Human Faces(Object Detection)

Human faces(Object Detection)

Build a **lightweight, CPU-friendly human face detection system** that goes end-to-end—from data prep to real-time deployment—using Ultralytics/YOLO.

Project Overview

This project has two parts:

1.Computer Vision — Train, evaluate, export and deploy a YOLOv5n face detector (Ultralytics).

2.Streamlit App (Tabular ML) — A general machine-learning app for any CSV: Data view, EDA, Train/Evaluate, and single-row Prediction.

Technologies Used

- **PYTHON**
- **ANALYTICS**
- **STATISTICS**
- **PLOTING**
- **STREAMLIT**
- **MACHINE LEARNING**
- **DEEP LEARNING**
- **GENAI**

Run a code

1. Data Preprocessing
python scripts/data_preprocessing.py
It will clean and give a proper formatted data to a model – faces_cleaned.csv

2. Exploratory_data_analysis
python scripts/exploratory_data_analysis.py
Will do EDA for the csv file and give eda plots

3. Feature selection
python scripts/feature.py
Will select feature and will return feature.npy

4. generate_labels
python scripts/generate_labels.py
helps to generate labels for all images – labels.npy

5.Train test split
Python scripts/ train_test_split.py
will train and test the model and will return x_test and y_train

Run a code

6.Model selection
python scripts/model_selection.py
helps to choose the model – yolov5n.pt

7. Train model
python scripts/train_model.py
Will train the model - runs

8. Convert to labels
python scripts/convert_to_labels.py
will convert to labels - labels

9.Evaluate
python scripts/evaluate.py
Will do the evaluation and will give results – evaluation_results

10. Deploy model
Python scripts/ deploy_model.py
Will deploy the evaluated model – annotated_mp4

Run a code

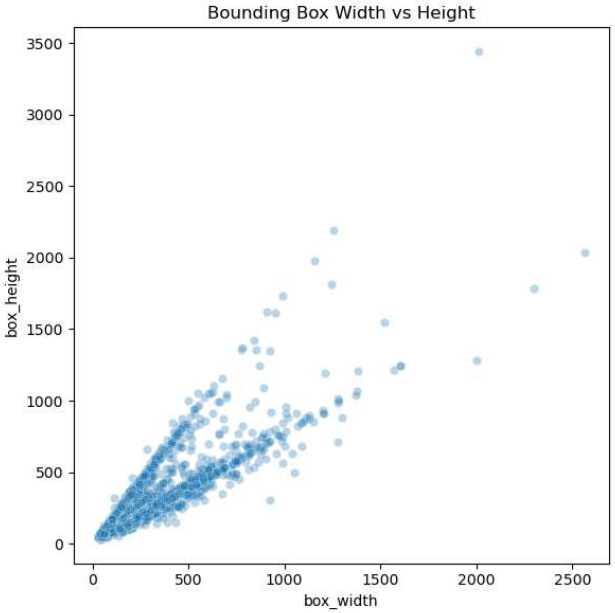
11.Hyperparameter evolve
python scripts/hyp_evolve.py
Will create yolov5 model – yolov5.yaml

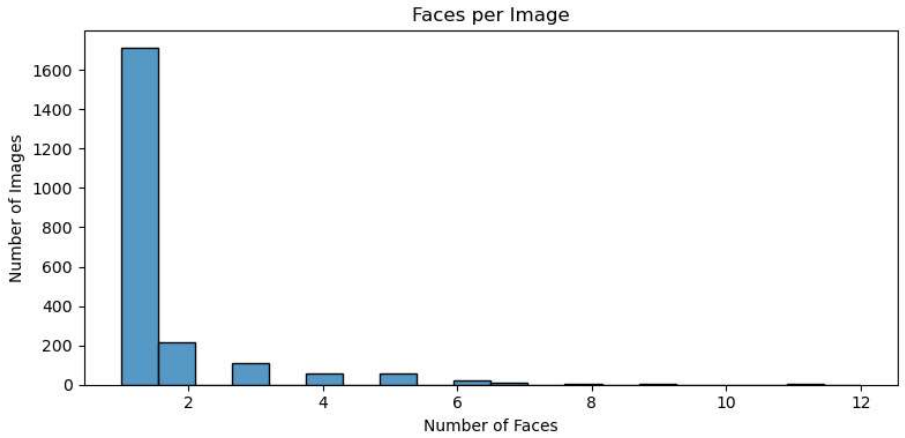
12.Export model
python scripts/export_model.py
will export the trained model

