CAPSTONE PROJECT 2 - ANN CLASSIFICATION - BANK CHURN RATE PROBABLITY PREDICTION USING VS CODE and STREAMLIT APP

By Omkar Gadade (PGA28)
THANE

TOOLS USED:

a) VS CODE
b) STREAMLIT (For Website Creation)
c) STREAMLIT Cloud (For Deployment purpose via GitHub Repo)
d) LIBRARIES: Pandas, Numpy, TensorFlow, Keras, ipykernel, TensorBoard, Scikit-learn
e) Excel

Data:

Input Features

O/P

RowNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	15634602	Hargrave	619	France	Female	42	2	0	1	1	1	101348.88	1
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
3	15619304	Onio	502	France	Female	42	8	159660.8	3	1	0	113931.57	1
4	15701354	Boni	699	France	Female	39	1	0	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.1	0

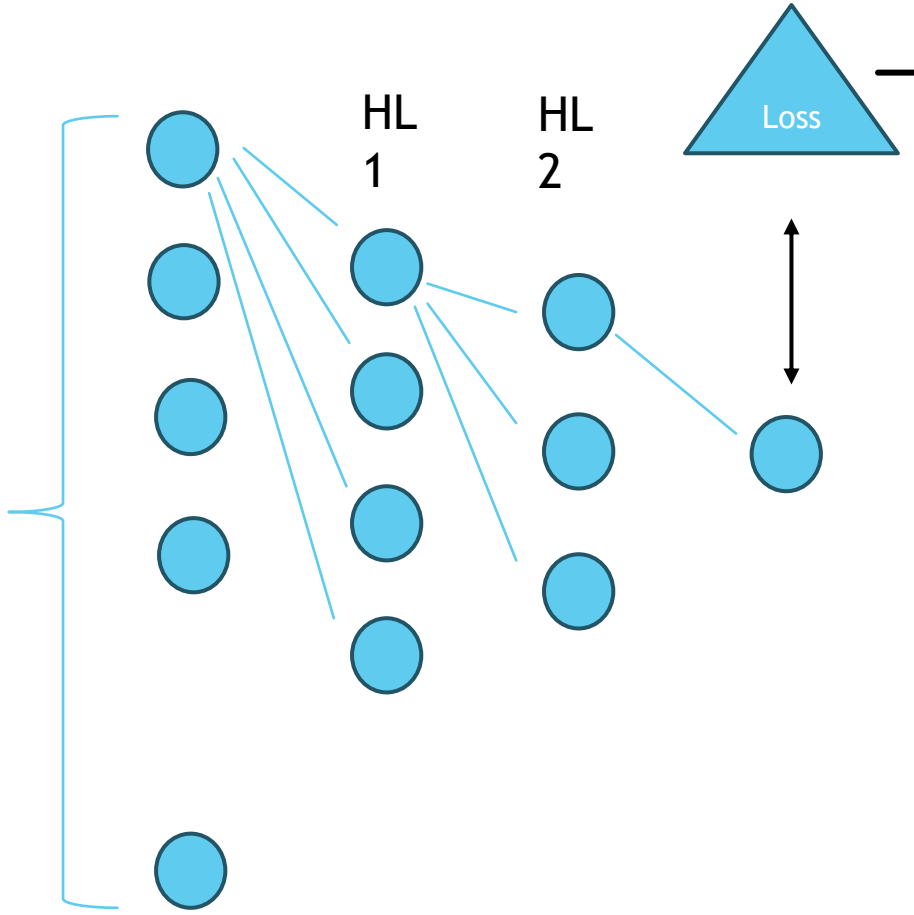
I/P Features:10

Procedure

CHURN Modelling Dataset .csv

I/P Features

BASIC F.E. (One Hot Encoding Label Encoding) (Standardization)



