

Handling The Missing Values

Team ID	PNT2022TMID17480
Project Title	Early Detection Of Chronic Kidney Disease Using Machine Learning

```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "execution_count": 11,
      "metadata": {
```

```
"id": "DE0P0ozyiEyl"
},
"outputs": [],
"source": [
    "import pandas as pd\n",
    "import numpy as np\n",
    "import matplotlib.pyplot as plt\n",
    "import sklearn as sk\n",
    "import seaborn as sns\n",
    "from sklearn.preprocessing import LabelEncoder\n",
    "from sklearn.model_selection import train_test_split\n",
    "from sklearn.preprocessing import StandardScaler\n",
    "from sklearn.linear_model import LogisticRegression\n",
    "from sklearn.neighbors import KNeighborsClassifier\n",
    "from sklearn.tree import DecisionTreeClassifier"
]
},
{
    "cell_type": "code",
    "source": [
        "df=pd.read_csv('chronickidneydisease.csv')\n",
        "df"
    ],
    "metadata": {
        "colab": {
```

```

"base_uri": "https://localhost:8080/",
"height": 468
},
"id": "baMtgb1EiPLs",
"outputId": "6f6af90b-a462-4898-e89a-b61a62d9c0e3"
},
"execution_count": 12,
"outputs": [
{
  "output_type": "execute_result",
  "data": {
    "text/plain": [
      "   id  age  bp   sg  al  su   rbc    pc    pcc  \\n",
      "0    0 48.0 80.0 1.020 1.0 0.0   NaN   normal notpresent  \\n",
      "1    1  7.0 50.0 1.020 4.0 0.0   NaN   normal notpresent  \\n",
      "2    2 62.0 80.0 1.010 2.0 3.0  normal   normal notpresent  \\n",
      "3    3 48.0 70.0 1.005 4.0 0.0  normal abnormal   present  \\n",
      "4    4 51.0 80.0 1.010 2.0 0.0  normal   normal notpresent  \\n",
      "..  ...  ...  ...  ...  ...  ...  ...    ...    ...  \\n",
      "395 395 55.0 80.0 1.020 0.0 0.0  normal   normal notpresent  \\n",
      "396 396 42.0 70.0 1.025 0.0 0.0  normal   normal notpresent  \\n",
      "397 397 12.0 80.0 1.020 0.0 0.0  normal   normal notpresent  \\n",
      "398 398 17.0 60.0 1.025 0.0 0.0  normal   normal notpresent  \\n",
      "399 399 58.0 80.0 1.025 0.0 0.0  normal   normal notpresent  \\n",
      "\\n",

```

```

"      ba ... pcv  wc  rc htn  dm cad appet  pe ane \\n",
"0  notpresent ... 44 7800 5.2 yes yes  no good  no  no  \n",
"1  notpresent ... 38 6000 NaN  no  no  no good  no  no  \n",
"2  notpresent ... 31 7500 NaN  no yes  no poor  no yes \n",
"3  notpresent ... 32 6700 3.9 yes  no  no poor  yes yes \n",
"4  notpresent ... 35 7300 4.6  no  no  no good  no  no  \n",
"..      ... .. \n",
"395 notpresent ... 47 6700 4.9  no  no  no good  no  no  \n",
"396 notpresent ... 54 7800 6.2  no  no  no good  no  no  \n",
"397 notpresent ... 49 6600 5.4  no  no  no good  no  no  \n",
"398 notpresent ... 51 7200 5.9  no  no  no good  no  no  \n",
"399 notpresent ... 53 6800 6.1  no  no  no good  no  no  \n",
"\n",
"  classification \n",
"0      ckd \n",
"1      ckd \n",
"2      ckd \n",
"3      ckd \n",
"4      ckd \n",
"..      ... \n",
"395      notckd \n",
"396      notckd \n",
"397      notckd \n",
"398      notckd \n",
"399      notckd \n",