13. Test onnx
python scripts/test_onnx.py
Will test and returns Onnx-results

14.Streamlit
python scripts/app.py
To push the project to streamlit dashboard

EDA plots


Bounding Box Width vs Height
box_height
box_width
0
500
1000
1500
2000
2500
3000
3500


Faces per Image
Number of Images
Number of Faces
0
200
400
600
800
1000
1200
1400
1600
2
4
6
8
10
12

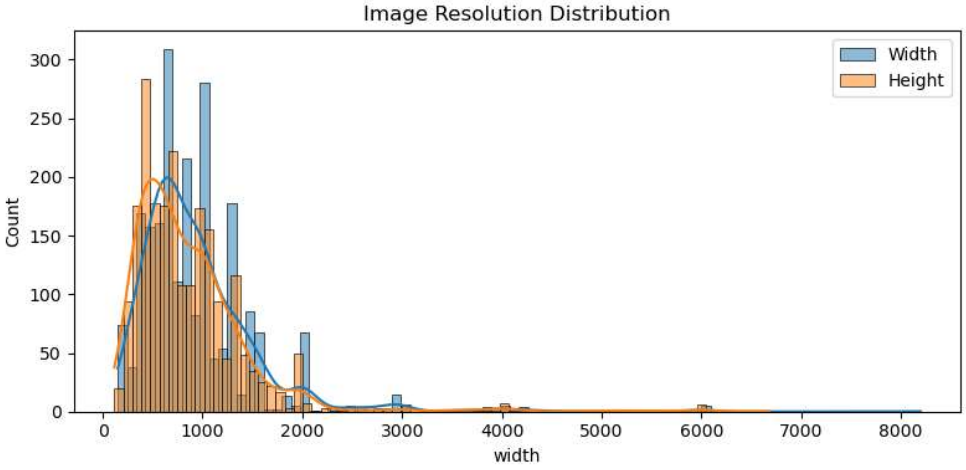
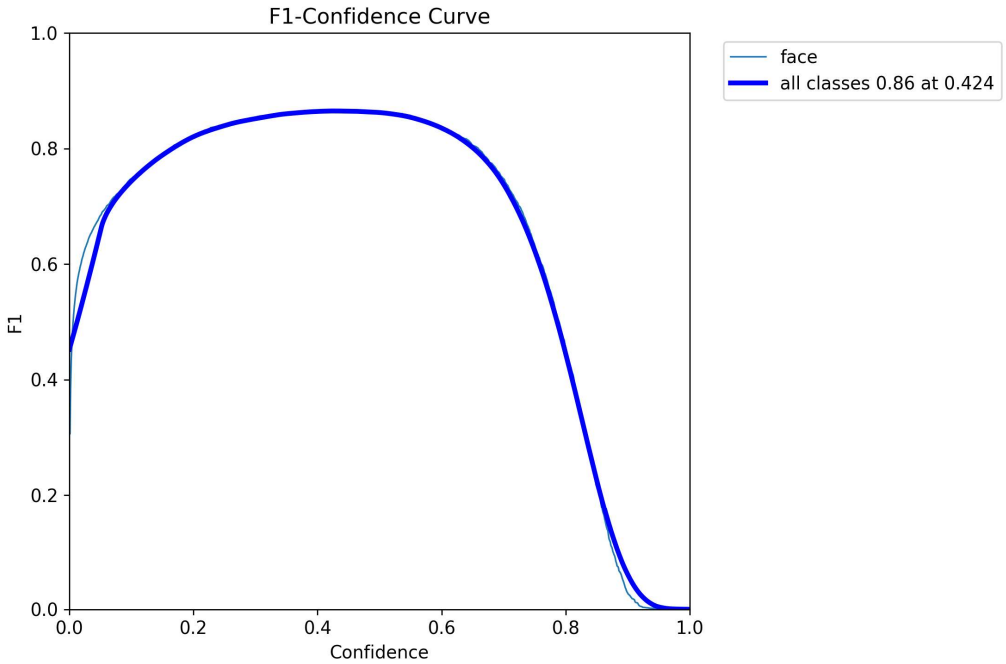
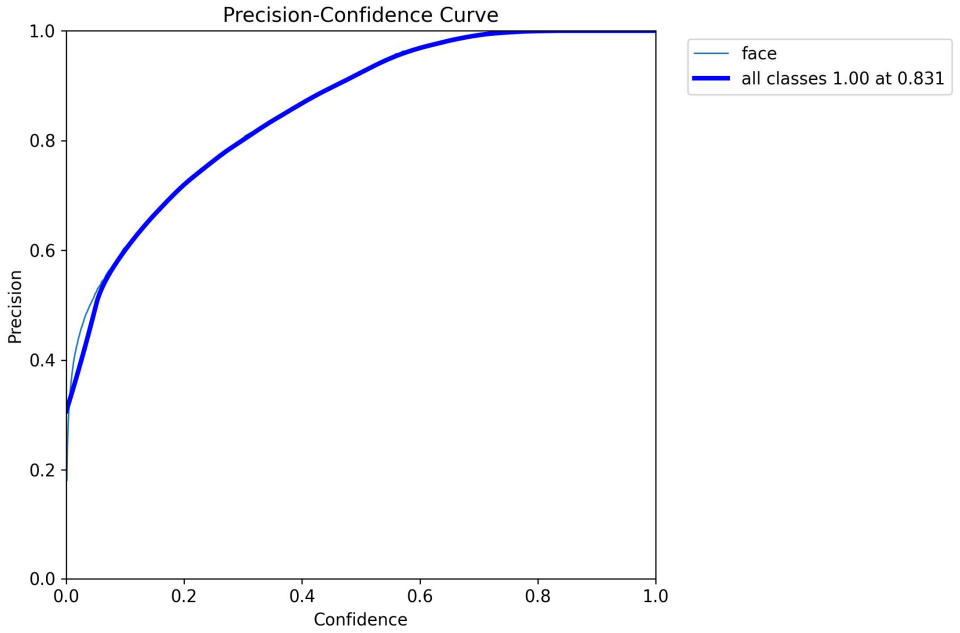