- HL1, HL2 : Relu
- Output: Sigmoid
- Optim: Adam Optimizer

Pickle (.pkl) and (.h5) file

STREAMLIT WEB APP

Deployment to Streamlit Cloud

ANN Model Parameters:

```
# Build ANN Model

model = Sequential([
    Dense(64,activation='relu',input_shape=(12,)),  ## HL1 Connected with input layer
    Dense(32, activation='relu'),    ## HL2
    Dense(1, activation='sigmoid')  ## Output Layer
])
```

✓ 0.1s

ANN Model Parameters:

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	832
dense_1 (Dense)	(None, 32)	2,080
dense_2 (Dense)	(None, 1)	33

Total params: 2,945 (11.50 KB)

Trainable params: 2,945 (11.50 KB)

Non-trainable params: 0 (0.00 B)

ANN Model Optimizer, Learning Rate and Loss:

```
import tensorflow
opt = tensorflow.keras.optimizers.Adam(learning_rate=0.01)
```

✓ 0.0s

```
## compile the model for forward and backward propogation
model.compile(optimizer=opt, loss='binary_crossentropy', metrics=['accuracy'])
```

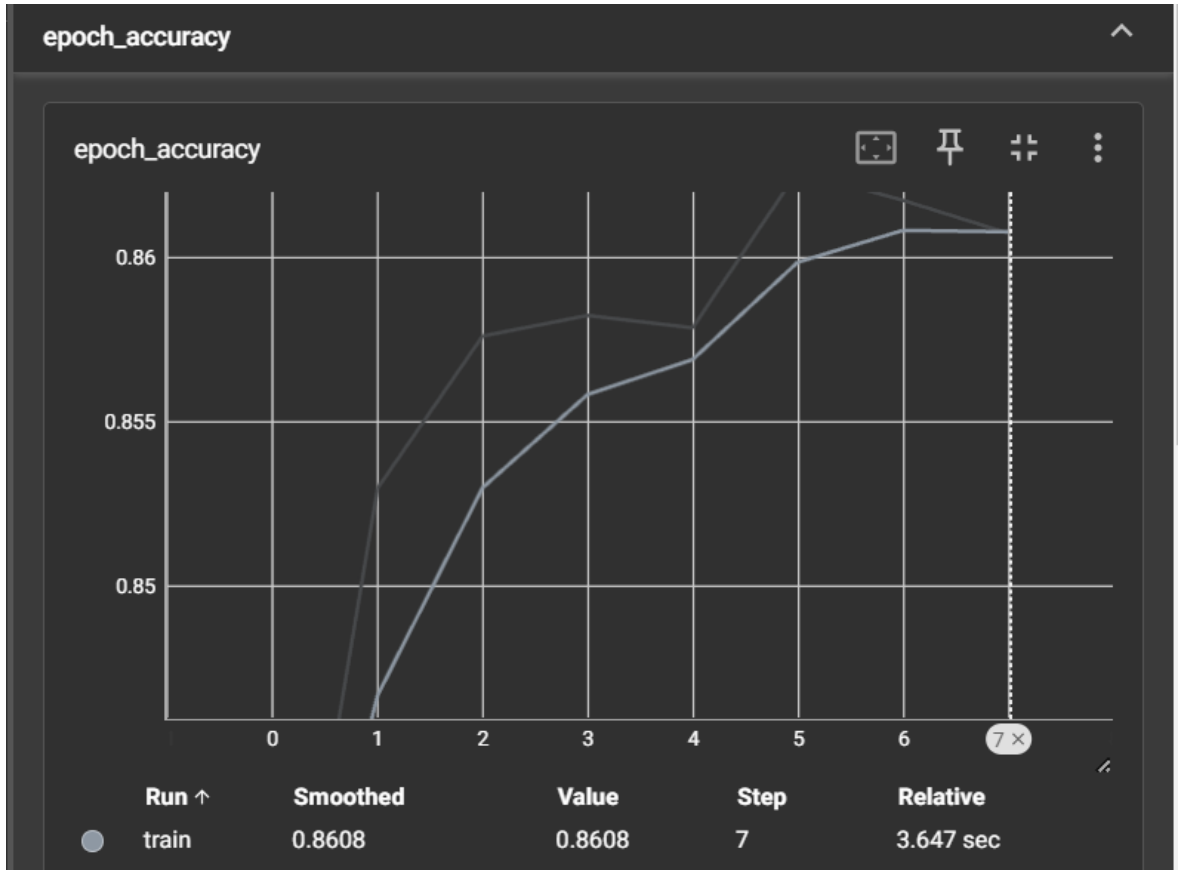
Training the model:

```
## Train the model
history = model.fit(
    X_train, y_train, validation_data = (X_test,y_test), epochs=100,
    callbacks=[tensorflow_callback,early_stopping_callback]
)
```

