok
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
Collecting pyngrok
  Downloading pyngrok-5.2.1.tar.gz (761 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 761.3/761.3 KB 17.1 MB/s eta 0:00:00
  Preparing metadata (setup.py) ... done
Requirement already satisfied: PyYAML in /usr/local/lib/python3.9/dist-packages (from pyngrok) (6.0)
Building wheels for collected packages: pyngrok
  Building wheel for pyngrok (setup.py) ... done
  Created wheel for pyngrok: filename=pyngrok-5.2.1-py3-none-any.whl size=19790 sha256=5ceb4666772c726f018e829f61f066b26477c650a6261e0ab
  Stored in directory: /root/.cache/pip/wheels/f6/89/59/49d4249e00957e94813ac136a335d10ed2e09a856c5096f95c
Successfully built pyngrok
Installing collected packages: pyngrok
Successfully installed pyngrok-5.2.1
```

```
from pyngrok import ngrok
```

```
ngrok.set_auth_token("2NoQxNH5rYDFIyhaLwB5bOqkHyN_2ysdvG3ToLFuTtWzk7bRr") #ngrok.com
```

```
!nohup streamlit run app.py --server.port 80 &
url = ngrok.connect(port = '80')
print(url)
```

```
nohup: appending output to 'nohup.out'
NgrokTunnel: "http://059d-35-197-113-128.ngrok-free.app" -> "http://localhost:80"
```

Colab paid products - Cancel contracts here