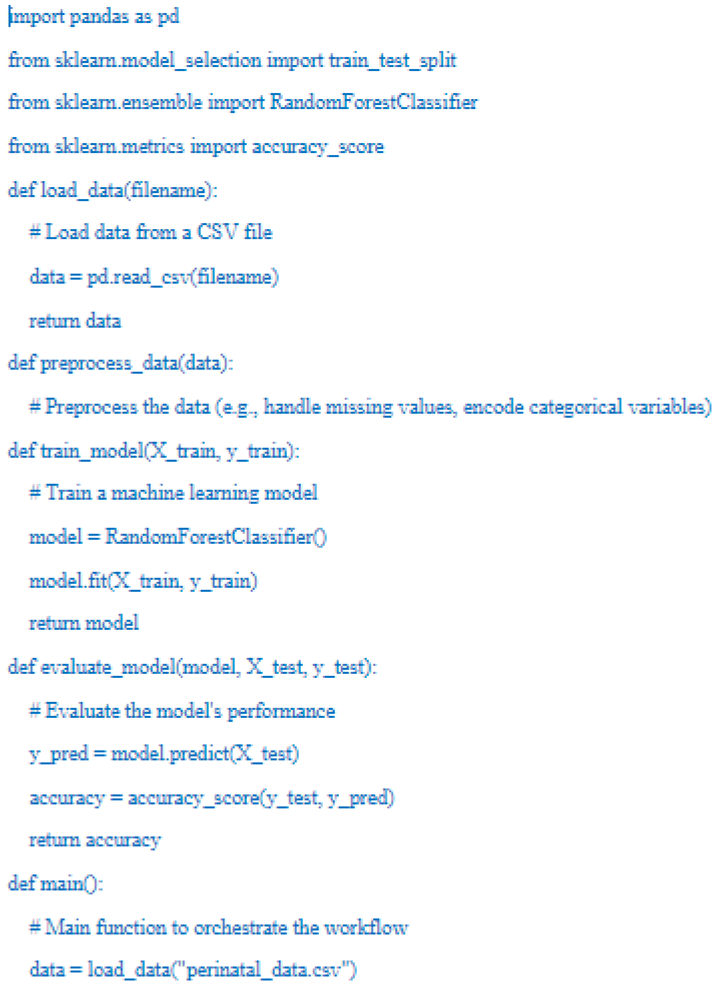
Project Development Phase

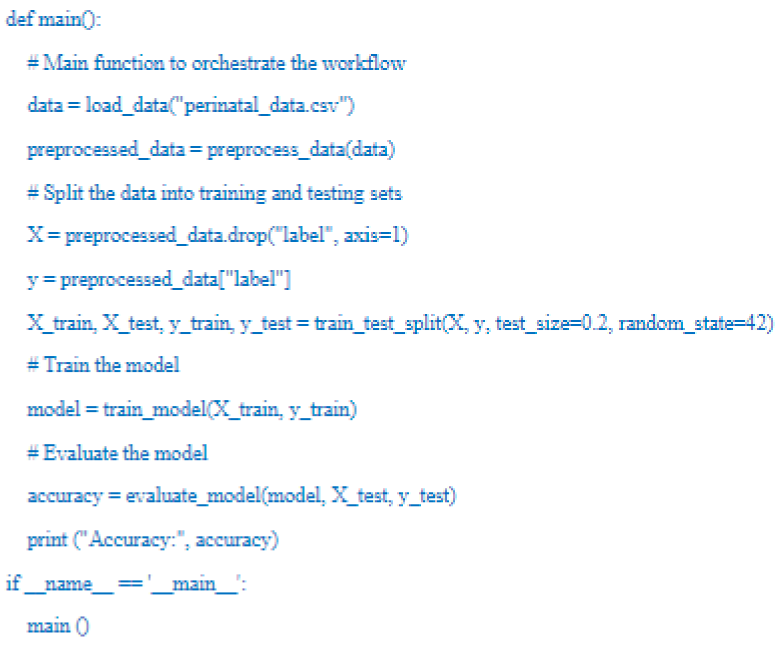
Date	18 May2023
Team ID	NM2023TMID00712
Project Name	Project - identifying perinatal health risks using machine learning techniques.

2.Code-Layout, Readability And Reusability

Code Layout, Readability, and Reusability are essential aspects to consider when developing a perinatal health risks identification system using machine learning (ML).



```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
def load_data(filename):
  # Load data from a CSV file
  data = pd.read_csv(filename)
  return data
def preprocess_data(data):
  # Preprocess the data (e.g., handle missing values, encode categorical variables)
def train_model(X_train, y_train):
  # Train a machine learning model
  model = RandomForestClassifier()
  model.fit(X_train, y_train)
  return model
def evaluate_model(model, X_test, y_test):
  # Evaluate the model's performance
  y_pred = model.predict(X_test)
  accuracy = accuracy_score(y_test, y_pred)
  return accuracy
def main():
  # Main function to orchestrate the workflow
  data = load_data("perinatal_data.csv")
```



```
def main():
    # Main function to orchestrate the workflow
    data = load_data("perinatal_data.csv")
    preprocessed_data = preprocess_data(data)
    # Split the data into training and testing sets
    X = preprocessed_data.drop("label", axis=1)
    y = preprocessed_data["label"]
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
    # Train the model
    model = train_model(X_train, y_train)
    # Evaluate the model
    accuracy = evaluate_model(model, X_test, y_test)
    print ("Accuracy:", accuracy)
if __name__ == '__main__':
    main ()
```

1. Code Layout:

- Use consistent indentation and spacing to improve code readability.

- Organize the code into logical sections or modules, separating different functionalities.

- Follow naming conventions for variables, functions, and classes to make the code more understandable.

- Include comments to explain the purpose of the code, important algorithms, or complex logic.

2. Readability:

- Write clear and concise code that is easy to understand and maintain.

- Use meaningful variable and function names that describe their purpose.

- Break down complex logic into smaller, well-documented steps.

- Avoid long and convoluted code blocks or nested structures.

- Follow best practices and coding standards of the programming language being used.

3. Reusability:

- Modularize the code into reusable functions or classes.

- Encapsulate ML models, preprocessing steps, and other components into separate modules.

- Design the code to be easily configurable and customizable for different use cases.

- Abstract common functionalities into utility functions or libraries that can be reused across multiple projects.

- Document the usage and inputs/outputs of reusable components to facilitate their reuse in other projects.