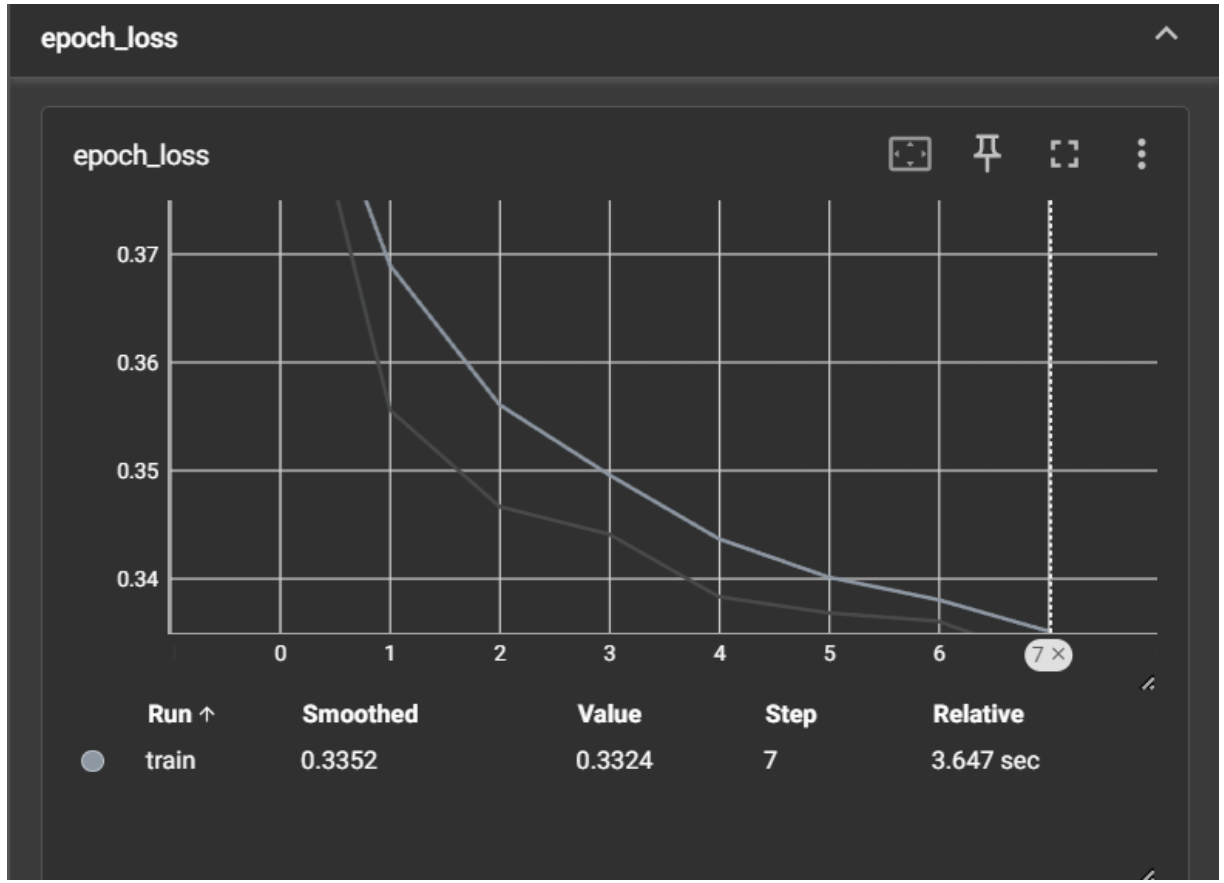
✓ 5.6s

```
Epoch 1/100
250/250 ━━━━━━━━━━━━━━━━━━━━ 2s 3ms/step - accuracy: 0.8056 - loss: 0.4442 - val_accuracy: 0.8600 - val_loss: 0.3494
Epoch 2/100
250/250 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8583 - loss: 0.3467 - val_accuracy: 0.8555 - val_loss: 0.3636
Epoch 3/100
250/250 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8572 - loss: 0.3435 - val_accuracy: 0.8595 - val_loss: 0.3431
Epoch 4/100
250/250 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8607 - loss: 0.3427 - val_accuracy: 0.8605 - val_loss: 0.3449
Epoch 5/100
250/250 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8579 - loss: 0.3393 - val_accuracy: 0.8620 - val_loss: 0.3441
Epoch 6/100
250/250 ━━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - accuracy: 0.8608 - loss: 0.3348 - val_accuracy: 0.8585 - val_loss: 0.3558
Epoch 7/100
250/250 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8638 - loss: 0.3344 - val_accuracy: 0.8570 - val_loss: 0.3556
Epoch 8/100
250/250 ━━━━━━━━━━━━━━━━━━━━ 1s 2ms/step - accuracy: 0.8664 - loss: 0.3236 - val_accuracy: 0.8620 - val_loss: 0.3454
```

