A Machine Learning Model for Diabetes

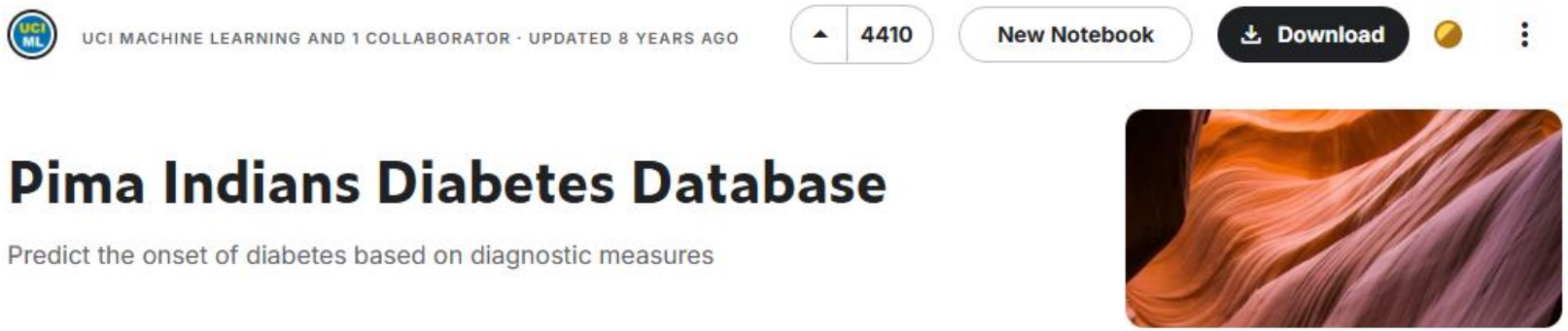
Hassan Shojaee-Mend
Assistant Professor of Medical Informatics
Gonabad University of Medical Sciences

Agenda

- ⇨ Downloading the Pima Diabetes dataset from Kaggle
- ⇨ Preprocessing the data
- ⇨ Training a model using Hugging Face AutoTrain
- ⇨ Deploying the model with Gradio in Hugging Face Spaces

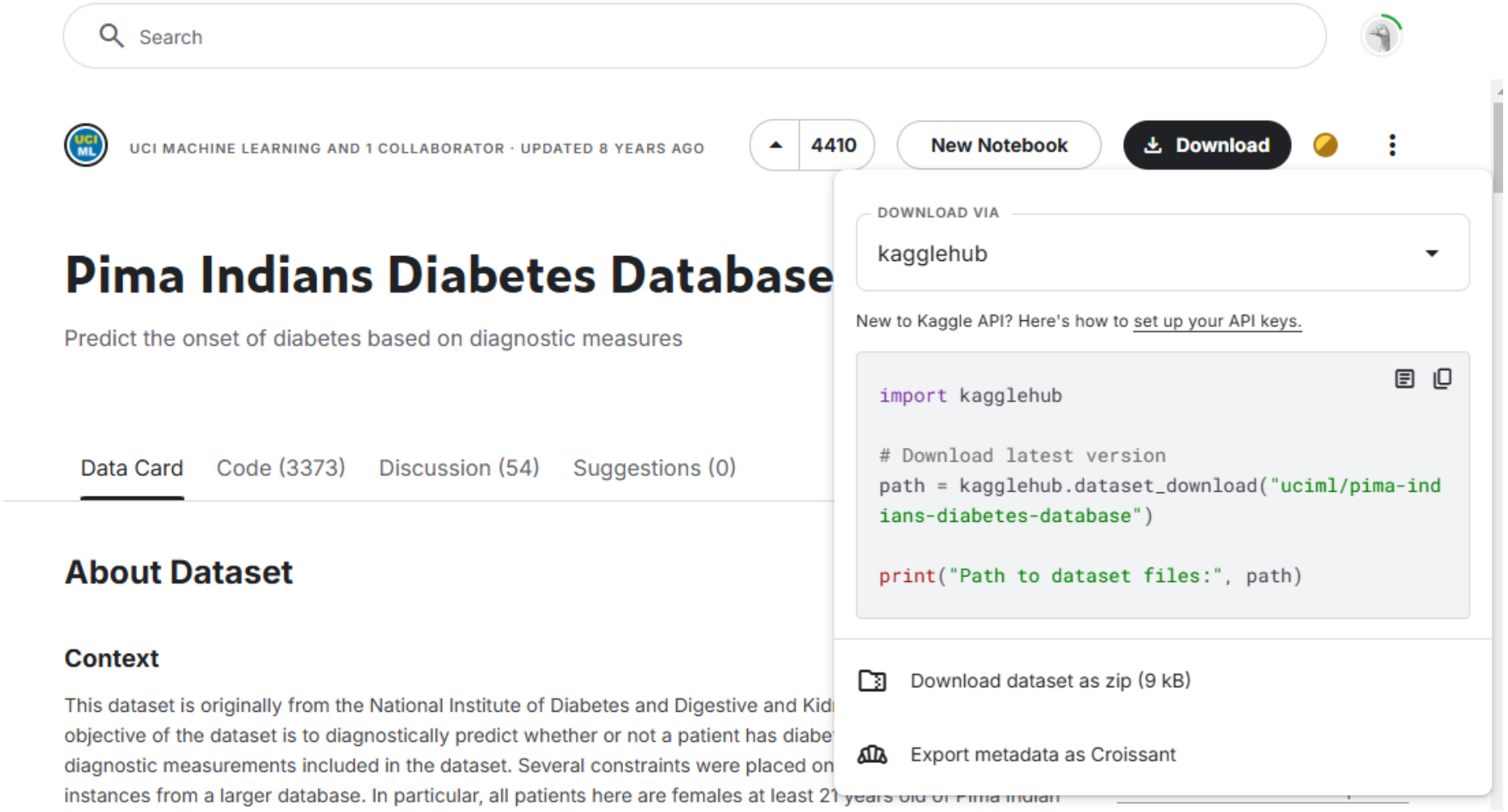
Pima Diabetes Dataset

- Predict the onset of diabetes based on diagnostic measures
- 768 patients are females at least 21 years old of Pima Indian heritage.
- Predictor variables includes the number of pregnancies the patient has had, their BMI, insulin level, age, and so on. (8 predictors)
- Target variable: Outcome (0,1)



