Project

"Predicting Osteoporosis Risk"

Submitted By,

Mavani Krishnkumar Manjibhai

INDEX

1. Project Introduction

1.1. Domain

The domain of this project is **Healthcare and Medical Data Analysis**, with a specific focus on **predicting osteoporosis risk** using **machine learning techniques** for early diagnosis and enhanced patient care.

This domain encompasses the analysis of patient data to assess the likelihood of osteoporosis, utilizing machine learning models to support healthcare professionals in making more accurate and timely decisions for patient management and intervention.

1.2. Definition

The project aims to develop a machine learning model for predicting osteoporosis risk in patients by analyzing various characteristics and medical history. The model will classify patients into two categories: those at risk of osteoporosis and those not at risk, using historical labeled data for training.

1.3. Problem

The problem is a **supervised classification** problem. The goal is to predict whether a patient is at risk of osteoporosis, which is a binary (categorical) outcome the patient either has osteoporosis or does not.

This problem is categorized under supervised learning since the model is trained on labeled data, which consists of historical patient information with known outcomes (i.e., whether the patient has osteoporosis). The model learns from this data to generate accurate predictions for new, unseen cases.

1.4. Benefits

- **Early Detection:** The model will help identify high-risk patients, allowing for timely intervention and prevention strategies.
- **Personalized Care:** By predicting osteoporosis risk accurately, healthcare providers can tailor treatment plans to individual patient needs.
- **Resource Optimization:** Understanding risk factors can assist in better allocation of healthcare resources for osteoporosis prevention and management.

1.5. Issues

1. **Missing Data:** Incomplete patient records may lead to missing values in critical columns (e.g., Alcohol Consumption, Medical Conditions, Medications), which could affect the accuracy and reliability of the model.
2. **Class Imbalance:** The dataset may contain more non-osteoporosis cases than osteoporosis cases, leading to a class imbalance problem. This could result in biased predictions towards the majority class, making the model less effective in detecting high-risk patients.
3. **Data Privacy Concerns:** Handling sensitive healthcare data such as patient medical history, body weight, and other personal information could raise privacy concerns and compliance issues with regulations.
4. **Multicollinearity:** Some of the features (e.g., Body Weight, Hormonal Changes) could be highly correlated with each other, leading to multicollinearity. This can make it harder for the model to assess the individual effect of these variables on the prediction.
5. **Data Quality and Noise:** If there are inconsistencies or errors in the dataset, such as incorrect labels or conflicting data (e.g., conflicting family history and prior fractures), it could adversely impact model training and prediction accuracy.

1.6. Challenges

1. **Feature Engineering:** Selecting the right set of features and transforming them appropriately (e.g., encoding categorical variables like Gender and Smoking status) can be challenging. Feature selection techniques and domain knowledge are required to ensure the most informative features are included.
2. **Model Selection and Tuning:** Identifying the best classification algorithm for the task (Logistic Regression, Random Forest, etc.) and tuning the model's hyperparameters using methods like GridSearchCV can be time-consuming and computationally expensive.
3. **Model Interpretability:** Given the importance of interpretability in healthcare applications, making sure the model's predictions are understandable by healthcare professionals is a challenge. Ensuring transparency and explainability of complex models, like Random Forests or Support Vector Machines, is crucial.

4. **Handling Imbalanced Classes:** Addressing the class imbalance issue, where osteoporosis cases are fewer, requires techniques like resampling or adjusting class weights to avoid biased predictions.
5. **Model Overfitting or Underfitting**: A model that is overly complex may overfit the training data, causing poor generalization to new, unseen instances. Conversely, a model that is too simplistic may underfit the data, failing to capture essential patterns and relationships.

1.7. Objective

The objective of this project is to develop a machine learning model that predicts the risk of osteoporosis in patients based on their characteristics and medical history. The model will classify patients as either having osteoporosis or not, utilizing historical labeled data for training.

1.8. Summary

This project aims to develop a machine learning model to predict osteoporosis risk in patients based on their medical history and characteristics. Key challenges include ensuring data quality, selecting relevant features, and addressing model overfitting or underfitting. The project also focuses on handling class imbalance and ensuring model interpretability for healthcare professionals.

