


# Diabetes Prediction Using ML





## Problem Specification

- The most usual and conventional method for diagnosis and detection of diabetic retinopathy is by using human fundus images or retinal images.
- In our study, we focus on prediction of DR using health records of the diabetic patients.
- By using machine learning techniques, knowledge is acquired through these records, containing numerical values, to predict whether the patient is having DR or not.
- For this prediction of DR different classification algorithms (Support Vector Machine, K nearest neighbor, bagged trees, Logistic Regression) have been used.




## DIABETES Types

- Type 1 diabetes
- Type 2 diabetes

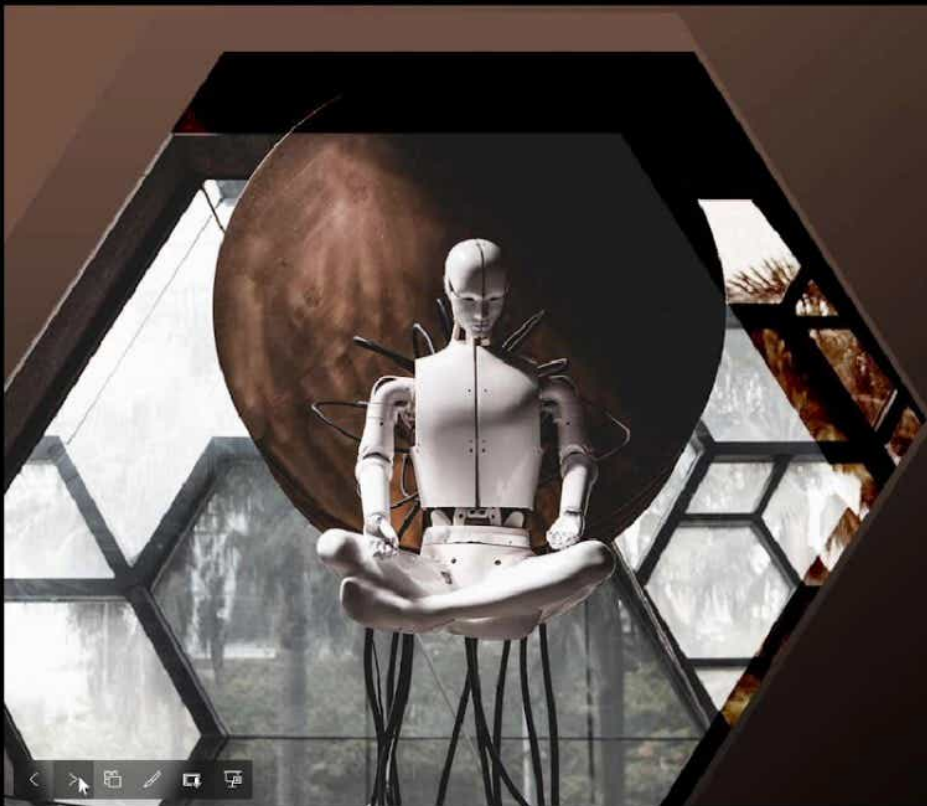
## Effects of DIABETES

- Heart Disease
- stroke
- kidney damage
- nerve damage



# Machine Learning Model

- Importing the Dataset
- Filling the Missing Values
- Exploratory Data Analysis
- Feature Engineering
- Implementing Machine Learning Models
- Making Predictions




## Lead Scoring Case Study



## Problem Specification

- An education company named X Education sells online courses to industry professionals. On any given day, many professionals who are interested in the courses land on their website and browse for courses.
- The company requires you to build a model wherein you need to assign a lead score to each of the leads such that the customers with higher lead score
- have a higher conversion chance and the customers with lower lead score have a lower conversion chance.
- The CEO, in particular, has given a ballpark of the target lead conversion rate to be around 80%.





# Machine Learning Model

- The solution is divided into the following sections:
  - Creation of Virtual Environment
  - Data Importing
  - Data Inspection
  - Data Cleaning
  - Building ML Model
  - ROC and Final Prediction



