IBM-NAAN MUDHALVAN AI BASED DIABETES PREDICTION SYSTEM

NAME : Dhesiyamani.K

REGISTER NO :620821243029

BRANCH :B.TECH.AI&DS

YEAR :3rd YEAR

TOPIC : ARTIFICIAL INTELLIGENCE

TITLE :AI BASED DIABETES PREDICTION SYSTEM

COLLEGE : GNANAMANI COLLEGE OF TECHNOLOGY

AI BASED DIABETES PREDICTION SYSTEM :

CODING:

cells": [

{

"cell\_type": "markdown",

"id": "1bae3b83",

"metadata": {

"papermill": {

"duration": 0.059794,

"end\_time": "2021-11-30T05:15:11.380525",

"exception": false,

"start\_time": "2021-11-30T05:15:11.320731",

"status": "completed"

},

"tags": []

},

"source": [

"# Diabetes Prediction using Python + Pandas\n",

"Diabetes is a chronic (long-lasting) health condition that affects how your body turns food into energy. Most of the food you eat is broken down into sugar (also called glucose) and released into your bloodstream. When your blood sugar goes up, it signals your pancreas to release insulin.\n"

]

},

{

"cell\_type": "markdown",

"id": "d78d2f2a",

"metadata": {

"papermill": {

"duration": 0.039393,

"end\_time": "2021-11-30T05:15:11.463399",

"exception": false,

"start\_time": "2021-11-30T05:15:11.424006",

"status": "completed"

},

"tags": []

},

"source": [

"# Table Content\n",

"\n",

"1. Introduction\n",

"2. Installing Libraries\n",

"3. Importing Data\n",

"4. Missing Value Analysis\n",

"5. Exploratory Data Analysis\n",

"6. Feature Engineering\n",

"7. Modeling\n",

"8. Hyperparameter Tuning\n",

"9. Prediction"

]

},

{

"cell\_type": "markdown",

"id": "d96aaf80",

"metadata": {

"papermill": {

"duration": 0.041399,

"end\_time": "2021-11-30T05:15:11.550742",

"exception": false,

"start\_time": "2021-11-30T05:15:11.509343",

"status": "completed"

},

"tags": []

},

"source": [

"# Introduction\n",

"According to WHO, Diabetes is a chronic disease that occurs either when the pancreas does not produce enough insulin or when the body cannot effectively use the insulin it produces. Insulin is a hormone that regulates blood sugar. Hyperglycaemia, or raised blood sugar, is a common effect of uncontrolled diabetes and over time leads to serious damage to many of the body's systems, especially the nerves and blood vessels.\n",

"\n",

"Between 2000 and 2016, there was a 5% increase in premature mortality rates (i.e. before the age of 70) from diabetes. In high-income countries the premature mortality rate due to diabetes decreased from 2000 to 2010 but then increased in 2010-2016. In lower-middle-income countries, the premature mortality rate due to diabetes increased across both periods.\n",

"\n",

"In this notebook, i will do some feature analysis and try to find out the rootcauses\n",

"\n",

"# <font size=\"5\">Objectives</font>\n",

"1. To experiment with different classification methods to see which yields the highest accuracy\n",

"2. Classify whether someone has diabetes or not from given features\n",

"3. To determine which features are the most indicative of diabetes\n",

"\n",

"# <font size=\"5\">Dataset</font>\n",

"I have used [Pima Indians Diabetes Database](https://www.kaggle.com/uciml/pima-indians-diabetes-database) Kaggle Dataset\n",

"\n",

"The dataset contains below features and labels:\n",

"1. Pregnancies\n",

"2. Glucose\n",

"3. BloodPressure\n",

"4. SkinThickness\n",

"5. Insulin\n",

"6. BMI\n",

"7. DiabetesPedigreeFunction\n",

"8. Age\n",

"9. Outcome\n"

]

},

{

"cell\_type": "markdown",

"id": "8963d260",

"metadata": {

"papermill": {

"duration": 0.038727,

"end\_time": "2021-11-30T05:15:11.645515",

"exception": false,

"start\_time": "2021-11-30T05:15:11.606788",

"status": "completed"

},

"tags": []

},

"source": [

"# Installing Libraries"

]