2. Design Details

2.1. Architecture

The architecture of the Prediction Osteoporosis Risk system adopts a structured, modular approach to ensure an efficient and systematic workflow for data analysis and model development. The key stages include:

2.1.1. Data Flow Architecture

1. **Import Necessary Libraries:**
   - **Objective:** Import essential libraries for data manipulation, preprocessing, visualization, model training, and evaluation.
   - **Libraries:** pandas, numpy, matplotlib, seaborn, sklearn, warnings, etc.
2. **Load the Dataset:**
   - **Objective:** Load the dataset into a pandas DataFrame and create a copy for further exploration and processing.
   - **Implementation:** Use pd.read_csv() to load the dataset osteoporosis.csv and store it in the old_df variable. Copy data from old_df to df for further processing.
3. **Data Exploration:**
   - **Objective**: Understand the data structure in the dataset.
   - **Tasks**:
     I Display shape of the dataset (df.shape).
     II Show the first 5 rows (df.head()).
     III Summarize data (df.info()).
4. **Data Preprocessing:**
   - **Objective**: Clean and preprocess the data to make it suitable for machine learning.
   - **Tasks**:
     I Handle Missing Values
       - Identify columns with missing values and their count.
       - Visualize missing values using a heatmap.
       - Fill missing categorical values with 'None'.
       - Recheck missing values and visualize again using a heatmap.

     II Handle Duplicates (Identify and count duplicate rows).
     III Identify Outliers in Every Columns.

IV Identify Unique columns in the data.

V Identify categorical features & numerical features columns.

VI Perform Statistical Operation in numeric columns

2.1.2. Exploratory Data Analysis (EDA) & Visualization

1. Osteoporosis
2. Age & Age distribution Osteoporosis
3. Correlation Matrix for Numerical Variables (Age, Osteoporosis)
4. Gender & Osteoporosis distribution Gender
5. Hormonal Changes & Osteoporosis by Hormonal Changes
6. Family History & Osteoporosis by Family History
7. Race/Ethnicity & Osteoporosis by Race/Ethnicity
8. Body Weight & Osteoporosis by Body Weight
9. Calcium Intake & Osteoporosis by Calcium Intake
10. Vitamin D Intake & Osteoporosis by Vitamin D Intake
11. Physical Activity & Osteoporosis by Physical Activity
12. Smoking & Osteoporosis by Smoking
13. Alcohol Consumption & Osteoporosis by Alcohol Consumption
14. Medical Condition & Osteoporosis by Medical Condition
15. Medications & Osteoporosis by Medications
16. Prior Fractures & Osteoporosis by Prior Fractures

2.1.3. Feature Engineering

1. Apply Label Encoding Techniques in the categorical features.
2. Define input features (X) and target variable (y).
3. Split data into training and testing sets using train_test_split().
4. Standardize the features using StandardScaler.

2.1.4. Model Flow Architecture

1. **Model Selection**
    - **Objective:** Train and evaluate multiple models to determine the best one for the classification task.
    - **Models:**

      I Random Forest Classifier

      II Decision Tree Classifier

      III Logistic Regression

IV SVC (Support Vector Classifier)

V Random Forest Classifier with GridSearchCV

VI Decision Tree Classifier with GridSearchCV

VII Logistic Regression with GridSearchCV

VIII SVC (Support Vector Classifier) with GridSearchCV

2. **Training**
   - **Objective:** Train each of the selected models on the training dataset.
     - Use model-specific fit methods (e.g., model.fit()).
3. **Prediction**
   - **Objective:** Use each trained model to make predictions on the test dataset.
     - Use model-specific predict methods (e.g., model.predict()).
4. **Performance Evaluation**
   - **Objective:** Evaluate each model's performance using various metrics.
   - **Metrics:**

     I Accuracy: accuracy_score().

     II Confusion Matrix: confusion_matrix().

     III Classification Report: classification_report().

2.1.5. Visualization Flow Architecture

1. **Accuracy Visualizations**
   - **Objective:** Visualize the accuracies of all models in a bar chart for comparison.
   - **Tasks:**

     I Create a dictionary to store model names and their accuracies.

     II Generate random colors for each model.

     III Plot accuracies in a bar chart using matplotlib.pyplot.bar().
2. **Confusion Matrix Visualizations**
   - **Objective:** Visualize the accuracies of all models in a heatmap for comparison.
   - **Visualizations:**

     I Create a dictionary to store model names and their confusion matrix.

     II A confusion matrix into a heatmap using seaborn.heatmap().