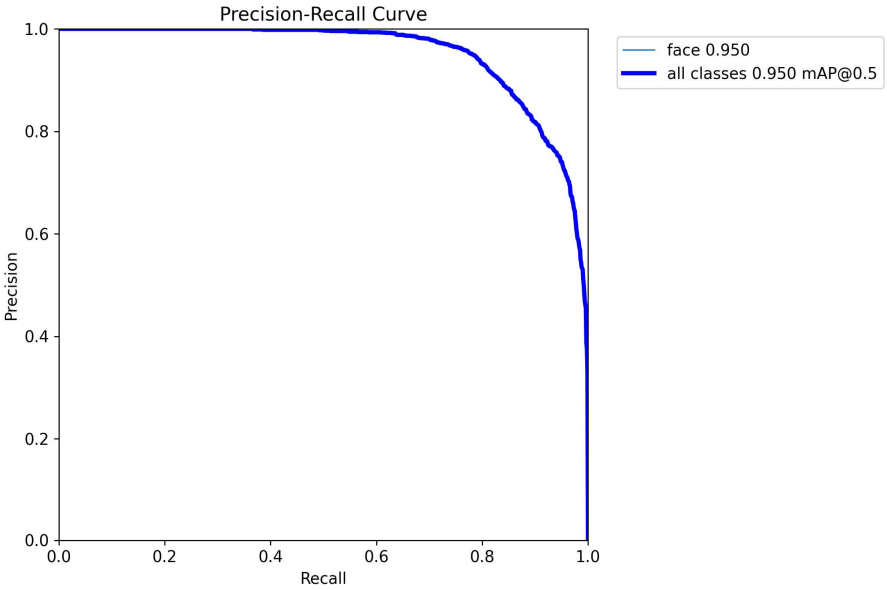
Image Resolution Distribution
Width
Height
Count
width
0
50
100
150
200
250
300
1000
2000
3000
4000
5000
6000
7000
8000