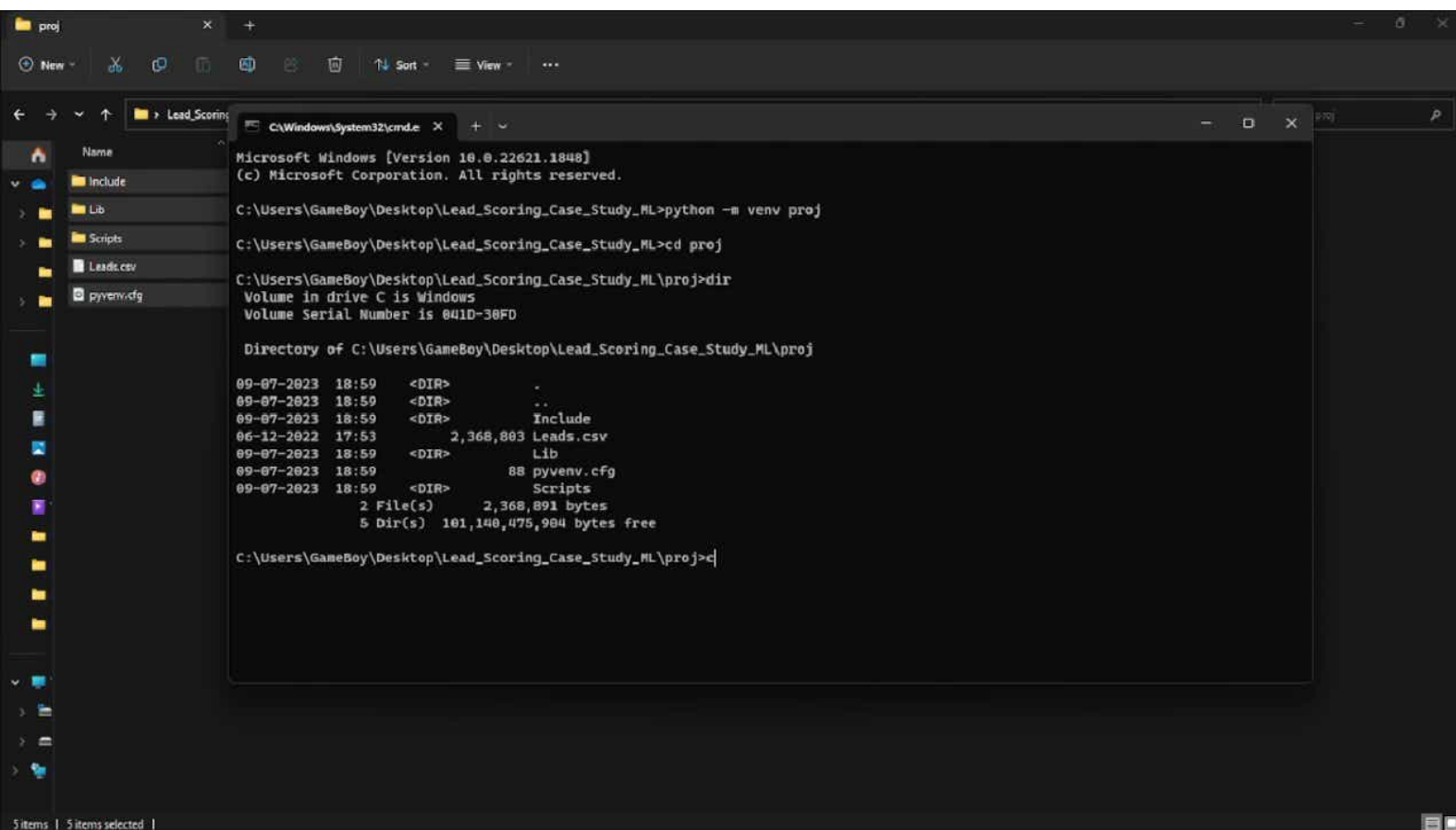
## Terminologies

- Recursive Feature Elimination(RFE)
- Generalized Linear Model (GLM)
- Variance Inflation Factor(VIF)
- Receiver Operating Characteristic (ROC)
- Optimal Cutoof point
- Precision
- Recall





