Ex no:7
14-04-2024

M. Rohith
3122 21 5001 085

Assignment-7

Machine Learning Lab

Diabetes Prediction

Program code:

```
# -*- coding: utf-8 -*-
"""PredictingDiabetes.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/17DcOzCHBVt7uw5_YeIEf0XRJL4FJXzyo?usp=sharing

# Diabetes Prediction using Decision Tree

## 1. Loading the dataset.
https://www.kaggle.com/datasets/iammustafatz/diabetes-prediction-dataset
"""

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("diabetes_prediction_dataset.csv")
df.head()

df.shape

"""## 2&3 Data Pre-processing and Exploratory Data Analysis

### Handling Missing Values
"""

df.isna().sum()
```

M. Rohith
3122 21 5001 085

```
"""### Handling Outliers"""

print(df.columns)
for i in df.columns:
    plt.figure(figsize=(5,4))
    plt.hist(df[i])
    plt.xlabel(i)
    plt.ylabel("Frequency")
    plt.show()

"""### Encoding"""

unique_groups = df['smoking_history'].unique()
print(unique_groups)

from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
df['smoking_history'] = le.fit_transform(df['smoking_history'])
df['gender'] = le.fit_transform(df['gender'])

unique_groups = df['smoking_history'].unique()
print(unique_groups)

"""### Normalization and Standardization"""

# from sklearn.preprocessing import MinMaxScaler, StandardScaler

# min_max_scaler = MinMaxScaler()

# X_normalized = min_max_scaler.fit_transform(df['HbA1c_level'])

# standard_scaler = StandardScaler()

# X_standardized =
standard_scaler.fit_transform(df['blood_glucose_level'])

# df['HbA1c_level'] = X_normalized[:, 0]

# df['blood_glucose_level'] = X_standardized[:, 1]

# plt.hist(df['HbA1c_level'])
# plt.show()
```

M. Rohith
3122 21 5001 085

```
# plt.hist(df['blood_glucose_level'])
# plt.show()

"""## 4 Feature Engineering"""

feature = df[list(df.columns[:-1])]
feature.head()

target = df[df.columns[-1]]
target.head()

from sklearn.feature_selection import SelectKBest, f_classif

k_best_selector = SelectKBest(score_func=f_classif, k=4)
k_best_selector

k_best_selector.fit(feature, target)

X_selected = k_best_selector.transform(feature)

print("Selected Features:", X_selected.shape[1])

selected_indices = k_best_selector.get_support(indices=True)
print("Indices of Selected Features:", selected_indices)
for i in selected_indices:
    print(df.columns[i])

"""## 5. Split the data into training, testing and validation
sets."""

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(feature, target,
test_size=0.33, random_state=42)

"""## 6. Model Selection."""

from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier()
rfc
```

M. Rohith
3122 21 5001 085

```
"""## 7. Train the Model"""

rf_classifier = RandomForestClassifier(n_estimators=100,
random_state=42)
rf_classifier.fit(X_train, y_train)

"""## 8. Measure the performance of the trained model."""

from sklearn.metrics import accuracy_score

y_pred = rf_classifier.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)

print(f"Accuracy: {accuracy*100:.2f}%")
```

Github Link:

https://colab.research.google.com/drive/17DcOzCHBVt7uw5_YeIEf0XRJL4FJXzyo?usp=sharing