```

```
"\n",
"[400 rows x 26 columns]"
],
"text/html": [
  "\n",
  " <div id=\"df-1d93ffec-5973-41e1-a880-8612aaf0d727\">\n",
  "   <div class=\"colab-df-container\">\n",
  "     <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>id</th>\n",
"    <th>age</th>\n",
"    <th>bp</th>\n",
"    <th>sg</th>\n",
"    <th>al</th>\n",
"    <th>su</th>\n",
"    <th>rbc</th>\n",
"    <th>pc</th>\n",
"    <th>pcc</th>\n",
"    <th>ba</th>\n",
"    <th>...</th>\n",
"    <th>pcv</th>\n",
"    <th>wc</th>\n",
"    <th>rc</th>\n",
"    <th>htn</th>\n",
"    <th>dm</th>\n",
"    <th>cad</th>\n",
"    <th>appet</th>\n",
"    <th>pe</th>\n",
"    <th>ane</th>\n",
"    <th>classification</th>\n",
"  </tr>\n",
" </thead>\n",
" <tbody>\n",
"  <tr>\n",
```

" <th>0</th>\n",
" <td>0</td>\n",
" <td>48.0</td>\n",
" <td>80.0</td>\n",
" <td>1.020</td>\n",
" <td>1.0</td>\n",
" <td>0.0</td>\n",
" <td>NaN</td>\n",
" <td>normal</td>\n",
" <td>notpresent</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>44</td>\n",
" <td>7800</td>\n",
" <td>5.2</td>\n",
" <td>yes</td>\n",
" <td>yes</td>\n",
" <td>no</td>\n",
" <td>good</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>ckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>1</th>\n",

" <td>1</td>\n",
" <td>7.0</td>\n",
" <td>50.0</td>\n",
" <td>1.020</td>\n",
" <td>4.0</td>\n",
" <td>0.0</td>\n",
" <td>NaN</td>\n",
" <td>normal</td>\n",
" <td>notpresent</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>38</td>\n",
" <td>6000</td>\n",
" <td>NaN</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>good</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>ckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>2</th>\n",
" <td>2</td>\n",

" <td>62.0</td>\n",
" <td>80.0</td>\n",
" <td>1.010</td>\n",
" <td>2.0</td>\n",
" <td>3.0</td>\n",
" <td>normal</td>\n",
" <td>normal</td>\n",
" <td>notpresent</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>31</td>\n",
" <td>7500</td>\n",
" <td>NaN</td>\n",
" <td>no</td>\n",
" <td>yes</td>\n",
" <td>no</td>\n",
" <td>poor</td>\n",
" <td>no</td>\n",
" <td>yes</td>\n",
" <td>ckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>3</th>\n",
" <td>3</td>\n",
" <td>48.0</td>\n",

" <td>70.0</td>\n",
" <td>1.005</td>\n",
" <td>4.0</td>\n",
" <td>0.0</td>\n",
" <td>normal</td>\n",
" <td>abnormal</td>\n",
" <td>present</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>32</td>\n",
" <td>6700</td>\n",
" <td>3.9</td>\n",
" <td>yes</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>poor</td>\n",
" <td>yes</td>\n",
" <td>yes</td>\n",
" <td>ckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>4</th>\n",
" <td>4</td>\n",
" <td>51.0</td>\n",
" <td>80.0</td>\n",

" <td>1.010</td>\n",
" <td>2.0</td>\n",
" <td>0.0</td>\n",
" <td>normal</td>\n",
" <td>normal</td>\n",
" <td>notpresent</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>35</td>\n",
" <td>7300</td>\n",
" <td>4.6</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>good</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>ckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>...</th>\n",
" <td>...</td>\n",
" <td>...</td>\n",
" <td>...</td>\n",
" <td>...</td>\n",

[illegible]

" <td>0.0</td>\n",
" <td>normal</td>\n",
" <td>normal</td>\n",
" <td>notpresent</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>47</td>\n",
" <td>6700</td>\n",
" <td>4.9</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>good</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>notckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>396</th>\n",
" <td>396</td>\n",
" <td>42.0</td>\n",
" <td>70.0</td>\n",
" <td>1.025</td>\n",
" <td>0.0</td>\n",
" <td>0.0</td>\n",