2.1.6. Model Deployment

1. Label Encoder
2. Scaler Model
3. Random Forest Classifier Model
4. Decision Tree Classifier Model
5. Logistic Regression Model
6. Support Vector Classifier Model
7. GirdSearchCV Random Forest Classifier Model
8. GirdSearchCV Decision Tree Classifier Model
9. GirdSearchCV Logistic Regression Model
10. GirdSearchCV Support Vector Classifier Model

2.1.7. Complete Flow Diagram

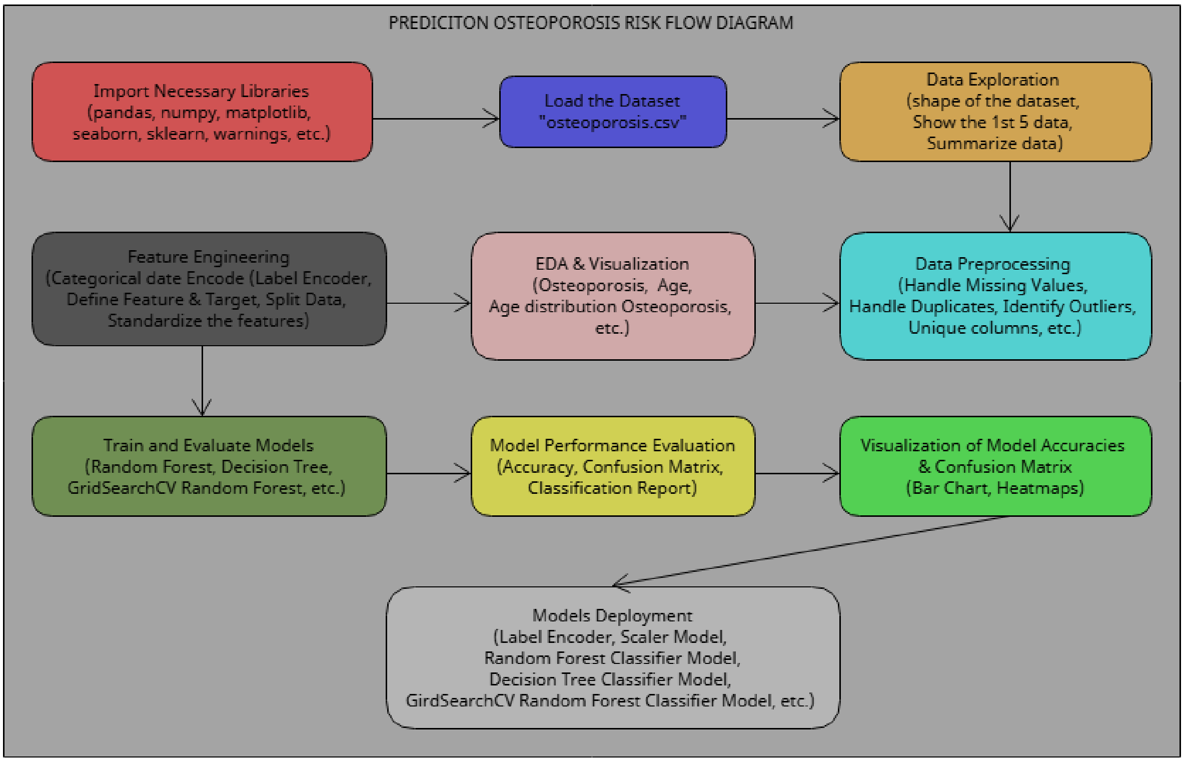


Fig 1 - Complete Model Flow Diagram

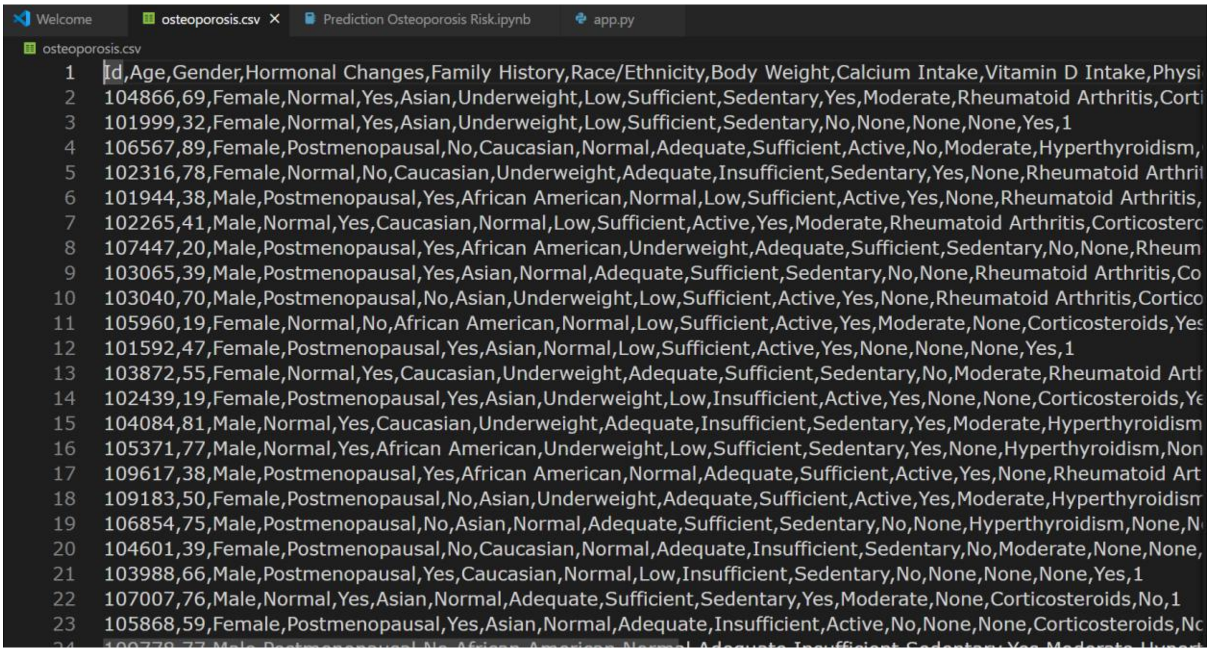
2.1.8. Implementation Workflow

1. **Import Libraries**: Load essential Python libraries (Pandas, NumPy, Matplotlib, Seaborn, Sklearn, etc.).
2. **Load Dataset**: Read "osteoporosis.csv" into a DataFrame and make a copy for safety.
3. **Data Exploration**: Display dataset shape, first 5 rows, column details, types, and non-null values.
4. **Data Preprocessing**: Handle missing values (fill categorical with "None"), detect and remove duplicates, identify unique categorical and numerical columns, and label encode categorical features.
5. **Feature Engineering**: Define input features (X) and target variable (y), split data into training/testing sets (80-20 split), and standardize numerical features.