},

{

"cell\_type": "code",

"execution\_count": 1,

"id": "d2175b7d",

"metadata": {

"\_cell\_guid": "b1076dfc-b9ad-4769-8c92-a6c4dae69d19",

"\_uuid": "8f2839f25d086af736a60e9eeb907d3b93b6e0e5",

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:11.738103Z",

"iopub.status.busy": "2021-11-30T05:15:11.737244Z",

"iopub.status.idle": "2021-11-30T05:15:13.074037Z",

"shell.execute\_reply": "2021-11-30T05:15:13.073404Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.123506Z"

},

"papermill": {

"duration": 1.388994,

"end\_time": "2021-11-30T05:15:13.074233",

"exception": false,

"start\_time": "2021-11-30T05:15:11.685239",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"/kaggle/input/pima-indians-diabetes-database/diabetes.csv\n"

]

}

],

"source": [

"import numpy as np # linear algebra\n",

"import pandas as pd # data processing, CSV file I/O (e.g. pd.read\_csv)\n",

"import seaborn as sns # for data visualization\n",

"import matplotlib.pyplot as plt # to plot charts\n",

"from collections import Counter\n",

"import os\n",

"\n",

"# Modeling\n",

"from sklearn.preprocessing import QuantileTransformer\n",

"from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score\n",

"from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier, GradientBoostingClassifier, VotingClassifier\n",

"from sklearn.linear\_model import LogisticRegression\n",

"from sklearn.neighbors import KNeighborsClassifier\n",

"from sklearn.tree import DecisionTreeClassifier\n",

"from sklearn.svm import SVC\n",

"from sklearn.model\_selection import GridSearchCV, cross\_val\_score, StratifiedKFold, learning\_curve, train\_test\_split\n",

"\n",

"\n",

"# Directory Structure\n",

"for dirname, \_, filenames in os.walk('/kaggle/input'):\n",

" for filename in filenames:\n",

" print(os.path.join(dirname, filename))"

]

},

{

"cell\_type": "markdown",

"id": "9354bc5e",

"metadata": {

"papermill": {

"duration": 0.114546,

"end\_time": "2021-11-30T05:15:13.229994",

"exception": false,

"start\_time": "2021-11-30T05:15:13.115448",

"status": "completed"

},

"tags": []

},

"source": [

"I have imported most common libraries used in python for machine learning such as Pandas, Seaborn, Matplitlib etc"

]

},

{

"cell\_type": "markdown",

"id": "435a0949",

"metadata": {

"papermill": {

"duration": 0.046981,

"end\_time": "2021-11-30T05:15:13.336076",

"exception": false,

"start\_time": "2021-11-30T05:15:13.289095",

"status": "completed"

},

"tags": []

},

"source": [

"# Importing Data"

]

},

{

"cell\_type": "code",

"execution\_count": 2,

"id": "3ddeabf8",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:13.425011Z",

"iopub.status.busy": "2021-11-30T05:15:13.424029Z",

"iopub.status.idle": "2021-11-30T05:15:13.440105Z",

"shell.execute\_reply": "2021-11-30T05:15:13.440699Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.147079Z"

},

"papermill": {

"duration": 0.063263,

"end\_time": "2021-11-30T05:15:13.440865",

"exception": false,

"start\_time": "2021-11-30T05:15:13.377602",

"status": "completed"

},

"tags": []

},

"outputs": [],

"source": [

"df = pd.read\_csv(\"../input/pima-indians-diabetes-database/diabetes.csv\")"

]

},

{

"cell\_type": "code",

"execution\_count": 3,

"id": "e8cb5c7b",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:13.525918Z",

"iopub.status.busy": "2021-11-30T05:15:13.524958Z",

"iopub.status.idle": "2021-11-30T05:15:13.549461Z",

"shell.execute\_reply": "2021-11-30T05:15:13.548890Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.166197Z"

},

"papermill": {

"duration": 0.068205,

"end\_time": "2021-11-30T05:15:13.549614",

"exception": false,

"start\_time": "2021-11-30T05:15:13.481409",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"name": "stdout",

"output\_type": "stream",

"text": [

"<class 'pandas.core.frame.DataFrame'>\n",

"RangeIndex: 768 entries, 0 to 767\n",

"Data columns (total 9 columns):\n",

" # Column Non-Null Count Dtype \n",

"--- ------ -------------- ----- \n",

" 0 Pregnancies 768 non-null int64 \n",

" 1 Glucose 768 non-null int64 \n",

" 2 BloodPressure 768 non-null int64 \n",

" 3 SkinThickness 768 non-null int64 \n",

" 4 Insulin 768 non-null int64 \n",

" 5 BMI 768 non-null float64\n",

" 6 DiabetesPedigreeFunction 768 non-null float64\n",

" 7 Age 768 non-null int64 \n",

" 8 Outcome 768 non-null int64 \n",

"dtypes: float64(2), int64(7)\n",

"memory usage: 54.1 KB\n"