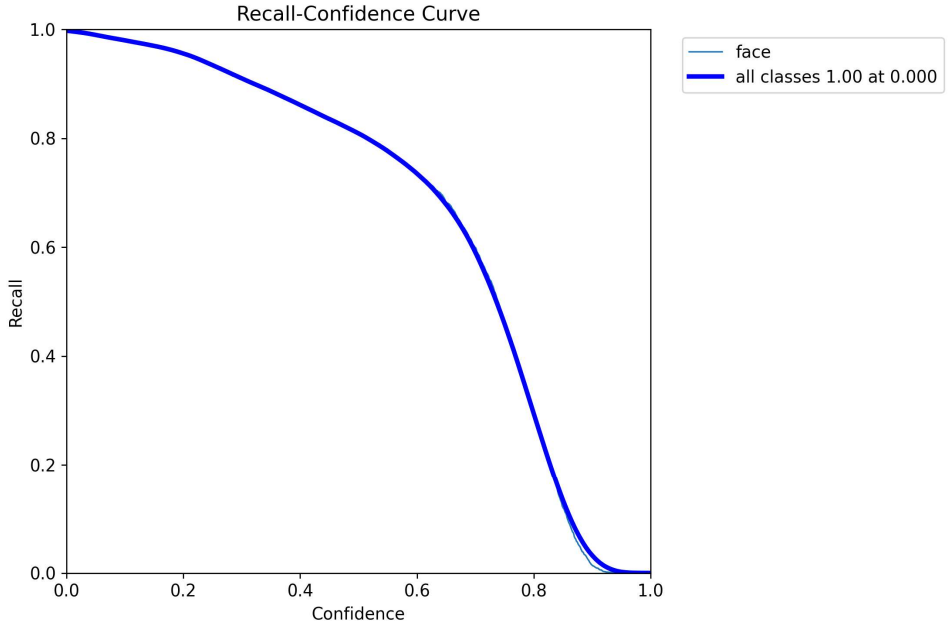
After training the model



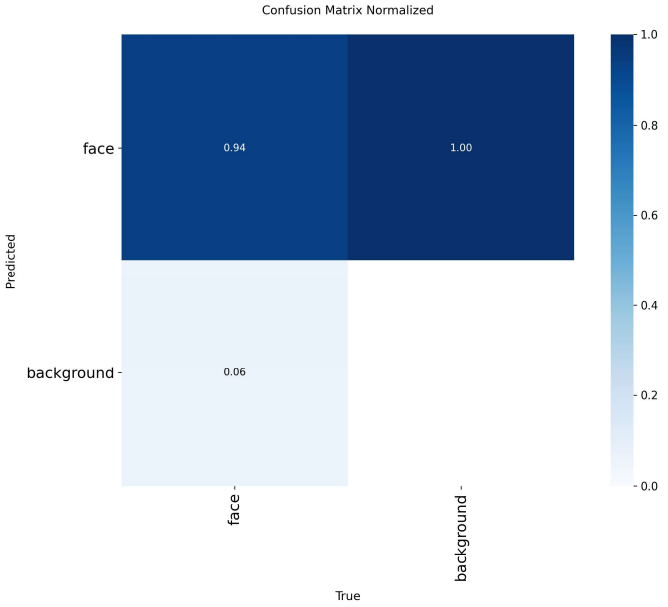


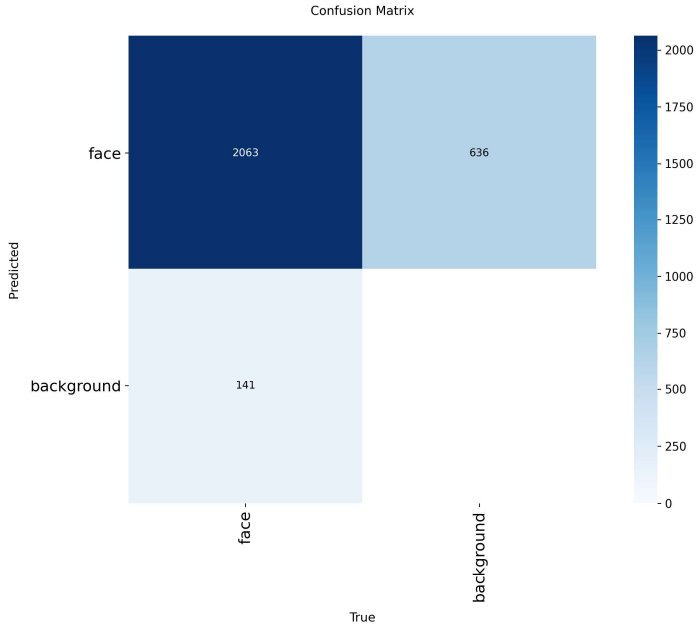
After training the model





After training the model





After training the model

Model Evaluation result

{} evaluation_results.json > ...


```
{
    "precision": 0.883630210173412,
    "recall": 0.8487667138432158,
    "map50": 0.949870084896072,
    "map50-95": 0.6277144821479653,
    "metrics/precision(B)": 0.883630210173412,
    "metrics/recall(B)": 0.8487667138432158,
    "metrics/mAP50(B)": 0.949870084896072,
    "metrics/mAP50-95(B)": 0.6277144821479653,
    "fitness": 0.659930042422776
}
```


Onnx Result

Streamlit

User

Number

1

Name

Kiran

Dataset

Upload CSV (optional)