6. **Model Training & Evaluation**: Train models without hyperparameter tuning, evaluate performance using accuracy, confusion matrix, and classification report, apply GridSearchCV for model optimization, and select the best-performing model.
7. **Model Deployment**: Load the trained model and scaler, predict osteoporosis risk for new input data, and deploy the model for real-time predictions.

2.2. Dataset Details

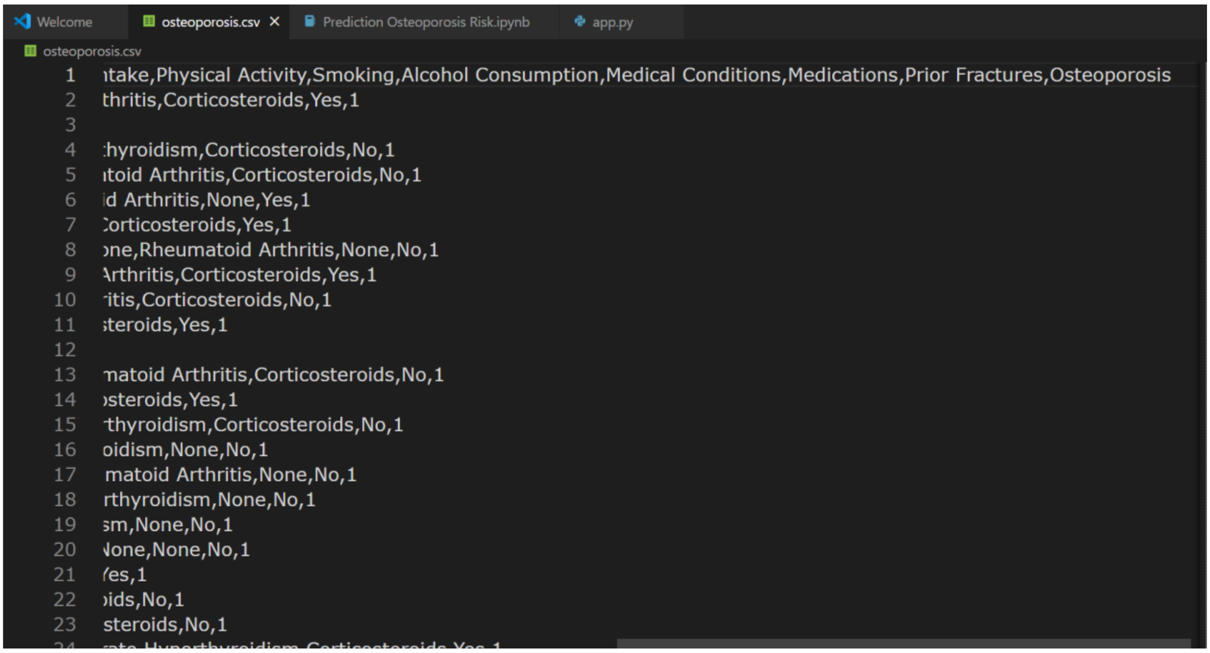
Field Name	**Description**
ID	Unique identifier for each patient
Age	Age of the patient
Gender	Gender of the patient
Hormonal Changes	Whether the patient has undergone hormonal changes
Family History	Whether the patient has a family history of osteoporosis
Race/Ethnicity	Race or ethnicity of the patient
Body Weight	Weight details of the patient
Calcium	Calcium levels in the patient's body
Vitamin D	Vitamin D levels in the patient's body
Physical Activity	Physical activity details of the patient
Smoking	Whether the patient smokes
Alcohol Consumption	Whether the patient consumes alcohol
Medical Conditions	Medical conditions of the patient
Medication	Medication details of the patient
Prior Fracture	Whether the patient has had a prior fracture
Osteoporosis	Whether the patient has osteoporosis