" <td>normal</td>\n",
" <td>normal</td>\n",
" <td>notpresent</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>54</td>\n",
" <td>7800</td>\n",
" <td>6.2</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>good</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>notckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>397</th>\n",
" <td>397</td>\n",
" <td>12.0</td>\n",
" <td>80.0</td>\n",
" <td>1.020</td>\n",
" <td>0.0</td>\n",
" <td>0.0</td>\n",
" <td>normal</td>\n",

" <td>normal</td>\n",
" <td>notpresent</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>49</td>\n",
" <td>6600</td>\n",
" <td>5.4</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>good</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>notckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>398</th>\n",
" <td>398</td>\n",
" <td>17.0</td>\n",
" <td>60.0</td>\n",
" <td>1.025</td>\n",
" <td>0.0</td>\n",
" <td>0.0</td>\n",
" <td>normal</td>\n",
" <td>normal</td>\n",

" <td>notpresent</td>\n",
" <td>notpresent</td>\n",
" <td>...</td>\n",
" <td>51</td>\n",
" <td>7200</td>\n",
" <td>5.9</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>good</td>\n",
" <td>no</td>\n",
" <td>no</td>\n",
" <td>notckd</td>\n",
" </tr>\n",
" <tr>\n",
" <th>399</th>\n",
" <td>399</td>\n",
" <td>58.0</td>\n",
" <td>80.0</td>\n",
" <td>1.025</td>\n",
" <td>0.0</td>\n",
" <td>0.0</td>\n",
" <td>normal</td>\n",
" <td>normal</td>\n",
" <td>notpresent</td>\n",


```

"    <td>notpresent</td>\n",
"    <td>...</td>\n",
"    <td>53</td>\n",
"    <td>6800</td>\n",
"    <td>6.1</td>\n",
"    <td>no</td>\n",
"    <td>no</td>\n",
"    <td>no</td>\n",
"    <td>good</td>\n",
"    <td>no</td>\n",
"    <td>no</td>\n",
"    <td>notckd</td>\n",
"  </tr>\n",
" </tbody>\n",
"</table>\n",
"<p>400 rows × 26 columns</p>\n",
"</div>\n",
"  <button class=\"colab-df-convert\"
onclick=\"convertToInteractive('df-1d93ffec-5973-41e1-a880-8612aaf0d727')\" \n",
"    title=\"Convert this dataframe to an interactive table.\" \n",
"    style=\"display:none;\">\n",
"  \n",
"  <svg xmlns=\"http://www.w3.org/2000/svg\"
height=\"24px\" viewBox=\"0 0 24 24\" \n",
"    width=\"24px\">\n",
"    <path d=\"M0 0h24v24H0V0z\" fill=\"none\"/>\n",

