



A Machine Learning Model for Diabetes

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

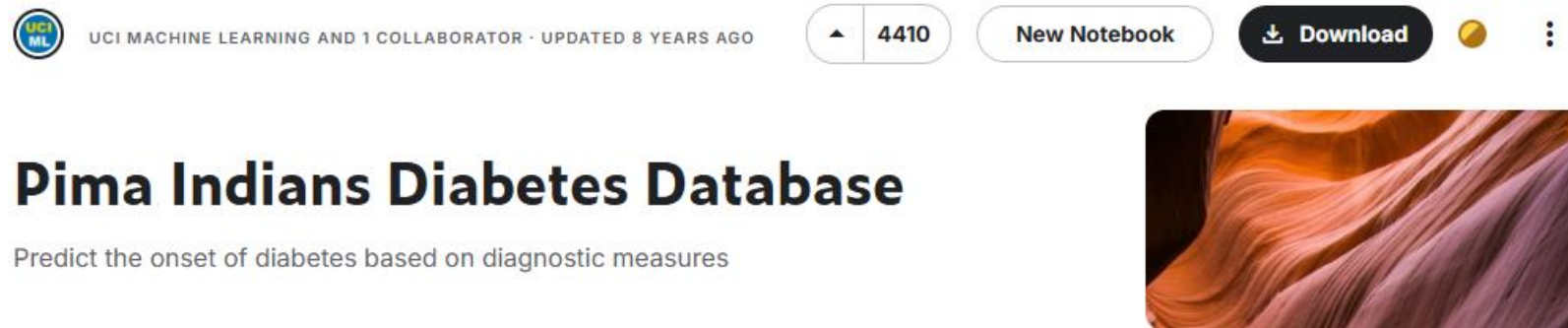


Agenda

- ⇒ 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

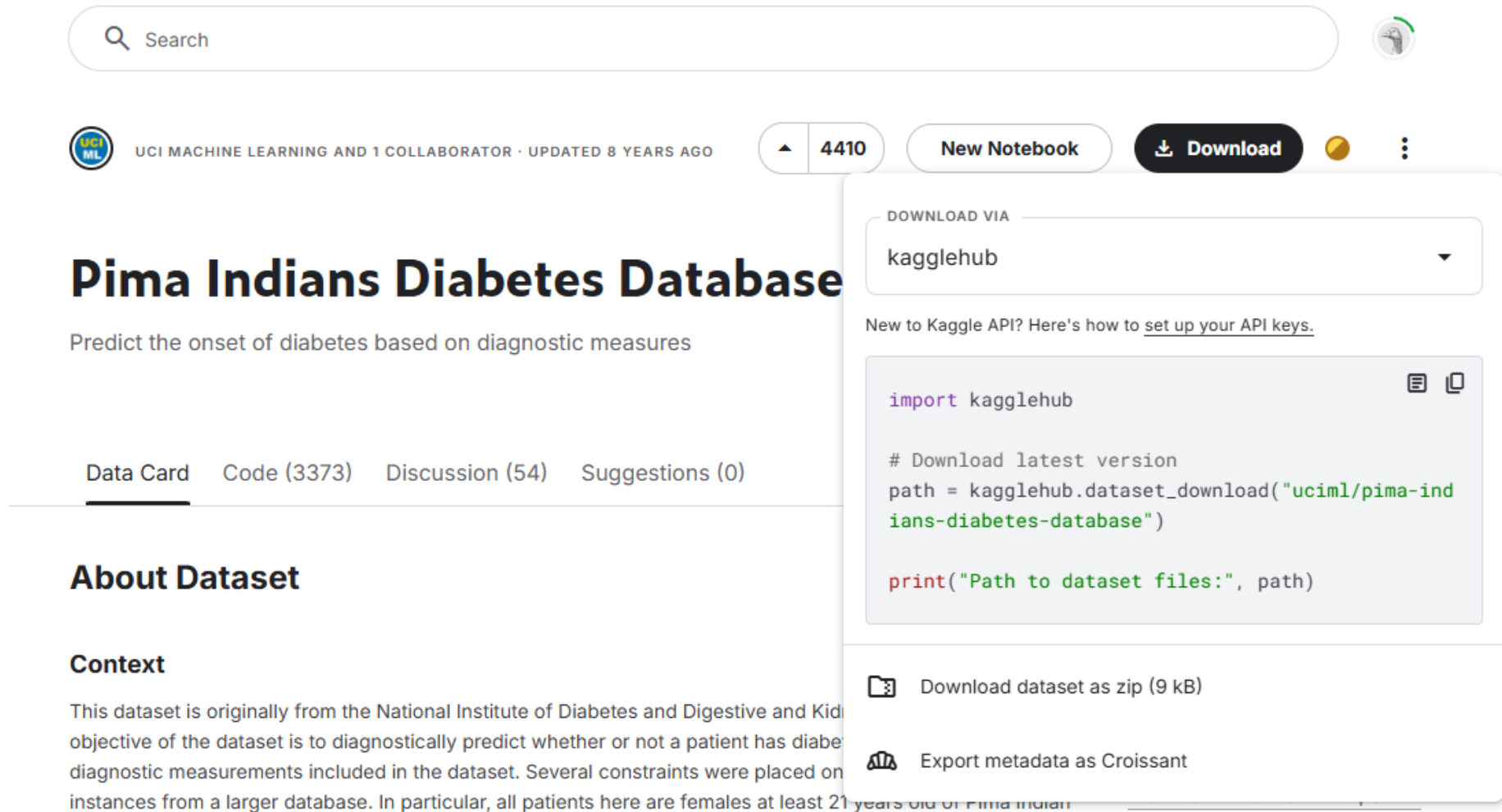
Pima Diabetes Dataset

- 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



The screenshot shows the Kaggle interface for the 'Pima Indians Diabetes Database'. At the top, there is a search bar and a navigation bar with the UCI ML logo, the text 'UCI MACHINE LEARNING AND 1 COLLABORATOR · UPDATED 8 YEARS AGO', a '4410' badge, a 'New Notebook' button, a 'Download' button, and a user profile icon. The main title is 'Pima Indians Diabetes Database' with the subtitle 'Predict the onset of diabetes based on diagnostic measures'. Below this are tabs for 'Data Card', 'Code (3373)', 'Discussion (54)', and 'Suggestions (0)'. The 'About Dataset' section is visible, starting with a 'Context' heading. A dropdown menu is open over the 'Download' button, showing 'DOWNLOAD VIA kagglehub' and a link to 'set up your API keys'. Below this, a code block contains Python code for downloading the dataset using kagglehub. At the bottom of the menu, there are two options: 'Download dataset as zip (9 kB)' and 'Export metadata as Croissant'.

Search

UCI ML UCI MACHINE LEARNING AND 1 COLLABORATOR · UPDATED 8 YEARS AGO 4410 New Notebook Download

Pima Indians Diabetes Database

Predict the onset of diabetes based on diagnostic measures

Data Card Code (3373) Discussion (54) Suggestions (0)

About Dataset

Context