Table 1 - Dataset Details



```
Welcome    osteoporosis.csv ×    Prediction Osteoporosis Risk.ipynb    app.py
osteoporosis.csv
Id,Age,Gender,Hormonal Changes,Family History,Race/Ethnicity,Body Weight,Calcium Intake,Vitamin D Intake,Physi
104866,69,Female,Normal,Yes,Asian,Underweight,Low,Sufficient,Sedentary,Yes,Moderate,Rheumatoid Arthritis,Corti
101999,32,Female,Normal,Yes,Asian,Underweight,Low,Sufficient,Sedentary,No,None,None,None,Yes,1
106567,89,Female,Postmenopausal,No,Caucasian,Normal,Adequate,Sufficient,Active,No,Moderate,Hyperthyroidism,
102316,78,Female,Normal,No,Caucasian,Underweight,Adequate,Insufficient,Sedentary,Yes,None,Rheumatoid Arthri
101944,38,Male,Postmenopausal,Yes,African American,Normal,Low,Sufficient,Active,Yes,None,Rheumatoid Arthritis,
102265,41,Male,Normal,Yes,Caucasian,Normal,Low,Sufficient,Active,Yes,Moderate,Rheumatoid Arthritis,Corticosterc
107447,20,Male,Postmenopausal,Yes,African American,Underweight,Adequate,Sufficient,Sedentary,No,None,Rheum
103065,39,Male,Postmenopausal,Yes,Asian,Normal,Adequate,Sufficient,Sedentary,No,None,Rheumatoid Arthritis,Co
103040,70,Male,Postmenopausal,No,Asian,Underweight,Low,Sufficient,Active,Yes,None,Rheumatoid Arthritis,Cortico
105960,19,Female,Normal,No,African American,Normal,Low,Sufficient,Active,Yes,Moderate,None,Corticosteroids,Yes
101592,47,Female,Postmenopausal,Yes,Asian,Normal,Low,Sufficient,Active,Yes,None,None,None,Yes,1
103872,55,Female,Normal,Yes,Caucasian,Underweight,Adequate,Sufficient,Sedentary,No,Moderate,Rheumatoid Arth
102439,19,Female,Postmenopausal,Yes,Asian,Underweight,Low,Insufficient,Active,Yes,None,None,Corticosteroids,Ye
104084,81,Male,Normal,Yes,Caucasian,Underweight,Adequate,Insufficient,Sedentary,Yes,Moderate,Hyperthyroidism
105371,77,Male,Normal,Yes,African American,Underweight,Low,Sufficient,Sedentary,Yes,None,Hyperthyroidism,Non
109617,38,Male,Postmenopausal,Yes,African American,Normal,Adequate,Sufficient,Active,Yes,None,Rheumatoid Art
109183,50,Female,Postmenopausal,No,Asian,Underweight,Adequate,Sufficient,Active,Yes,Moderate,Hyperthyroidisn
106854,75,Male,Postmenopausal,No,Asian,Normal,Adequate,Sufficient,Sedentary,No,None,Hyperthyroidism,None,N
104601,39,Female,Postmenopausal,No,Caucasian,Normal,Adequate,Insufficient,Sedentary,No,Moderate,None,None,
103988,66,Male,Postmenopausal,Yes,Caucasian,Normal,Low,Insufficient,Sedentary,No,None,None,None,Yes,1
107007,76,Male,Normal,Yes,Asian,Normal,Adequate,Sufficient,Sedentary,Yes,Moderate,None,Corticosteroids,No,1
105868,59,Female,Postmenopausal,Yes,Asian,Normal,Adequate,Insufficient,Active,No,None,None,Corticosteroids,Nc
```