https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

Downloading the Dataset from Kaggle



https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

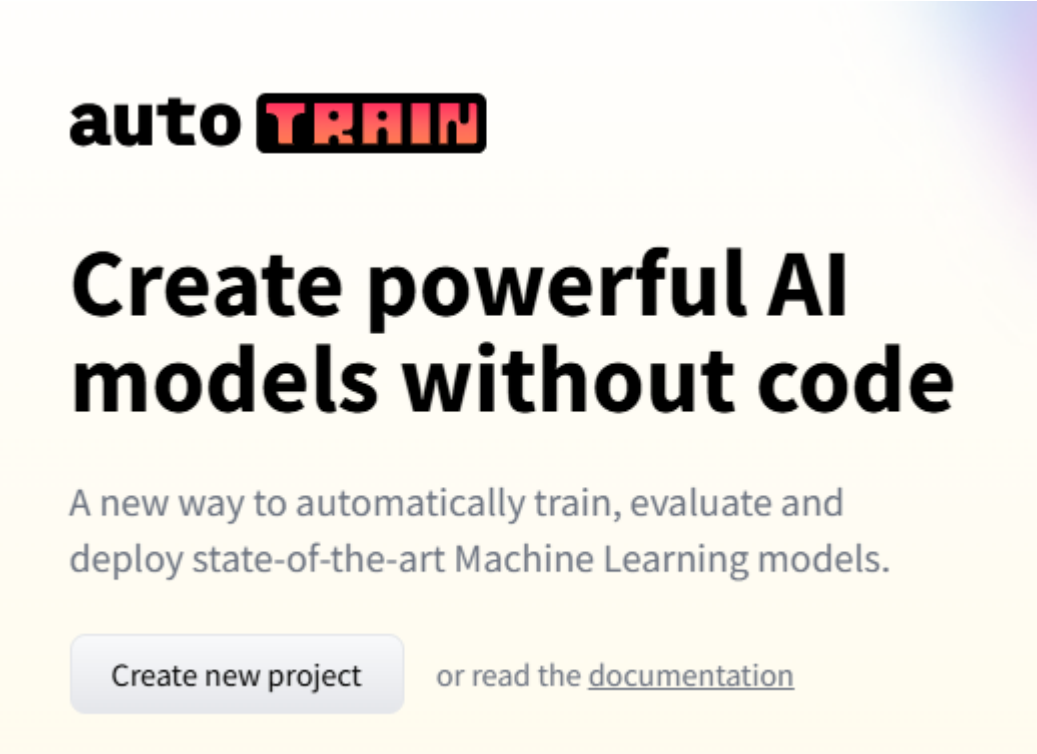
Data Preprocessing and Visualization

- Replace zeros with NaN for columns where zero is invalid (Glucose, BloodPressure, SkinThickness, Insulin, BMI)

# Pregnancies Number of times pregnant	# Glucose Plasma glucose concentration a 2 hours in an oral glucose tolerance test	# BloodPressure Diastolic blood pressure (mm Hg)	# SkinThickness Triceps skin fold thickness (mm)	# Insulin 2-Hour serum insulin (mu U/ml)	# BMI Body mass index in kg/(height in m)
0 … 17	0 … 199	0 … 122	0 … 99	0 … 846	0 …
6	148	72	35	0	33.6
1	85	66	29	0	26.6
8	183	64	0	0	23.3
1	89	66	23	94	28.1

Training with AutoTrain

- A no-code tool for training ML models
- Go to Hugging Face AutoTrain
- Upload the preprocessed dataset
- Choose the target column (diabetes outcome)
- Select model type and train
- Wait for training to complete!



https://huggingface.co/autotrain

Deploying on Hugging Face Spaces

- Deploy with Gradio: A Python library for building ML apps
- Create a new Hugging Face Space
- Select Gradio as the SDK
- Write a simple Gradio interface
- Upload the trained model
- Create requirements.txt
- Deploy!

Writing the Gradio App Code (app.py)

```
import gradio as gr
import numpy as np
import joblib
model = joblib.load("model4.joblib")
def  myfunc(Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age):
  data = np.array([[Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age]])
  p = model.predict_proba(data)
  return p[0][1]

demo = gr.Interface(
  fn=myfunc,
  inputs=[
    gr.Number(label="Pregnancies"),
    gr.Number(label="Glucose"),
    gr.Number(label="Blood Pressure"),
    gr.Number(label="Skin Thickness"),
    gr.Number(label="Insulin"),
    gr.Number(label="BMI"),
    gr.Number(label="Diabetes Pedigree Function"),
    gr.Number(label="Age")],
  outputs=gr.Textbox(label="Probability"),
  title="Diabetes prediction App",
  description="Enter patient info to predict diabetes risk")
demo.launch()
```

requirements.txt

joblib

numpy

scikit-learn

Diabetes prediction App

Enter patient info to predict diabetes risk

Pregnancies

6

Glucose

148

Blood Pressure

120

Skin Thickness

50

Insulin

124

BMI

23

Probability

0.5416666666666666

Testing and Sharing Your Model

- Use the Gradio web UI to test the deployed model
- Enter patient data and get predictions
- Share the Hugging Face Space link with others

Thank you