]

}

],

"source": [

"# Get familier with dataset structure\n",

"\n",

"df.info()"

]

},

{

"cell\_type": "markdown",

"id": "d412a00f",

"metadata": {

"papermill": {

"duration": 0.038842,

"end\_time": "2021-11-30T05:15:13.628132",

"exception": false,

"start\_time": "2021-11-30T05:15:13.589290",

"status": "completed"

},

"tags": []

},

"source": [

"Excepting BMI and DiabetesPedigreeFunction all the columns are integer. Outcome is the label containing 1 and 0 values. 1 means person has diabetes and 0 mean person is not diabetic"

]

},

{

"cell\_type": "code",

"execution\_count": 4,

"id": "731f9a17",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:13.716272Z",

"iopub.status.busy": "2021-11-30T05:15:13.715629Z",

"iopub.status.idle": "2021-11-30T05:15:13.750333Z",

"shell.execute\_reply": "2021-11-30T05:15:13.750782Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.184479Z"

},

"papermill": {

"duration": 0.082634,

"end\_time": "2021-11-30T05:15:13.750959",

"exception": false,

"start\_time": "2021-11-30T05:15:13.668325",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Pregnancies</th>\n",

" <th>Glucose</th>\n",

" <th>BloodPressure</th>\n",

" <th>SkinThickness</th>\n",

" <th>Insulin</th>\n",

" <th>BMI</th>\n",

" <th>DiabetesPedigreeFunction</th>\n",

" <th>Age</th>\n",

" <th>Outcome</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>3.845052</td>\n",

" <td>120.894531</td>\n",

" <td>69.105469</td>\n",

" <td>20.536458</td>\n",

" <td>79.799479</td>\n",

" <td>31.992578</td>\n",

" <td>0.471876</td>\n",

" <td>33.240885</td>\n",

" <td>0.348958</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>3.369578</td>\n",

" <td>31.972618</td>\n",

" <td>19.355807</td>\n",

" <td>15.952218</td>\n",

" <td>115.244002</td>\n",

" <td>7.884160</td>\n",

" <td>0.331329</td>\n",

" <td>11.760232</td>\n",

" <td>0.476951</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.078000</td>\n",

" <td>21.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>1.000000</td>\n",

" <td>99.000000</td>\n",

" <td>62.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>27.300000</td>\n",

" <td>0.243750</td>\n",

" <td>24.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>3.000000</td>\n",

" <td>117.000000</td>\n",

" <td>72.000000</td>\n",

" <td>23.000000</td>\n",

" <td>30.500000</td>\n",

" <td>32.000000</td>\n",

" <td>0.372500</td>\n",

" <td>29.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>6.000000</td>\n",

" <td>140.250000</td>\n",

" <td>80.000000</td>\n",

" <td>32.000000</td>\n",

" <td>127.250000</td>\n",

" <td>36.600000</td>\n",

" <td>0.626250</td>\n",

" <td>41.000000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>17.000000</td>\n",

" <td>199.000000</td>\n",

" <td>122.000000</td>\n",

" <td>99.000000</td>\n",

" <td>846.000000</td>\n",

" <td>67.100000</td>\n",

" <td>2.420000</td>\n",

" <td>81.000000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Pregnancies Glucose BloodPressure SkinThickness Insulin \\\n",

"count 768.000000 768.000000 768.000000 768.000000 768.000000 \n",

"mean 3.845052 120.894531 69.105469 20.536458 79.799479 \n",

"std 3.369578 31.972618 19.355807 15.952218 115.244002 \n",

"min 0.000000 0.000000 0.000000 0.000000 0.000000 \n",

"25% 1.000000 99.000000 62.000000 0.000000 0.000000 \n",

"50% 3.000000 117.000000 72.000000 23.000000 30.500000 \n",

"75% 6.000000 140.250000 80.000000 32.000000 127.250000 \n",

"max 17.000000 199.000000 122.000000 99.000000 846.000000 \n",

"\n",

" BMI DiabetesPedigreeFunction Age Outcome \n",

"count 768.000000 768.000000 768.000000 768.000000 \n",

"mean 31.992578 0.471876 33.240885 0.348958 \n",

"std 7.884160 0.331329 11.760232 0.476951 \n",

"min 0.000000 0.078000 21.000000 0.000000 \n",

"25% 27.300000 0.243750 24.000000 0.000000 \n",

"50% 32.000000 0.372500 29.000000 0.000000 \n",

"75% 36.600000 0.626250 41.000000 1.000000 \n",

"max 67.100000 2.420000 81.000000 1.000000 "