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on some diagnostic measurements included in the dataset. Several constraints were placed on the selection of the instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian

DOWNLOAD VIA kagglehub

New to Kaggle API? Here's how to [set up your API keys](#).

```
import kagglehub

# Download latest version
path = kagglehub.dataset_download("uciml/pima-indians-diabetes-database")

print("Path to dataset files:", path)
```


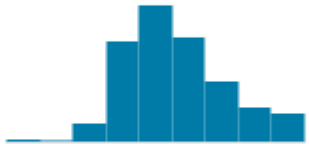


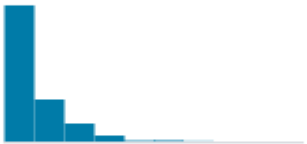

Download dataset as zip (9 kB)

Export metadata as Croissant

<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)
					
017	0199	0122	099	0846	0
6	148	72	35	0	33.6
1	85	66	29	0	26.6
8	183	64	0	0	23.3
1	00	66	00	0.1	20.1

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>

auto **TRAIN**

Create powerful AI models without code


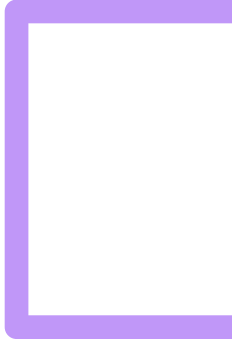
A new way to automatically train, evaluate and deploy state-of-the-art Machine Learning models.

Create new project

or read the [documentation](#)



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]

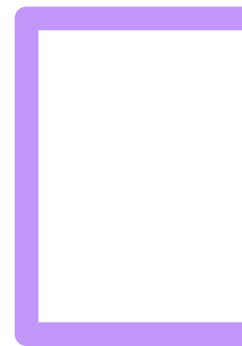
demo = gr.Interface(
    fn=myfunc,
    inputs=[
        gr.Number(label="Pregnancies"),
        gr.Number(label="Glucose"),
        gr.Number(label="Blood Pressure"),
        gr.Number(label="Skin Thickness"),
        gr.Number(label="Insulin"),
        gr.Number(label="BMI"),
        gr.Number(label="Diabetes 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