Drag and drop file here

Limit 200MB per file • CSV

Browse files

faces_cleaned.csv

82.2KB

Human Face Detection — Tabular + Image Prediction

Data | EDA - Visual | Train / Predict

Dataset Preview

Source: (uploaded) faces_cleaned.csv

	image_name	width	height	x0	y0	x1	y1
0	00001722.jpg	1333	2000	490	320	687	664
1	00001044.jpg	2000	1333	791	119	1200	436
2	00001050.jpg	667	1000	304	155	407	331
3	00001736.jpg	626	417	147	14	519	303
4	00003121.jpg	626	418	462	60	599	166
5	00000400.jpg	600	400	148	61	377	242
6	00002571.jpg	960	720	117	159	317	334
7	00000366.jpg	900	601	170	88	506	350
8	00002565.jpg	1267	712	241	40	963	513
9	00001939.jpg	612	424	195	57	383	209

Columns & dtypes

	column	dtype
image_	image_name	object
width	width	int64

Streamlit

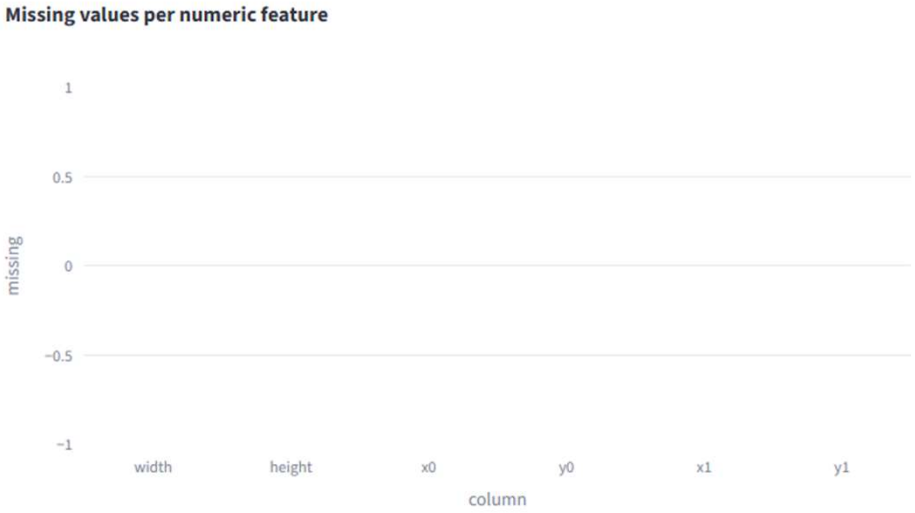
Data EDA - Visual Train / Predict

Quick Numeric EDA

Descriptive Statistics

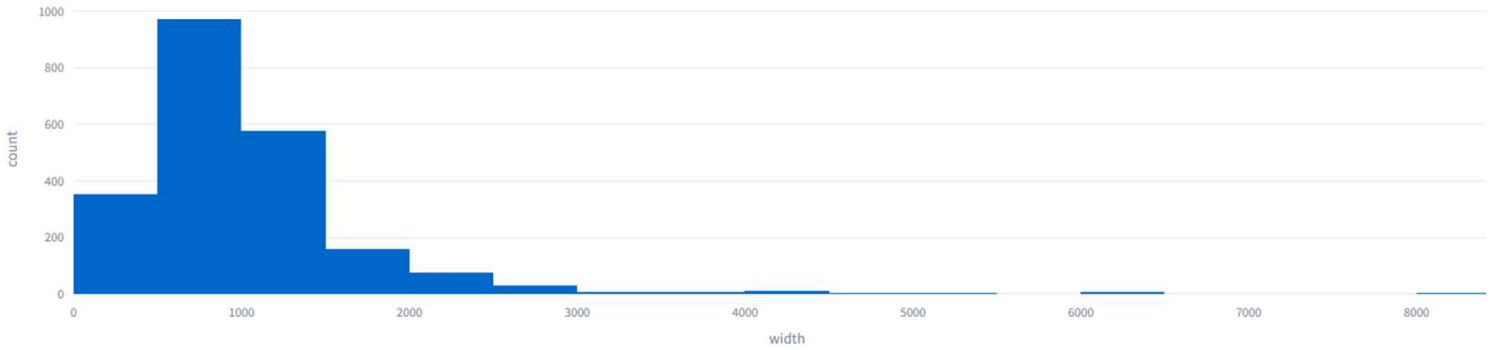
	count	mean	std	min	25%	50%	75%	max
width	2204	956.5454	649.9971	150	600	800	1200	8192
height	2204	874.8612	650.2013	115	459.75	700.5	1050	6680
x0	2204	350.9619	301.6931	1	163	281	444	3976
y0	2204	138.2886	171.2599	1	45	90.5	173	2375
x1	2204	617.0839	441.9905	88	346.75	509	774	5544
y1	2204	407.3807	331.9798	48	212	327	492.25	4471

