

```

import pandas as pd
import numpy as np
from sklearn.linear_model import LogisticRegression

df=pd.read_excel("/content/drive/MyDrive/diabetes (1).xlsx")
df.head()

{"summary": {
    "name": "df",
    "rows": 768,
    "fields": [
        {
            "column": "preg",
            "properties": {
                "dtype": "number",
                "std": 3,
                "min": 0,
                "max": 17,
                "num_unique_values": 17,
                "samples": [6, 1, 3]
            },
            "semantic_type": "\\"",
            "description": """
                column: plas
                properties: {
                    dtype: "number",
                    std: 31,
                    min: 0,
                    max: 199,
                    num_unique_values: 136,
                    samples: [151, 101, 112]
                }
            """
        },
        {
            "column": "pres",
            "properties": {
                "dtype": "number",
                "std": 19,
                "min": 0,
                "max": 122,
                "num_unique_values": 47,
                "samples": [86, 46, 85]
            },
            "semantic_type": "\\"",
            "description": """
                column: pres
                properties: {
                    dtype: "number",
                    std: 19,
                    min: 0,
                    max: 122,
                    num_unique_values: 47,
                    samples: [86, 46, 85]
                }
            """
        },
        {
            "column": "skin",
            "properties": {
                "dtype": "number",
                "std": 15,
                "min": 0,
                "max": 99,
                "num_unique_values": 51,
                "samples": [7, 12, 48]
            },
            "semantic_type": "\\"",
            "description": """
                column: skin
                properties: {
                    dtype: "number",
                    std: 15,
                    min: 0,
                    max: 99,
                    num_unique_values: 51,
                    samples: [7, 12, 48]
                }
            """
        },
        {
            "column": "insu",
            "properties": {
                "dtype": "number",
                "std": 115,
                "min": 0,
                "max": 846,
                "num_unique_values": 186,
                "samples": [52, 41, 183]
            },
            "semantic_type": "\\"",
            "description": """
                column: insu
                properties: {
                    dtype: "number",
                    std: 115,
                    min: 0,
                    max: 846,
                    num_unique_values: 186,
                    samples: [52, 41, 183]
                }
            """
        },
        {
            "column": "pedi",
            "properties": {
                "dtype": "number",
                "std": 7.8841603203754405,
                "min": 0.0,
                "max": 67.1,
                "num_unique_values": 248,
                "samples": [19.9, 31.0, 38.1]
            },
            "semantic_type": "\\"",
            "description": """
                column: pedi
                properties: {
                    dtype: "number",
                    std: 7.8841603203754405,
                    min: 0.0,
                    max: 67.1,
                    num_unique_values: 248,
                    samples: [19.9, 31.0, 38.1]
                }
            """
        },
        {
            "column": "age",
            "properties": {
                "dtype": "number",
                "std": 0.33132859501277484,
                "min": 0.078,
                "max": 2.42,
                "num_unique_values": 517,
                "samples": [1.731, 0.426, 0.138]
            },
            "semantic_type": "\\"",
            "description": """
                column: age
                properties: {
                    dtype: "number",
                    std: 0.33132859501277484,
                    min: 0.078,
                    max: 2.42,
                    num_unique_values: 517,
                    samples: [1.731, 0.426, 0.138]
                }
            """
        },
        {
            "column": "target",
            "properties": {
                "dtype": "number",
                "std": 11,
                "min": 21,
                "max": 81,
                "num_unique_values": 52,
                "samples": [1]
            },
            "semantic_type": "\\"",
            "description": """
                column: target
                properties: {
                    dtype: "number",
                    std: 11,
                    min: 21,
                    max: 81,
                    num_unique_values: 52,
                    samples: [1]
                }
            """
        }
    ]
}

```

```
[\"n      60,\n      47,\n      72\n      ],\n      \"semantic_type\": \"\",\n      \"description\": \"\n      \",\n      \"column\": \"class\",\n      \"properties\": {\n      \"dtype\": \"category\",\n      \"num_unique_values\": 2,\n      \"samples\": [\n          \"tested_negative\",\n          \"tested_positive\"\n      ],\n      \"semantic_type\": \"\",\n      \"description\": \"\"\n    }\n},\n\"type\":\"dataframe\", \"variable_name\":\"df\"}\n\ndf.isnull().sum()\n\npreg      0\nplas      0\npres      0\nskin      0\ninsu      0\nmass      0\npedi      0\nage       0\nclass     0\ndtype: int64\n\ninp=df[["age","mass","insu","plas"]]\nout=df["class"]\n\nLogr=LogisticRegression()\n\nLogr.fit(inp,out)\n\nLogisticRegression()\n\nage=int(input("Enter the age:"))\nmass=int(input("Enter the mass:"))\ninsulin=int(input("Enter the insulin level:"))\nplasma=int(input("Enter the plasma level:"))\npred=Logr.predict([[age, mass, insulin, plasma]])\nprint(pred)\n\nEnter the age:23\nEnter the mass:78\nEnter the insulin level:99\nEnter the plasma level:32\n['tested_negative']\n\n/usr/local/lib/python3.12/dist-packages/sklearn/utils/\nvalidation.py:2739: UserWarning: X does not have valid feature names,\nbut LogisticRegression was fitted with feature names\n  warnings.warn(\n\nLogr.score(inp,out)\n\n0.7669270833333334
```