

Streams_Prediction_Calculator

November 18, 2024

0.1 Sreams Prediction Calculator

Building an interactive widget-based user interface for predicting song streams based on input features using the cloud function we built. The interface allows users to input values for features through sliders and toggle buttons, sends the data to the cloud function for predictions, and displays the results dynamically.

```
[1]: import ipywidgets as widgets
from IPython.display import display
import requests
import numpy as np

# Define the URL for your cloud function
url = "https://us-west2-streamsregression.cloudfunctions.net/streams_prediction"

# Define input widgets with realistic ranges and bold titles
acousticness = widgets.FloatSlider(description='Acousticness:',min=0.0, max=1.0,
    ↪0, step=0.01, value=0.3,style={'description_width': 'initial'})
liveness = widgets.FloatSlider(description='Liveness:',min=0.0, max=1.2, step=0.01,
    ↪0, value=0.2,style={'description_width': 'initial'})
speechiness = widgets.FloatSlider(description='Speechiness:',min=0.0, max=0.4,
    ↪step=0.01, value=0.1,style={'description_width': 'initial'})
instrumentalness = widgets.FloatSlider(description='Instrumentalness:',min=0.0,
    ↪max=1.0, step=0.01, value=0.3,style={'description_width': 'initial'})
duration_ms = widgets.IntSlider(description='Duration:',min=10000, max=600000,
    ↪step=10000, value=200000,style={'description_width': 'initial'})
valence = widgets.FloatSlider(description='Valence:',min=0.0, max=1.0, step=0.01,
    ↪0, value=0.5,style={'description_width': 'initial'})
comments = widgets.IntSlider(description='Comments:',min=0, max=2000, step=10,
    ↪value=100,style={'description_width': 'initial'})

# ToggleButtons for Licensed
licensed = widgets.ToggleButtons(options=[('Yes', 1), ('No',
    ↪0)],description='Licensed:',value=1,style={'description_width': 'initial'})
# ToggleButtons for Album Single
album_single = widgets.ToggleButtons(options=[('Yes', 1), ('No',
    ↪0)],description='Album Single:',value=1,style={'description_width':
    ↪'initial'})
```

```

# Define output widget
output = widgets.Output()

# Define function to handle button click event
def on_predict_button_clicked(b):
    with output:
        output.clear_output()
        # Prepare input dictionary
        input_data = {
            "Acousticness": acousticness.value,
            "Liveness": liveness.value,
            "Speechiness": speechiness.value,
            "Instrumentalness": instrumentalness.value,
            "Licensed": licensed.value,
            "Duration_ms": duration_ms.value,
            "Valence": valence.value,
            "Comments": comments.value,
            "Album_single": album_single.value
        }
        # Call the cloud function with the input data
        try:
            response = requests.post(url, json=input_data)
            response_data = response.json()
            if "Stream Prediction (log)" in response_data:
                streams_prediction = np.expml(response_data["Stream Prediction_
↵(log)"])
                print("\033[1;34mPredicted Streams:\033[0m",
↵f"{streams_prediction:,.2f} streams")
            else:
                print("Error:", response_data.get("message", "Unexpected server_
↵response."))
        except Exception as e:
            print("Error:", str(e))

# Define predict button with blue and bold styling
predict_button = widgets.Button(
    description='Predict',
    button_style='primary',
    layout=widgets.Layout(width='100px', height='40px'),
    style={'font_weight': 'bold'})
predict_button.on_click(on_predict_button_clicked)

# Add a title with `widgets.HTML`
title = widgets.HTML("<h3><b>Song Features Input Form</b></h3>")
spacer = widgets.HTML("<br>")

```

```
[2]: # Display input widgets
display(widgets.VBox([title, acousticness, liveness, speechiness,
↳instrumentalness,duration_ms, valence, comments, licensed, album_single,
↳spacer, predict_button, output]))

VBox(children=(HTML(value='<h3><b>Song Features Input Form</b></h3>'),
↳FloatSlider(value=0.3, description='Aco...
```