Let's get coding



Select C:\Windows\System32\cmd.exe

C:\Users\rgukt\Desktop\Lead\_Scoring\_casestudy\_ML\proj>Scripts\activate

```
(proj) C:\Users\rgukt\Desktop\Lead_Scoring_casestudy_ML\proj>pip install ipykernel
Collecting ipykernel
  Using cached ipykernel-6.29.4-py3-none-any.whl.metadata (6.3 kB)
Collecting comm>=0.1.1 (from ipykernel)
  Using cached comm-0.2.2-py3-none-any.whl.metadata (3.7 kB)
Collecting debugpy>=1.6.5 (from ipykernel)
  Using cached debugpy-1.8.1-cp312-cp312-win_amd64.whl.metadata (1.1 kB)
Collecting ipython>=7.23.1 (from ipykernel)
  Using cached ipython-8.24.0-py3-none-any.whl.metadata (4.9 kB)
Collecting jupyter-client>=6.1.12 (from ipykernel)
  Downloading jupyter_client-8.6.2-py3-none-any.whl.metadata (8.3 kB)
Collecting jupyter-core!>=5.0.*,>=4.12 (from ipykernel)
  Using cached jupyter_core-5.7.2-py3-none-any.whl.metadata (3.4 kB)
Collecting matplotlib-inline>=0.1 (from ipykernel)
  Using cached matplotlib_inline-0.1.7-py3-none-any.whl.metadata (3.9 kB)
Collecting nest-asyncio (from ipykernel)
  Using cached nest_asyncio-1.6.0-py3-none-any.whl.metadata (2.8 kB)
Collecting packaging (from ipykernel)
  Using cached packaging-24.0-py3-none-any.whl.metadata (3.2 kB)
Collecting psutil (from ipykernel)
  Using cached psutil-5.9.8-cp37-abi3-win_amd64.whl.metadata (22 kB)
Collecting pyzmq>=24 (from ipykernel)
  Using cached pyzmq-26.0.3-cp312-cp312-win_amd64.whl.metadata (6.1 kB)
Collecting tornado>=6.1 (from ipykernel)
  Using cached tornado-6.4-cp38-abi3-win_amd64.whl.metadata (2.6 kB)
Collecting traitlets>=5.4.0 (from ipykernel)
  Using cached traitlets-5.14.3-py3-none-any.whl.metadata (10 kB)
Collecting decorator (from ipython>=7.23.1->ipykernel)
  Using cached decorator-5.1.1-py3-none-any.whl.metadata (4.0 kB)
Collecting jedi>=0.16 (from ipython>=7.23.1->ipykernel)
  Using cached jedi-0.19.1-py2.py3-none-any.whl.metadata (22 kB)
Collecting prompt_toolkit<3.1.0,>=3.0.41 (from ipython>=7.23.1->ipykernel)
  Using cached prompt_toolkit-3.0.43-py3-none-any.whl.metadata (6.5 kB)
Collecting pygments>=2.4.0 (from ipython>=7.23.1->ipykernel)
  Using cached pygments-2.18.0-py3-none-any.whl.metadata (2.5 kB)
Collecting stack_data (from ipython>=7.23.1->ipykernel)
  Using cached stack_data-0.6.3-py3-none-any.whl.metadata (18 kB)
Collecting colorama (from ipython>=7.23.1->ipykernel)
  Using cached colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB)
Collecting python-dateutil>=2.8.2 (from jupyter-client>=6.1.12->ipykernel)
  Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl.metadata (8.4 kB)
```