Fig 2 - Dataset



```
Welcome    osteoporosis.csv ×    Prediction Osteoporosis Risk.ipynb    app.py
osteoporosis.csv
ıtake,Physical Activity,Smoking,Alcohol Consumption,Medical Conditions,Medications,Prior Fractures,Osteoporosis
thritis,Corticosteroids,Yes,1

:hyroidism,Corticosteroids,No,1
ıtoid Arthritis,Corticosteroids,No,1
id Arthritis,None,Yes,1
Corticosteroids,Yes,1
ɔne,Rheumatoid Arthritis,None,No,1
Arthritis,Corticosteroids,Yes,1
itis,Corticosteroids,No,1
steroids,Yes,1

matoid Arthritis,Corticosteroids,No,1
ɔsteroids,Yes,1
thyroidism,Corticosteroids,No,1
oidism,None,No,1
matoid Arthritis,None,No,1
rthyroidism,None,No,1
sm,None,No,1
None,None,No,1
Yes,1
ɔids,No,1
steroids,No,1
```

Fig 3 - Dataset

3. Implementation Details of Project

https://storage.me-qr.com/pdf/08ae2f46-e632-4315-90d4-f8c6a1c6d2d6.pdf

Fig 4 – Implementation Code PDF & Link

4. Streamlit App

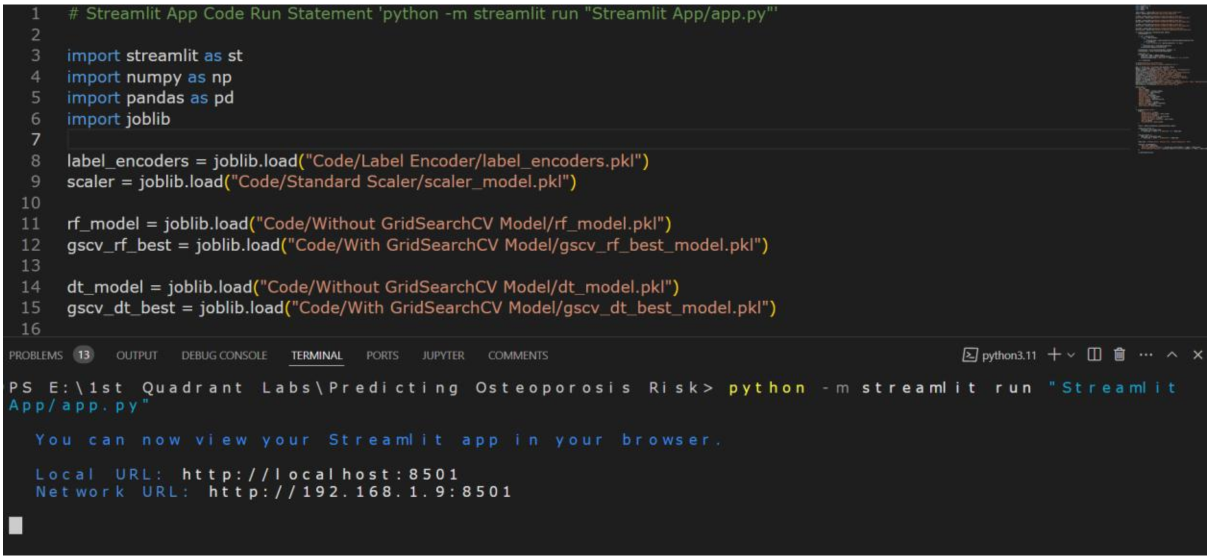
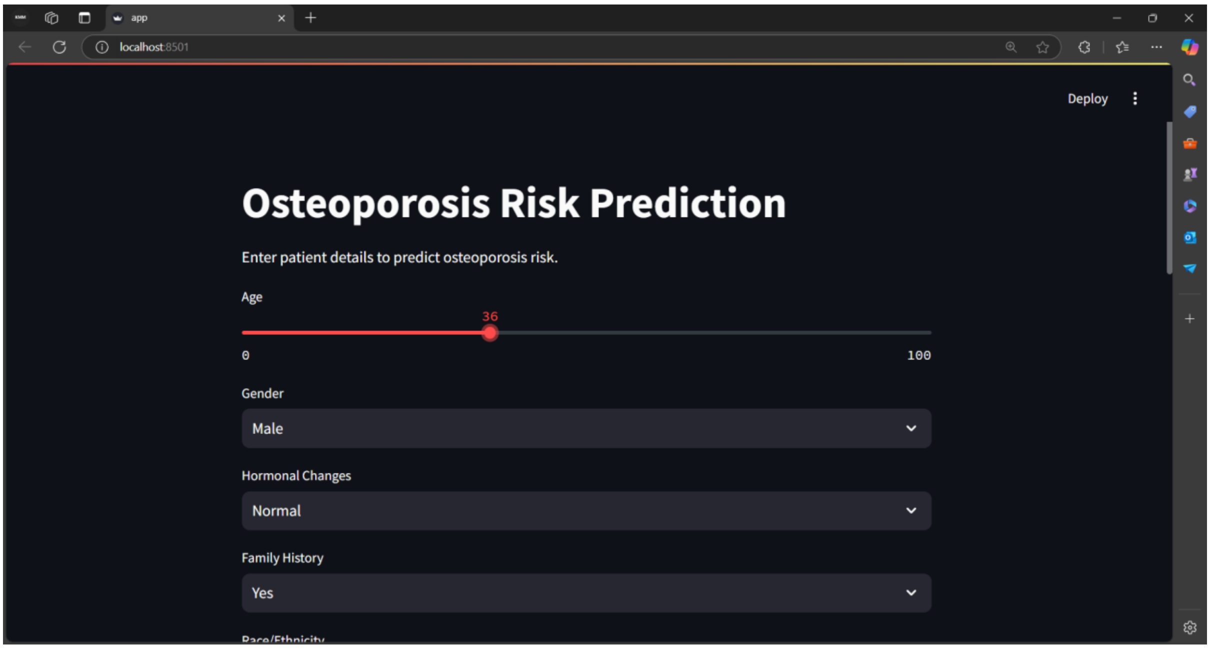