]

},

"execution\_count": 4,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df.describe()"

]

},

{

"cell\_type": "markdown",

"id": "8a8834ce",

"metadata": {

"papermill": {

"duration": 0.042081,

"end\_time": "2021-11-30T05:15:13.832527",

"exception": false,

"start\_time": "2021-11-30T05:15:13.790446",

"status": "completed"

},

"tags": []

},

"source": [

"There are 768 records in the dataset, in which mean age of people is 33"

]

},

{

"cell\_type": "markdown",

"id": "eb0fbb3c",

"metadata": {

"papermill": {

"duration": 0.039823,

"end\_time": "2021-11-30T05:15:13.912765",

"exception": false,

"start\_time": "2021-11-30T05:15:13.872942",

"status": "completed"

},

"tags": []

},

"source": [

"# Missing Value Analysis"

]

},

{

"cell\_type": "markdown",

"id": "b6c0e311",

"metadata": {

"papermill": {

"duration": 0.039307,

"end\_time": "2021-11-30T05:15:13.992391",

"exception": false,

"start\_time": "2021-11-30T05:15:13.953084",

"status": "completed"

},

"tags": []

},

"source": [

"Next, i will cleanup the dataset which is the important part of data science. Missing data can lead to wrong statistics during modeling and predictions."

]

},

{

"cell\_type": "code",

"execution\_count": 5,

"id": "620d511b",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:14.083930Z",

"iopub.status.busy": "2021-11-30T05:15:14.079840Z",

"iopub.status.idle": "2021-11-30T05:15:14.109071Z",

"shell.execute\_reply": "2021-11-30T05:15:14.108552Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.235530Z"

},

"papermill": {

"duration": 0.076312,

"end\_time": "2021-11-30T05:15:14.109246",

"exception": false,

"start\_time": "2021-11-30T05:15:14.032934",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",

" <tr style=\"text-align: right;\">\n",

" <th></th>\n",

" <th>Pregnancies</th>\n",

" <th>Glucose</th>\n",

" <th>BloodPressure</th>\n",

" <th>SkinThickness</th>\n",

" <th>Insulin</th>\n",

" <th>BMI</th>\n",

" <th>DiabetesPedigreeFunction</th>\n",

" <th>Age</th>\n",

" <th>Outcome</th>\n",

" </tr>\n",

" </thead>\n",

" <tbody>\n",

" <tr>\n",

" <th>count</th>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" <td>768.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>mean</th>\n",

" <td>3.845052</td>\n",

" <td>120.894531</td>\n",

" <td>69.105469</td>\n",

" <td>20.536458</td>\n",

" <td>79.799479</td>\n",

" <td>31.992578</td>\n",

" <td>0.471876</td>\n",

" <td>33.240885</td>\n",

" <td>0.348958</td>\n",

" </tr>\n",

" <tr>\n",

" <th>std</th>\n",

" <td>3.369578</td>\n",

" <td>31.972618</td>\n",

" <td>19.355807</td>\n",

" <td>15.952218</td>\n",

" <td>115.244002</td>\n",

" <td>7.884160</td>\n",

" <td>0.331329</td>\n",

" <td>11.760232</td>\n",

" <td>0.476951</td>\n",

" </tr>\n",

" <tr>\n",

" <th>min</th>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.078000</td>\n",

" <td>21.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>25%</th>\n",

" <td>1.000000</td>\n",

" <td>99.000000</td>\n",

" <td>62.000000</td>\n",

" <td>0.000000</td>\n",

" <td>0.000000</td>\n",

" <td>27.300000</td>\n",

" <td>0.243750</td>\n",

" <td>24.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>50%</th>\n",

" <td>3.000000</td>\n",

" <td>117.000000</td>\n",

" <td>72.000000</td>\n",

" <td>23.000000</td>\n",

" <td>30.500000</td>\n",

" <td>32.000000</td>\n",

" <td>0.372500</td>\n",

" <td>29.000000</td>\n",

" <td>0.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>75%</th>\n",

" <td>6.000000</td>\n",

" <td>140.250000</td>\n",

" <td>80.000000</td>\n",

" <td>32.000000</td>\n",

" <td>127.250000</td>\n",

" <td>36.600000</td>\n",

" <td>0.626250</td>\n",

" <td>41.000000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" <tr>\n",

" <th>max</th>\n",

" <td>17.000000</td>\n",

" <td>199.000000</td>\n",

" <td>122.000000</td>\n",

" <td>99.000000</td>\n",

" <td>846.000000</td>\n",

" <td>67.100000</td>\n",

" <td>2.420000</td>\n",

" <td>81.000000</td>\n",

" <td>1.000000</td>\n",

" </tr>\n",

" </tbody>\n",

"</table>\n",

"</div>"