Select C:\Windows\System32\cmd.exe

```
Using cached psutil-5.9.8-cp37-abi3-win_amd64.whl (255 kB)
Using cached jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
Downloading platformdirs-4.2.2-py3-none-any.whl (18 kB)
Using cached prompt_toolkit-3.0.43-py3-none-any.whl (386 kB)
Using cached pygments-2.18.0-py3-none-any.whl (1.2 MB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Using cached pywin32-306-cp312-cp312-win_amd64.whl (9.2 MB)
Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Using cached decorator-5.1.1-py3-none-any.whl (9.1 kB)
Using cached stack_data-0.6.3-py3-none-any.whl (24 kB)
Using cached asttokens-2.4.1-py2.py3-none-any.whl (27 kB)
Using cached executing-2.0.1-py2.py3-none-any.whl (24 kB)
Using cached parso-0.8.4-py2.py3-none-any.whl (103 kB)
Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Using cached pure_eval-0.2.2-py3-none-any.whl (11 kB)
Using cached wcwidth-0.2.13-py2.py3-none-any.whl (34 kB)
Installing collected packages: wcwidth, pywin32, pure-eval, traitlets, tornado, six, pyzmq, pygments, psutil, prompt-toolkit, platformdirs, parso, packaging, nest-asyncio, executing, decorator, debugpy, colorama, python-dateutil, matplotlib-inline, jupyter-core, jedi, comm, asttokens, stack-data, jupyter-client, ipython, ipykernel
Successfully installed asttokens-2.4.1 colorama-0.4.6 comm-0.2.2 debugpy-1.8.1 decorator-5.1.1 executing-2.0.1 ipykernel-6.29.4 ipython-8.24.0 jedi-0.19.1 jupyter-client-8.6.2 jupyter-core-5.7.2 matplotlib-inline-0.1.7 nest-asyncio-1.6.0 packaging-24.0 parso-0.8.4 platformdirs-4.2.2 prompt-toolkit-3.0.43 psutil-5.9.8 pure-eval-0.2.2 pygments-2.18.0 python-dateutil-2.9.0.post0 pywin32-306 pyzmq-26.0.3 six-1.16.0 stack-data-0.6.3 tornado-6.4 traitlets-5.14.3 wcwidth-0.2.13
```

```
(proj) C:\Users\rngukt\Desktop\Lead_Scoring_casestudy_ML\proj>python kernel install --user --name=coincidentproject
```

```
Installed kernelspec coincidentproject in C:\Users\rngukt\AppData\Roaming\jupyter\kernels\coincidentproject
```

```
(proj) C:\Users\rngukt\Desktop\Lead_Scoring_casestudy_ML\proj>_
```

```
C:\Windows\System32\cmd.exe
Using cached psutil-5.9.8-cp37-abi3-win_amd64.whl (255 kB)
Using cached jedi-0.19.1-py2.py3-none-any.whl (1.6 MB)
Downloading platformdirs-4.2.2-py3-none-any.whl (18 kB)
Using cached prompt_toolkit-3.0.43-py3-none-any.whl (386 kB)
Using cached pygments-2.18.0-py3-none-any.whl (1.2 MB)
Using cached python_dateutil-2.9.0.post0-py2.py3-none-any.whl (229 kB)
Using cached pywin32-306-cp312-cp312-win_amd64.whl (9.2 MB)
Using cached colorama-0.4.6-py2.py3-none-any.whl (25 kB)
Using cached decorator-5.1.1-py3-none-any.whl (9.1 kB)
Using cached stack_data-0.6.3-py3-none-any.whl (24 kB)
Using cached asttokens-2.4.1-py2.py3-none-any.whl (27 kB)
Using cached executing-2.0.1-py2.py3-none-any.whl (24 kB)
Using cached parso-0.8.4-py2.py3-none-any.whl (103 kB)
Using cached six-1.16.0-py2.py3-none-any.whl (11 kB)
Using cached pure_eval-0.2.2-py3-none-any.whl (11 kB)
Using cached wcwidth-0.2.13-py2.py3-none-any.whl (34 kB)
Installing collected packages: wcwidth, pywin32, pure-eval, traitlets, tornado, six, pyzmq, pygments, psutil, prompt-toolkit, platformdirs, parso, packaging, nest-asyncio, executing, decorator, debugpy, colorama, python-dateutil, matplotlib-inline, jupyter-core, jedi, comm, asttokens, stack-data, jupyter-client, ipython, ipykernel
Successfully installed asttokens-2.4.1 colorama-0.4.6 comm-0.2.2 debugpy-1.8.1 decorator-5.1.1 executing-2.0.1 ipykernel-6.29.4 ipython-8.24.0 jedi-0.19.1 jupyter-client-8.6.2 jupyter-core-5.7.2 matplotlib-inline-0.1.7 nest-asyncio-1.6.0 packaging-24.0 parso-0.8.4 platformdirs-4.2.2 prompt-toolkit-3.0.43 psutil-5.9.8 pure-eval-0.2.2 pygments-2.18.0 python-dateutil-2.9.0.post0 pywin32-306 pyzmq-26.0.3 six-1.16.0 stack-data-0.6.3 tornado-6.4 traitlets-5.14.3 wcwidth-0.2.13

(proj) C:\Users\rngukt\Desktop\Lead_Scoring_casestudy_ML\proj>ipython kernel install --user --name=coincidentproject
Installed kernelspec coincidentproject in C:\Users\rngukt\AppData\Roaming\jupyter\kernels\coincidentproject

(proj) C:\Users\rngukt\Desktop\Lead_Scoring_casestudy_ML\proj>jupyter notebook_
```

Home Page - Select or create a notebook

Lead\_score - Jupyter Notebook

Leads.csv - Jupyter Text Editor

How to install 'warnings' package

localhost:8888/notebooks/Lead\_score.ipynb

Guest

jupyter

Lead\_score

Last Checkpoint: 11 minutes ago (unsaved changes)

Logout

FileEditViewInsertCellKernelWidgetsHelp

TrustedML

RunCode

## Lead Scoring Case Study

```
In [6]: #pip install numpy
#pip install pandas
#pip install pytest-warnings

In [7]: # Import the warnings
import warnings
warnings.filterwarnings("ignore")

In [8]: # Import libs for analysis, data handling
import numpy as np
import pandas as pd
pd.set_option("display.max_columns", None)
```

### Step 1: Importing the Dataset

```
In [ ]: leads = pd.read_csv("Leads.csv")
leads.head()
```