```

```
"    <path d=\"M18.56 5.44l.94 2.06.94-2.06 2.06-.94-2.06-.94-2.06-
.94 2.06-2.06.94zm-11 1L8.5 8.5l.94-2.06 2.06-.94-2.06-.94L8.5 2.5l-.94 2.06-
2.06.94zm10 10l.94 2.06.94-2.06 2.06-.94-2.06-.94-.94-2.06-.94 2.06-
2.06.94z\"/><path d=\"M17.41 7.96l-1.37-1.37c-.4-.4-.92-.59-1.43-.59-.52 0-
1.04.2-1.43.59L10.3 9.45l-7.72 7.72c-.78.78-.78 2.05 0 2.83L4 21.41c.39.39.59
1.41.59.51 0 1.02-.2 1.41-.59l7.78-7.78 2.81-2.81c.8-.78.8-2.07 0-2.86zM5.41
20L4 18.59l7.72-7.72 1.47 1.35L5.41 20z\"/>\n",
```

```
"  </svg>\n",
```

```
"    </button>\n",
```

```
"    \n",
```

```
"  <style>\n",
```

```
"    .colab-df-container {\n",
```

```
"      display: flex;\n",
```

```
"      flex-wrap: wrap;\n",
```

```
"      gap: 12px;\n",
```

```
"    }\n",
```

```
"\n",
```

```
"    .colab-df-convert {\n",
```

```
"      background-color: #E8F0FE;\n",
```

```
"      border: none;\n",
```

```
"      border-radius: 50%;\n",
```

```
"      cursor: pointer;\n",
```

```
"      display: none;\n",
```

```
"      fill: #1967D2;\n",
```

```
"      height: 32px;\n",
```

```
"      padding: 0 0 0 0;\n",
```

```
"      width: 32px;\n",
```

```
"  }\n",
"\n",
"  .colab-df-convert:hover {\n",
"    background-color: #E2EBFA;\n",
"    box-shadow: 0px 1px 2px rgba(60, 64, 67, 0.3), 0px 1px 3px 1px
rgba(60, 64, 67, 0.15);\n",
"    fill: #174EA6;\n",
"  }\n",
"\n",
"  [theme=dark] .colab-df-convert {\n",
"    background-color: #3B4455;\n",
"    fill: #D2E3FC;\n",
"  }\n",
"\n",
"  [theme=dark] .colab-df-convert:hover {\n",
"    background-color: #434B5C;\n",
"    box-shadow: 0px 1px 3px 1px rgba(0, 0, 0, 0.15);\n",
"    filter: drop-shadow(0px 1px 2px rgba(0, 0, 0, 0.3));\n",
"    fill: #FFFFFF;\n",
"  }\n",
" </style>\n",
"\n",
"  <script>\n",
"    const buttonEl =\n",
"      document.querySelector('#df-1d93ffec-5973-41e1-a880-
8612aaf0d727 button.colab-df-convert');\n",
```

```

"    buttonEl.style.display =\n",
"    google.colab.kernel.accessAllowed ? 'block' : 'none';\n",
"\n",
"    async function convertToInteractive(key) {\n",
"        const element = document.querySelector('#df-1d93ffec-5973-41e1-
a880-8612aaf0d727');\n",
"        const dataTable =\n",
"        await
google.colab.kernel.invokeFunction('convertToInteractive',\n",
"                                   [key], {});\n",
"        if (!dataTable) return;\n",
"\n",
"        const docLinkHtml = 'Like what you see? Visit the ' +\n",
"        '<a target=\"_blank\"
href=https://colab.research.google.com/notebooks/data_table.ipynb>data table
notebook</a>\n",
"        + ' to learn more about interactive tables.';\n",
"        element.innerHTML = ";\n",
"        dataTable['output_type'] = 'display_data';\n",
"        await google.colab.output.renderOutput(dataTable, element);\n",
"        const docLink = document.createElement('div');\n",
"        docLink.innerHTML = docLinkHtml;\n",
"        element.appendChild(docLink);\n",
"    }\n",
"    </script>\n",
"    </div>\n",
" </div>\n",

```

```
    " "
  ]
},
"metadata": {},
"execution_count": 12
}
]
},
{
  "cell_type": "code",
  "source": [
    "df.info()"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "wLfQMNI5iPTI",
    "outputId": "4851efeb-1955-4079-f5c5-618a61344ce7"
  },
  "execution_count": 13,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
```

```

"text": [
  "<class 'pandas.core.frame.DataFrame'>\n",
  "RangeIndex: 400 entries, 0 to 399\n",
  "Data columns (total 26 columns):\n",
  " #   Column          Non-Null Count  Dtype  \n",
  "---  -
  " 0   id              400 non-null   int64  \n",
  " 1   age             391 non-null   float64\n",
  " 2   bp              388 non-null   float64\n",
  " 3   sg              353 non-null   float64\n",
  " 4   al              354 non-null   float64\n",
  " 5   su              351 non-null   float64\n",
  " 6   rbc             248 non-null   object \n",
  " 7   pc              335 non-null   object \n",
  " 8   pcc             396 non-null   object \n",
  " 9   ba              396 non-null   object \n",
  "10   bgr             356 non-null   float64\n",
  "11   bu              381 non-null   float64\n",
  "12   sc              383 non-null   float64\n",
  "13   sod             313 non-null   float64\n",
  "14   pot             312 non-null   float64\n",
  "15   hemo            348 non-null   float64\n",
  "16   pcv             330 non-null   object \n",
  "17   wc              295 non-null   object \n",
  "18   rc              270 non-null   object \n",