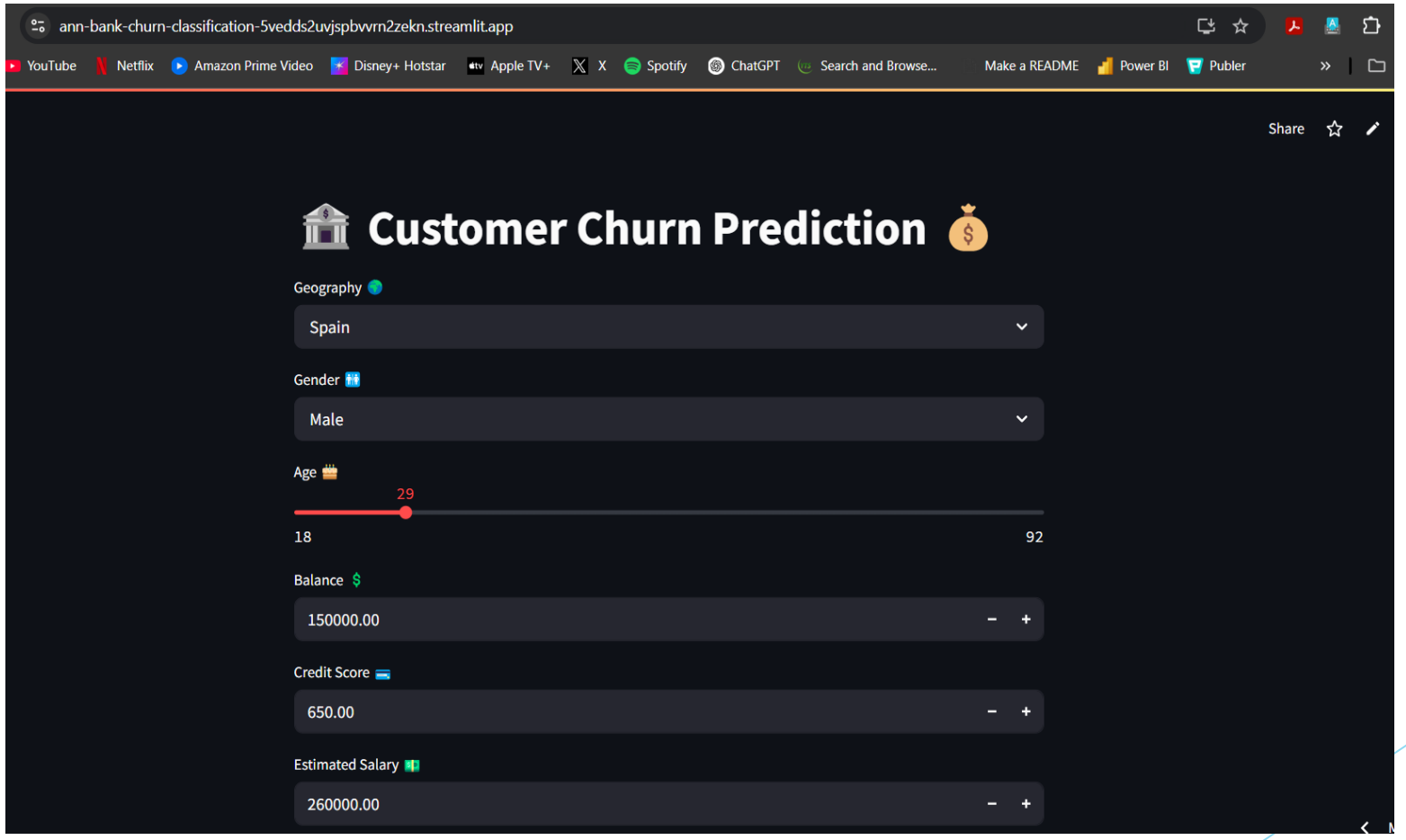
Tensorboard Visualization for Epoch Accuracy and Loss