Missing values

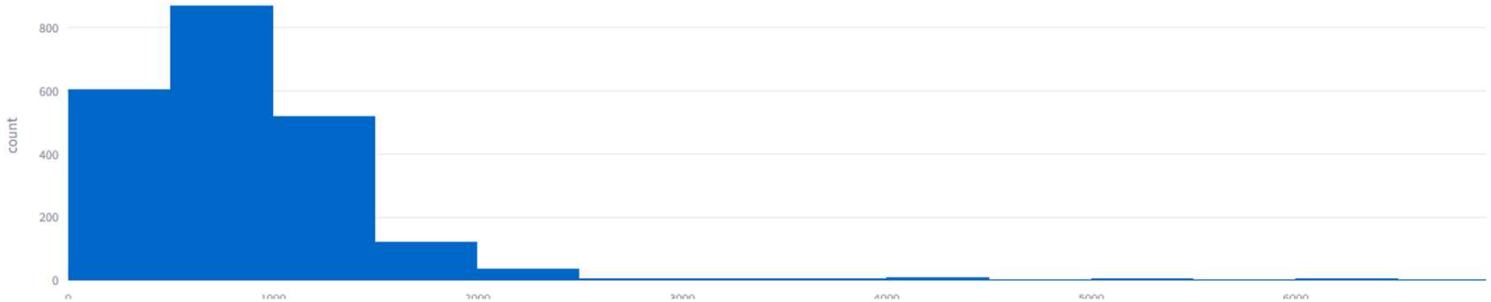


Streamlit

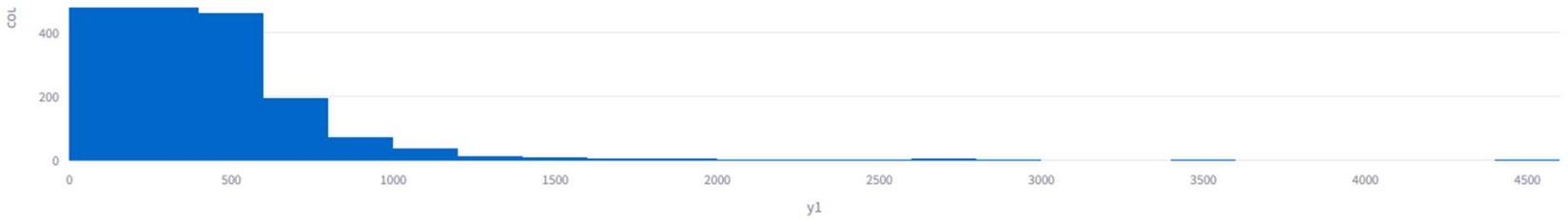
Histogram of width



Histogram of height

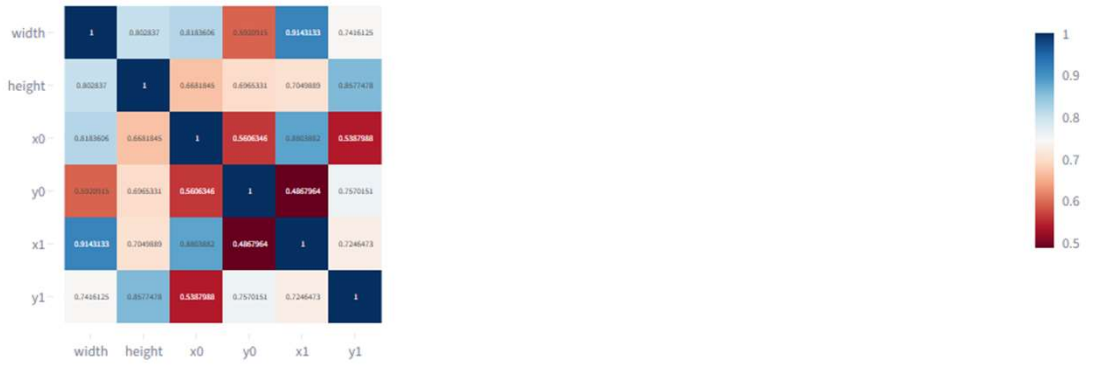


Streamlit



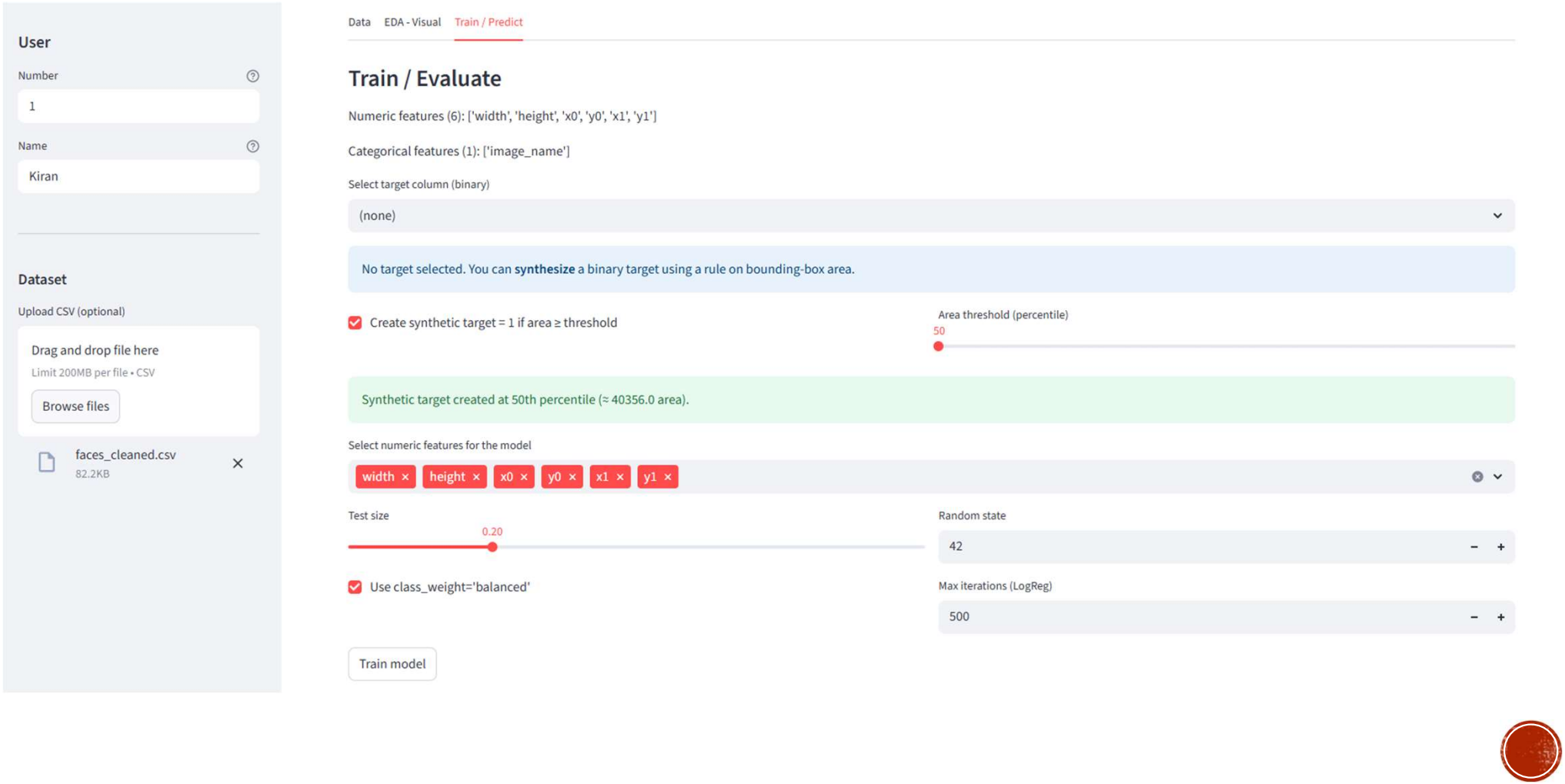
Correlation heatmap

Correlation

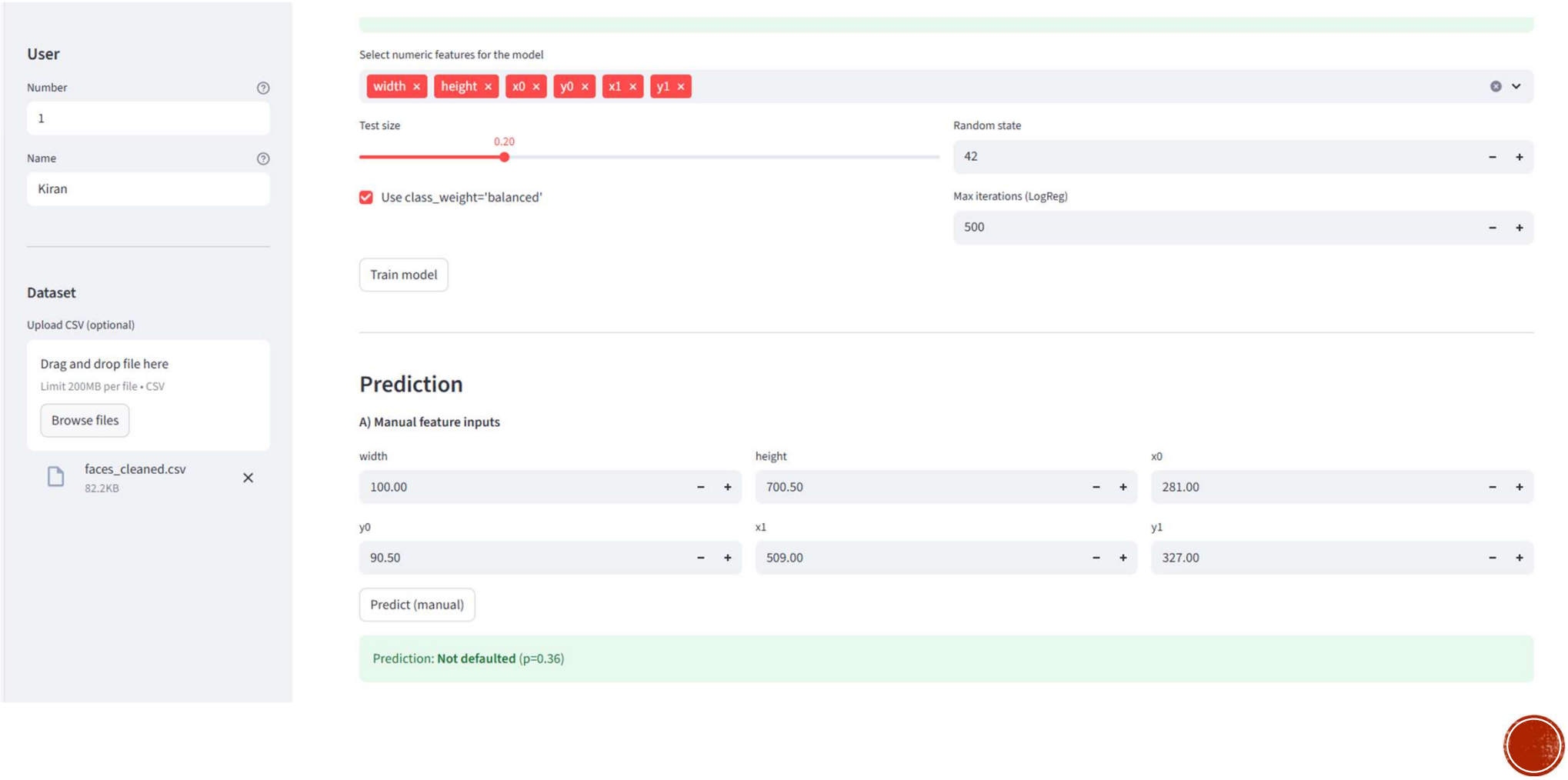


Tip: If you get a blank page, check your terminal for Python errors. Most issues are due to a single-class target or missing numeric features.

Streamlit


User
Number
1
Name
Kiran
Dataset
Upload CSV (optional)
Drag and drop file here