```

```

" 19 htn      398 non-null  object \n",
" 20 dm       398 non-null  object \n",
" 21 cad      398 non-null  object \n",
" 22 appet    399 non-null  object \n",
" 23 pe       399 non-null  object \n",
" 24 ane      399 non-null  object \n",
" 25 classification 400 non-null  object \n",
"dtypes: float64(11), int64(1), object(14)\n",
"memory usage: 81.4+ KB\n"
]
}
]
},
{
  "cell_type": "code",
  "source": [
    "df.classification.unique()"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "ZE2wfOI0iPaZ",
    "outputId": "c6a0a9dd-30fd-4ebb-956c-c4f48da7099b"
  },

```

```

"execution_count": 14,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "array(['ckd', 'ckd\\t', 'notckd'], dtype=object)"
      ]
    },
    "metadata": {},
    "execution_count": 14
  }
],
{
  "cell_type": "code",
  "source": [
    "for col in df:\n",
    "    unique=df[col].value_counts()\n",
    "    print(unique,\\\"\\n===== \\n\\\")"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },

```



```
"id": "1zQ7Ct8ZiPg-",
"outputId": "0536ddef-a798-4298-d479-8770e7162b68"
},
"execution_count": 15,
"outputs": [
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "0    1\n",
      "263   1\n",
      "273   1\n",
      "272   1\n",
      "271   1\n",
      "    ..\n",
      "130   1\n",
      "129   1\n",
      "128   1\n",
      "127   1\n",
      "399   1\n",
      "Name: id, Length: 400, dtype: int64 \n",
      "===== \n",
      "\n",
      "60.0  19\n",
      "65.0  17\n
```

```
"48.0  12\n",
"55.0  12\n",
"50.0  12\n",
"      ..\n",
"83.0   1\n",
"27.0   1\n",
"14.0   1\n",
"81.0   1\n",
"79.0   1\n",
"Name: age, Length: 76, dtype: int64\n",
"===== \n",
"\n",
"80.0  116\n",
"70.0  112\n",
"60.0   71\n",
"90.0   53\n",
"100.0  25\n",
"50.0    5\n",
"110.0   3\n",
"140.0   1\n",
"180.0   1\n",
"120.0   1\n",
"Name: bp, dtype: int64\n",
"===== \n",
"\n",
```

"1.020 106\n",
"1.010 84\n",
"1.025 81\n",
"1.015 75\n",
"1.005 7\n",
"Name: sg, dtype: int64 \n",
"===== \n",
"\n",
"0.0 199\n",
"1.0 44\n",
"2.0 43\n",
"3.0 43\n",
"4.0 24\n",
"5.0 1\n",
"Name: al, dtype: int64 \n",
"===== \n",
"\n",
"0.0 290\n",
"2.0 18\n",
"3.0 14\n",
"4.0 13\n",
"1.0 13\n",
"5.0 3\n",
"Name: su, dtype: int64 \n",
"===== \n",

"\n",
"normal 201\n",
"abnormal 47\n",
"Name: rbc, dtype: int64 \n",
"===== \n",
"\n",
"normal 259\n",
"abnormal 76\n",
"Name: pc, dtype: int64 \n",
"===== \n",
"\n",
"notpresent 354\n",
"present 42\n",
"Name: pcc, dtype: int64 \n",
"===== \n",
"\n",
"notpresent 374\n",
"present 22\n",
"Name: ba, dtype: int64 \n",
"===== \n",
"\n",
"99.0 10\n",
"93.0 9\n",
"100.0 9\n",
"107.0 8\n",