],

"text/plain": [

" Pregnancies Glucose BloodPressure SkinThickness Insulin \\\n",

"count 768.000000 768.000000 768.000000 768.000000 768.000000 \n",

"mean 3.845052 120.894531 69.105469 20.536458 79.799479 \n",

"std 3.369578 31.972618 19.355807 15.952218 115.244002 \n",

"min 0.000000 0.000000 0.000000 0.000000 0.000000 \n",

"25% 1.000000 99.000000 62.000000 0.000000 0.000000 \n",

"50% 3.000000 117.000000 72.000000 23.000000 30.500000 \n",

"75% 6.000000 140.250000 80.000000 32.000000 127.250000 \n",

"max 17.000000 199.000000 122.000000 99.000000 846.000000 \n",

"\n",

" BMI DiabetesPedigreeFunction Age Outcome \n",

"count 768.000000 768.000000 768.000000 768.000000 \n",

"mean 31.992578 0.471876 33.240885 0.348958 \n",

"std 7.884160 0.331329 11.760232 0.476951 \n",

"min 0.000000 0.078000 21.000000 0.000000 \n",

"25% 27.300000 0.243750 24.000000 0.000000 \n",

"50% 32.000000 0.372500 29.000000 0.000000 \n",

"75% 36.600000 0.626250 41.000000 1.000000 \n",

"max 67.100000 2.420000 81.000000 1.000000 "

]

},

"execution\_count": 5,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"\n",

"df.describe()"

]

},

{

"cell\_type": "code",

"execution\_count": 6,

"id": "208d3d58",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:14.198805Z",

"iopub.status.busy": "2021-11-30T05:15:14.197859Z",

"iopub.status.idle": "2021-11-30T05:15:14.202101Z",

"shell.execute\_reply": "2021-11-30T05:15:14.201636Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.279629Z"

},

"papermill": {

"duration": 0.051573,

"end\_time": "2021-11-30T05:15:14.202261",

"exception": false,

"start\_time": "2021-11-30T05:15:14.150688",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"data": {

"text/plain": [

"Pregnancies 0\n",

"Glucose 0\n",

"BloodPressure 0\n",

"SkinThickness 0\n",

"Insulin 0\n",

"BMI 0\n",

"DiabetesPedigreeFunction 0\n",

"Age 0\n",

"Outcome 0\n",

"dtype: int64"

]

},

"execution\_count": 6,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"# Explore missing values\n",

"\n",

"df.isnull().sum()"

]

},

{

"cell\_type": "markdown",

"id": "4e8a94a5",

"metadata": {

"papermill": {

"duration": 0.040776,

"end\_time": "2021-11-30T05:15:14.284473",

"exception": false,

"start\_time": "2021-11-30T05:15:14.243697",

"status": "completed"

},

"tags": []

},

"source": [

"I observed that there is no missing values in dataset however the features like Glucose, BloodPressure, Insulin, SkinThickness has 0 values which is not possible. We have to replace 0 values with either mean or median values of specific column"

]

},

{

"cell\_type": "code",

"execution\_count": 7,

"id": "c07e7755",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:14.379626Z",

"iopub.status.busy": "2021-11-30T05:15:14.378791Z",

"iopub.status.idle": "2021-11-30T05:15:14.382475Z",

"shell.execute\_reply": "2021-11-30T05:15:14.381902Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.288646Z"

},

"papermill": {

"duration": 0.056765,

"end\_time": "2021-11-30T05:15:14.382643",

"exception": false,

"start\_time": "2021-11-30T05:15:14.325878",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"data": {

"text/plain": [

"99.0 17\n",

"100.0 17\n",

"111.0 14\n",

"129.0 14\n",

"125.0 14\n",

" ..\n",

"191.0 1\n",

"177.0 1\n",

"44.0 1\n",

"62.0 1\n",

"190.0 1\n",

"Name: Glucose, Length: 136, dtype: int64"

]