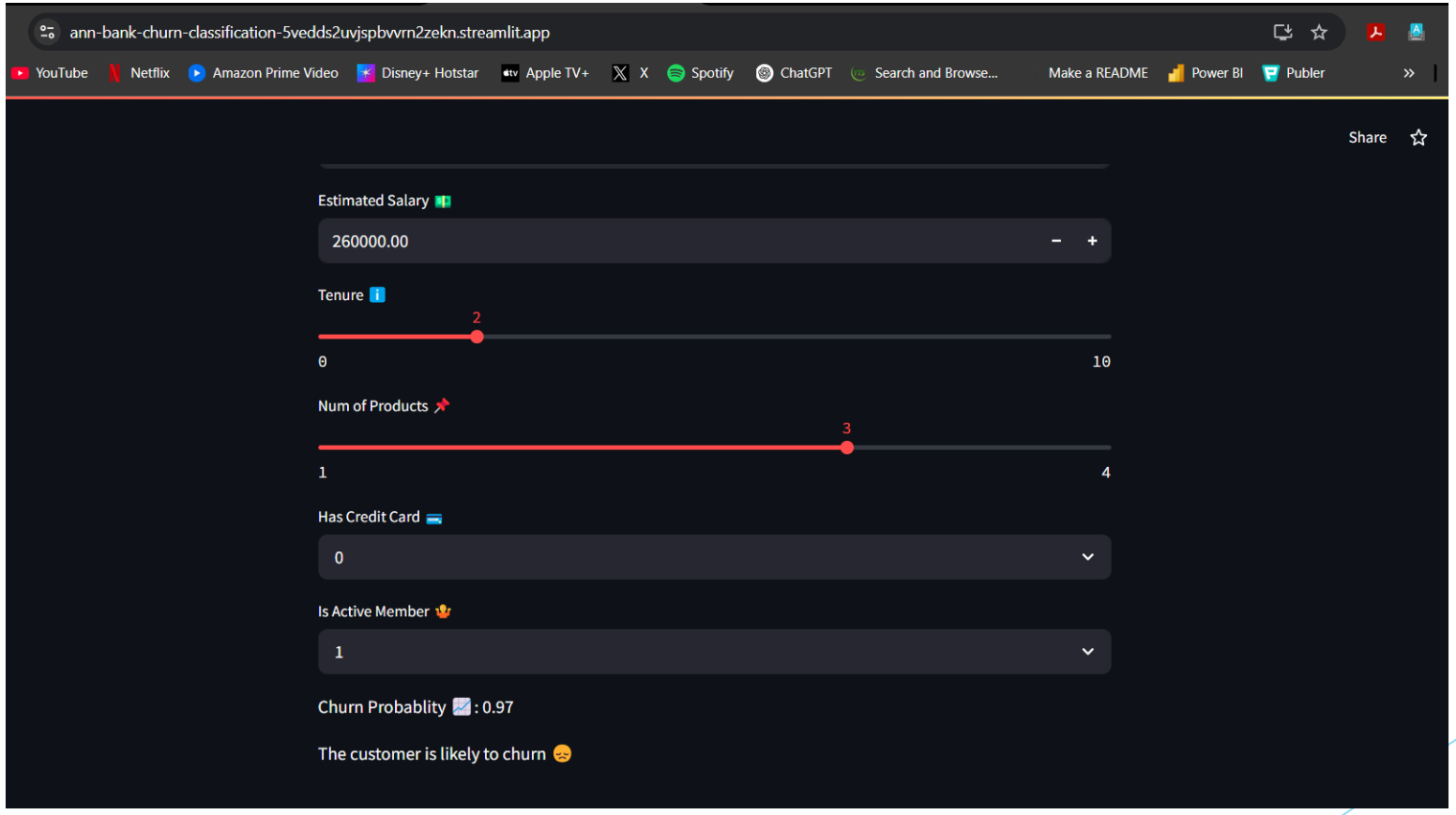
Tensorboard Visualization for Epoch Accuracy and Loss