"131.0 6\n",
" ..\n",
"288.0 1\n",
"182.0 1\n",
"84.0 1\n",
"256.0 1\n",
"226.0 1\n",
"Name: bgr, Length: 146, dtype: int64 \n",
"=====\n",
"\n",
"46.0 15\n",
"25.0 13\n",
"19.0 11\n",
"40.0 10\n",
"50.0 9\n",
" ..\n",
"176.0 1\n",
"145.0 1\n",
"92.0 1\n",
"322.0 1\n",
"186.0 1\n",
"Name: bu, Length: 118, dtype: int64 \n",
"=====\n",
"\n",
"1.2 40\n",

"1.1 24\n",
"0.5 23\n",
"1.0 23\n",
"0.9 22\n",
" ..\n",
"3.8 1\n",
"12.2 1\n",
"9.2 1\n",
"13.8 1\n",
"0.4 1\n",
"Name: sc, Length: 84, dtype: int64 \n",
"===== \n",
"\n",
"135.0 40\n",
"140.0 25\n",
"141.0 22\n",
"139.0 21\n",
"138.0 20\n",
"142.0 20\n",
"137.0 19\n",
"150.0 17\n",
"136.0 17\n",
"147.0 13\n",
"145.0 11\n",
"132.0 10\n",

```
"146.0  10\n",
"131.0   9\n",
"144.0   9\n",
"133.0   8\n",
"130.0   7\n",
"134.0   6\n",
"143.0   4\n",
"124.0   3\n",
"127.0   3\n",
"122.0   2\n",
"113.0   2\n",
"120.0   2\n",
"125.0   2\n",
"128.0   2\n",
"114.0   2\n",
"126.0   1\n",
"163.0   1\n",
"115.0   1\n",
"129.0   1\n",
"4.5     1\n",
"104.0   1\n",
"111.0   1\n",
"Name: sod, dtype: int64\n",
"===== \n",
"\n",
```

"3.5 30\n",
"5.0 30\n",
"4.9 27\n",
"4.7 17\n",
"4.8 16\n",
"3.9 14\n",
"3.8 14\n",
"4.1 14\n",
"4.2 14\n",
"4.0 14\n",
"4.4 14\n",
"4.5 13\n",
"4.3 12\n",
"3.7 12\n",
"3.6 8\n",
"4.6 7\n",
"3.4 5\n",
"5.2 5\n",
"5.3 4\n",
"5.7 4\n",
"3.2 3\n",
"5.5 3\n",
"6.3 3\n",
"5.4 3\n",
"2.9 3\n",

"3.3 3\n",
"5.6 2\n",
"3.0 2\n",
"6.5 2\n",
"2.5 2\n",
"5.9 2\n",
"5.8 2\n",
"7.6 1\n",
"47.0 1\n",
"6.6 1\n",
"5.1 1\n",
"6.4 1\n",
"2.8 1\n",
"2.7 1\n",
"39.0 1\n",
"Name: pot, dtype: int64 \n",
"===== \n",
"\n",
"15.0 16\n",
"10.9 8\n",
"13.6 7\n",
"13.0 7\n",
"9.8 7\n",
" ..\n",
"6.8 1\n",

"8.5 1\n",
"7.3 1\n",
"12.8 1\n",
"17.6 1\n",
"Name: hemo, Length: 115, dtype: int64 \n",
"===== \n",
"\n",
"41 21\n",
"52 21\n",
"44 19\n",
"48 19\n",
"40 16\n",
"43 14\n",
"42 13\n",
"45 13\n",
"32 12\n",
"36 12\n",
"33 12\n",
"50 12\n",
"28 12\n",
"34 11\n",
"37 11\n",
"30 9\n",
"29 9\n",
"35 9\n",

"46 9\n",
"31 8\n",
"24 7\n",
"39 7\n",
"26 6\n",
"38 5\n",
"53 4\n",
"51 4\n",
"49 4\n",
"47 4\n",
"54 4\n",
"25 3\n",
"27 3\n",
"22 3\n",
"19 2\n",
"23 2\n",
"15 1\n",
"21 1\n",
"17 1\n",
"20 1\n",
"\\t43 1\n",
"18 1\n",
"9 1\n",
"14 1\n",
"\\t? 1\n",