Fig 5 - Streamlit App Run



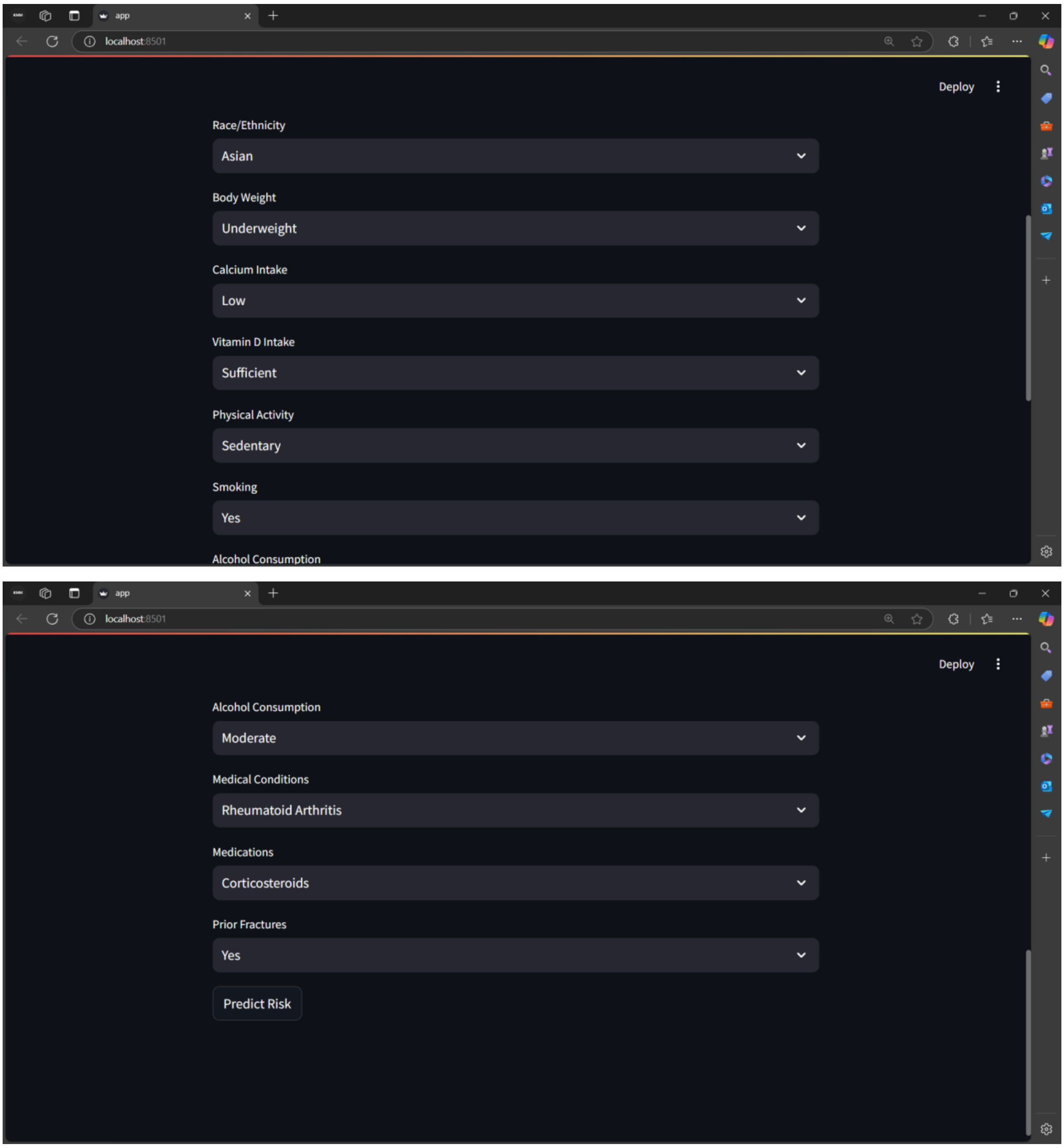


Fig 8 - Streamlit App Run in Data Input