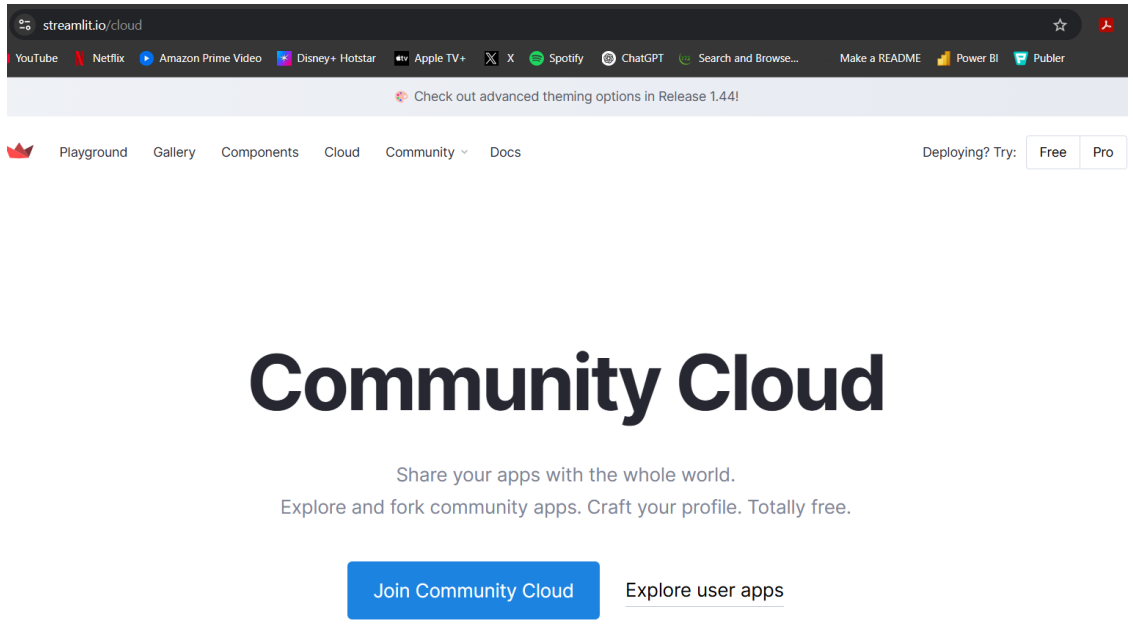
Streamlit APP Deployment



Streamlit APP Deployment



Streamlit APP Deployment



- URL: https://ann-bank-churn-classification-5vedds2uvjspbvvrn2zekn.streamlit.app/

GitHub Repo: https://github.com/Omkar-Gadade/ANN-Bank-Churn-Classification