"16 1\n",
"Name: pcv, dtype: int64 \n",
"===== \n",
"\n",
"9800 11\n",
"6700 10\n",
"9200 9\n",
"9600 9\n",
"7200 9\n",
" ..\n",
"19100 1\n",
"\\t? 1\n",
"12300 1\n",
"14900 1\n",
"12700 1\n",
"Name: wc, Length: 92, dtype: int64 \n",
"===== \n",
"\n",
"5.2 18\n",
"4.5 16\n",
"4.9 14\n",
"4.7 11\n",
"4.8 10\n",
"3.9 10\n",
"4.6 9\n",

"3.4 9\n",
"5.9 8\n",
"5.5 8\n",
"6.1 8\n",
"5.0 8\n",
"3.7 8\n",
"5.3 7\n",
"5.8 7\n",
"5.4 7\n",
"3.8 7\n",
"5.6 6\n",
"4.3 6\n",
"4.2 6\n",
"3.2 5\n",
"4.4 5\n",
"5.7 5\n",
"6.4 5\n",
"5.1 5\n",
"6.2 5\n",
"6.5 5\n",
"4.1 5\n",
"3.6 4\n",
"6.3 4\n",
"6.0 4\n",
"4.0 3\n",

"3.3 3\n",
"4 3\n",
"3.5 3\n",
"2.9 2\n",
"3.1 2\n",
"2.6 2\n",
"2.1 2\n",
"2.5 2\n",
"2.8 2\n",
"3.0 2\n",
"2.7 2\n",
"5 2\n",
"2.3 1\n",
"\\t? 1\n",
"2.4 1\n",
"3 1\n",
"8.0 1\n",
"Name: rc, dtype: int64 \n",
"===== \n",
"\n",
"no 251\n",
"yes 147\n",
"Name: htn, dtype: int64 \n",
"===== \n",
"\n",

"no 258\n",
"yes 134\n",
"\\tno 3\n",
"\\tyes 2\n",
" yes 1\n",
"Name: dm, dtype: int64 \n",
"===== \n",
"\n",
"no 362\n",
"yes 34\n",
"\\tno 2\n",
"Name: cad, dtype: int64 \n",
"===== \n",
"\n",
"good 317\n",
"poor 82\n",
"Name: appet, dtype: int64 \n",
"===== \n",
"\n",
"no 323\n",
"yes 76\n",
"Name: pe, dtype: int64 \n",
"===== \n",
"\n",
"no 339\n",

```

"yes    60\n",
"Name: ane, dtype: int64 \n",
"===== \n",
"\n",
"ckd    248\n",
"notckd  150\n",
"ckd\\t    2\n",
"Name: classification, dtype: int64 \n",
"===== \n",
"\n"
]
}
]
},
{
"cell_type": "code",
"source": [
"# cleaning 'PCV'\n",
"df['pcv']=df['pcv'].apply(lambda x:x if type(x)==type(3.5) else
x.replace('\\t43','43').replace('\\t?','Nan'))\n",
"\n",
"# cleaning \"WC\"\n",
"df['wc']=df['wc'].apply(lambda x:x if type(x)==type(3.5) else
x.replace('\\t?','Nan').replace('\\t6200','6200').replace('\\t8400','8400'))\n",
"\n",
"# cleaning \"RC\"\n",

```



```

    "df['rc']=df['rc'].apply(lambda x:x if type(x)==type(3.5) else
x.replace('\t?','Nan'))\n",
    "\n",
    "# cleaning \"dm\"\n",
    "df['dm']=df['dm'].apply(lambda x:x if type(x)==type(3.5) else
x.replace('\tno','no').replace('\tyes','yes').replace(' yes','yes'))\n",
    "\n",
    "# cleaning \"CAD\"\n",
    "df['cad']=df['cad'].apply(lambda x:x if type(x)==type(3.5) else
x.replace('\tno','no'))\n",
    "\n",
    "# cleaning \"Classification\"\n",
    "df['classification']=df['classification'].apply(lambda x:x if type(x)==type(3.5)
else x.replace('ckd\t','ckd'))"
],
"metadata": {
    "id": "XIU-2-0siPn1"
},
"execution_count": 16,
"outputs": []
},
{
    "cell_type": "code",
    "source": [
        "df.drop('id',axis=1,inplace=True)"
    ],
    "metadata": {