Limit 200MB per file • CSV
Browse files
faces_cleaned.csv
82.2KB
Data
EDA - Visual
Train / Predict
Train / Evaluate
Numeric features (6): ['width', 'height', 'x0', 'y0', 'x1', 'y1']
Categorical features (1): ['image_name']
Select target column (binary)
(none)
No target selected. You can synthesize a binary target using a rule on bounding-box area.
Create synthetic target = 1 if area ≥ threshold
Area threshold (percentile)
50
Synthetic target created at 50th percentile (≈ 40356.0 area).
Select numeric features for the model
width × height × x0 × y0 × x1 × y1 ×
Test size
0.20
Random state
42
Use class_weight='balanced'
Max iterations (LogReg)
500
Train model

Streamlit


User
Number
1
Name
Kiran
Dataset
Upload CSV (optional)
Drag and drop file here
Limit 200MB per file • CSV
Browse files
faces_cleaned.csv
82.2KB
Select numeric features for the model
width × height × x0 × y0 × x1 × y1 ×
Test size
0.20
Random state
42
Use class_weight='balanced'
Max iterations (LogReg)
500
Train model
Prediction
A) Manual feature inputs
width
100.00
height
700.50
x0
281.00
y0
90.50
x1
509.00
y1
327.00
Predict (manual)
Prediction: Not defaulted (p=0.36)