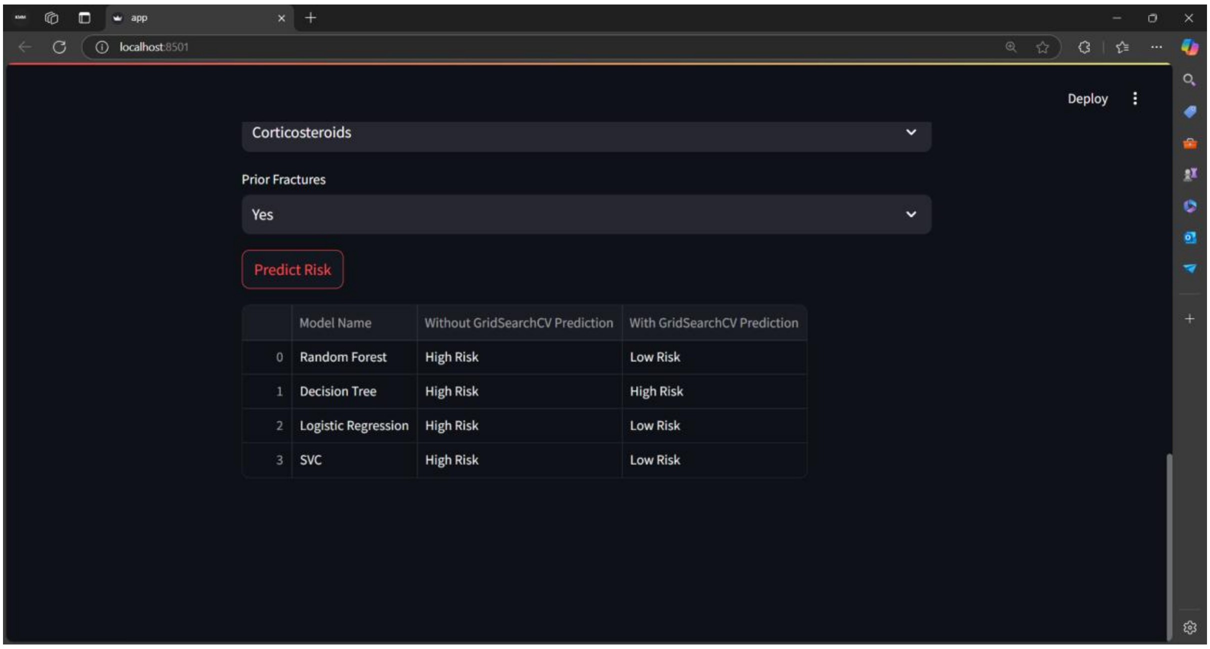


Fig 9 - Streamlit App Run in Data Input & Predict Osteoporosis Risk