```

```
    "id": "67EJvZ3-iPuk"
  },
  "execution_count": 17,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "mistyped=[[ 'pcv','rc','wc']]\n",
    "for i in mistyped:\n",
    "    df[i]=df[i].astype('float')",
  ],
  "metadata": {
    "id": "dF4AbUSQord-"
  },
  "execution_count": 18,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "cat_cols=list(df.select_dtypes('object'))\n",
    "cat_cols"
  ],
  "metadata": {
```

```
"colab": {
  "base_uri": "https://localhost:8080/"
},
"id": "ciHoMeS3orl8",
"outputId": "71b7067f-91cd-4552-a9bd-aca86cc76e9a"
},
"execution_count": 19,
"outputs": [
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [
        "['rbc',\n",
        " 'pc',\n",
        " 'pcc',\n",
        " 'ba',\n",
        " 'htn',\n",
        " 'dm',\n",
        " 'cad',\n",
        " 'appet',\n",
        " 'pe',\n",
        " 'ane',\n",
        " 'classification']"
      ]
    },
  },
]
```

```
    "metadata": {},
    "execution_count": 19
  }
]
},
{
  "cell_type": "code",
  "source": [
    "num_cols=list(df.select_dtypes(['int64','float64']))\n",
    "num_cols"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "7eIXia8uortp",
    "outputId": "67ec9565-724a-498e-c6fb-898105bbd44a"
  },
  "execution_count": 20,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "['age',\n",
```

```
    " 'bp',\n",
    " 'sg',\n",
    " 'al',\n",
    " 'su',\n",
    " 'bgr',\n",
    " 'bu',\n",
    " 'sc',\n",
    " 'sod',\n",
    " 'pot',\n",
    " 'hemo',\n",
    " 'pcv',\n",
    " 'wc',\n",
    " 'rc']"
  ]
},
"metadata": {},
"execution_count": 20
}
]
},
{
  "cell_type": "markdown",
  "source": [
    "***Handling Missing Values***"
  ],
}
```

```
"metadata": {
  "id": "hUB64ri9paB5"
},
{
  "cell_type": "code",
  "source": [
    "# Checking missing/Nan values\n",
    "df.isnull().sum().sort_values(ascending=False)"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "SmzakNsHor1P",
    "outputId": "6f5cac11-f337-4878-d21c-164f96a8b79c"
  },
  "execution_count": 21,
  "outputs": [
    {
      "output_type": "execute_result",
      "data": {
        "text/plain": [
          "rbc      152\n",
          "rc      131\n",
```

```
"wc          106\n",  
"pot         88\n",  
"sod         87\n",  
"pcv         71\n",  
"pc          65\n",  
"hemo        52\n",  
"su          49\n",  
"sg          47\n",  
"al          46\n",  
"bgr         44\n",  
"bu          19\n",  
"sc          17\n",  
"bp          12\n",  
"age         9\n",  
"ba          4\n",  
"pcc         4\n",  
"htn         2\n",  
"dm          2\n",  
"cad         2\n",  
"appet       1\n",  
"pe          1\n",  
"ane         1\n",  
"classification 0\n",  
"dtype: int64"
```

```
]
```

```
    },
    "metadata": {},
    "execution_count": 21
  }
]
},
{
  "cell_type": "markdown",
  "source": [
    "*****"
  ],
  "metadata": {
    "id": "Rb62gGlKplv2"
  }
},
{
  "cell_type": "code",
  "source": [],
  "metadata": {
    "id": "SdYx4uPTor8P"
  },
  "execution_count": 20,
  "outputs": []
},
{
```



```
"cell_type": "code",
"source": [],
"metadata": {
  "id": "bvPZgjaBosDh"
},
"execution_count": 20,
"outputs": []
}
]
}
```