},

"execution\_count": 7,

"metadata": {},

"output\_type": "execute\_result"

}

],

"source": [

"df['Glucose'] = df['Glucose'].replace(0,df['Glucose'].mean())\n",

"df.Glucose.value\_counts()"

]

},

{

"cell\_type": "code",

"execution\_count": 8,

"id": "34560425",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:14.473157Z",

"iopub.status.busy": "2021-11-30T05:15:14.470796Z",

"iopub.status.idle": "2021-11-30T05:15:14.476704Z",

"shell.execute\_reply": "2021-11-30T05:15:14.476027Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.305659Z"

},

"papermill": {

"duration": 0.05235,

"end\_time": "2021-11-30T05:15:14.476846",

"exception": false,

"start\_time": "2021-11-30T05:15:14.424496",

"status": "completed"

},

"tags": []

},

"outputs": [],

"source": [

"# Correcting missing values in blood pressure\n",

"\n",

"df[df['BloodPressure'] == 0]['BloodPressure'].value\_counts()\n",

"df['BloodPressure'] = df['BloodPressure'].replace(0,df['BloodPressure'].mean())"

]

},

{

"cell\_type": "markdown",

"id": "a1e4470c",

"metadata": {

"papermill": {

"duration": 0.041225,

"end\_time": "2021-11-30T05:15:14.558833",

"exception": false,

"start\_time": "2021-11-30T05:15:14.517608",

"status": "completed"

},

"tags": []

},

"source": [

"There are 35 records with 0 BloodPressure in dataset"

]

},

{

"cell\_type": "code",

"execution\_count": 9,

"id": "66003a92",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:14.649006Z",

"iopub.status.busy": "2021-11-30T05:15:14.647988Z",

"iopub.status.idle": "2021-11-30T05:15:14.650837Z",

"shell.execute\_reply": "2021-11-30T05:15:14.650309Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.315761Z"

},

"papermill": {

"duration": 0.051419,

"end\_time": "2021-11-30T05:15:14.650984",

"exception": false,

"start\_time": "2021-11-30T05:15:14.599565",

"status": "completed"

},

"tags": []

},

"outputs": [],

"source": [

"# Correcting missing values in BMI\n",

"\n",

"df[df['BMI'] == 0]['BMI'].value\_counts()\n",

"df['BMI'] = df['BMI'].replace(0, df['BMI'].median())"

]

},

{

"cell\_type": "code",

"execution\_count": 10,

"id": "f9594160",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:14.737773Z",

"iopub.status.busy": "2021-11-30T05:15:14.736822Z",

"iopub.status.idle": "2021-11-30T05:15:14.742769Z",

"shell.execute\_reply": "2021-11-30T05:15:14.743251Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.326010Z"

},

"papermill": {

"duration": 0.050558,

"end\_time": "2021-11-30T05:15:14.743420",

"exception": false,

"start\_time": "2021-11-30T05:15:14.692862",

"status": "completed"

},

"tags": []

},

"outputs": [],

"source": [

"# Correct missing values in Insulin and SkinThickness\n",

"\n",

"df['SkinThickness'] = df['SkinThickness'].replace(0, df['SkinThickness'].median())\n",

"df['Insulin'] = df['Insulin'].replace(0, df['Insulin'].median())"

]

},

{

"cell\_type": "code",

"execution\_count": 11,

"id": "310ebfc8",

"metadata": {

"execution": {

"iopub.execute\_input": "2021-11-30T05:15:14.828895Z",

"iopub.status.busy": "2021-11-30T05:15:14.827939Z",

"iopub.status.idle": "2021-11-30T05:15:14.860293Z",

"shell.execute\_reply": "2021-11-30T05:15:14.860732Z",

"shell.execute\_reply.started": "2021-11-30T05:02:07.341235Z"

},

"papermill": {

"duration": 0.076804,

"end\_time": "2021-11-30T05:15:14.860906",

"exception": false,

"start\_time": "2021-11-30T05:15:14.784102",

"status": "completed"

},

"tags": []

},

"outputs": [

{

"data": {

"text/html": [

"<div>\n",

"<style scoped>\n",

" .dataframe tbody tr th:only-of-type {\n",

" vertical-align: middle;\n",

" }\n",

"\n",

" .dataframe tbody tr th {\n",

" vertical-align: top;\n",

" }\n",

"\n",

" .dataframe thead th {\n",

" text-align: right;\n",

" }\n",

"</style>\n",

"<table border=\"1\" class=\"dataframe\">\n",

" <thead>\n",