**Poorvi\_Raut\_HOW2X\_CHATGPT**

**Poorvi Raut: 20009560**

**Output of the code generated by CHATGPT in python Programming Language using Jupyter Notebook**

**pip install pandas**

**pip install numpy**

**pip install matplotlib**

**import pandas as pd**

**#1. Read in the CSV file**

**data = pd.read\_csv('/Users/Owner/Desktop/Spring 2023/CS 513 KDD/breast-cancer-wisconsin.csv')**

**# 1.1 Summarize each column**

**print(data.describe()**

**Sample F1 F2 F3 F4 \**

**count 6.990000e+02 699.000000 699.000000 699.000000 699.000000**

**mean 1.071704e+06 4.417740 3.134478 3.207439 2.806867**

**std 6.170957e+05 2.815741 3.051459 2.971913 2.855379**

**min 6.163400e+04 1.000000 1.000000 1.000000 1.000000**

**25% 8.706885e+05 2.000000 1.000000 1.000000 1.000000**

**50% 1.171710e+06 4.000000 1.000000 1.000000 1.000000**

**75% 1.238298e+06 6.000000 5.000000 5.000000 4.000000**

**max 1.345435e+07 10.000000 10.000000 10.000000 10.000000**

**F5 F7 F8 F9 Class**

**count 699.000000 699.000000 699.000000 699.000000 699.000000**

**mean 3.216023 3.437768 2.866953 1.589413 2.689557**

**std 2.214300 2.438364 3.053634 1.715078 0.951273**

**min 1.000000 1.000000 1.000000 1.000000 2.000000**

**25% 2.000000 2.000000 1.000000 1.000000 2.000000**

**50% 2.000000 3.000000 1.000000 1.000000 2.000000**

**75% 4.000000 5.000000 4.000000 1.000000 4.000000**

**max 10.000000 10.000000 10.000000 10.000000 4.000000**

**#1.2 Check for missing values**

**print(data.isnull().sum())**

**Sample 0**

**F1 0**

**F2 0**

**F3 0**

**F4 0**

**F5 0**

**F6 0**

**F7 0**

**F8 0**

**F9 0**

**Class 0**

**dtype: int64**

**# 1.3 Replace missing values with the mean of the column**

**data.fillna(data.mean(), inplace=True)**

**Output**

**Sample F1 F2 F3 F4 F5 F6 F7 F8 F9 Class**

**1 1000025 5 1 1 1 2 1.000 3 1 1 2**

**2 1002945 5 4 4 5 7 10.000 3 2 1 2**

**3 1015425 3 1 1 1 2 2.000 3 1 1 2**

**4 1016277 6 8 8 1 3 4.000 3 7 1 2**

**5 1017023 4 1 1 3 2 1.000 3 1 1 2**

**6 1017122 8 10 10 8 7 10.000 9 7 1 4**

**7 1018099 1 1 1 1 2 10.000 3 1 1 2**

**8 1018561 2 1 2 1 2 1.000 3 1 1 2**

**9 1033078 2 1 1 1 2 1.000 1 1 5 2**

**10 1033078 4 2 1 1 2 1.000 2 1 1 2**

**11 1035283 1 1 1 1 1 1.000 3 1 1 2**

**12 1036172 2 1 1 1 2 1.000 2 1 1 2**

**13 1041801 5 3 3 3 2 3.000 4 4 1 4**

**14 1043999 1 1 1 1 2 3.000 3 1 1 2**

**15 1044572 8 7 5 10 7 9.000 5 5 4 4**

**16 1047630 7 4 6 4 6 1.000 4 3 1 4**

**17 1048672 4 1 1 1 2 1.000 2 1 1 2**

**18 1049815 4 1 1 1 2 1.000 3 1 1 2**

**19 1050670 10 7 7 6 4 10.000 4 1 2 4**

**20 1050718 6 1 1 1 2 1.000 3 1 1 2**

**21 1054590 7 3 2 10 5 10.000 5 4 4 4**

**22 1054593 10 5 5 3 6 7.000 7 10 1 4**

**23 1056784 3 1 1 1 2 1.000 2 1 1 2**

**24 1057013 8 4 5 1 2 3.545 7 3 1 4**

**25 1059552 1 1 1 1 2 1.000 3 1 1 2**

**26 1065726 5 2 3 4 2 7.000 3 6 1 4**

**27 1066373 3 2 1 1 1 1.000 2 1 1 2**

**28 1066979 5 1 1 1 2 1.000 2 1 1 2**

**29 1067444 2 1 1 1 2 1.000 2 1 1 2**

**30 1070935 1 1 3 1 2 1.000 1 1 1 2**

**31 1070935 3 1 1 1 1 1.000 2 1 1 2**

**32 1071760 2 1 1 1 2 1.000 3 1 1 2**

**33 1072179 10 7 7 3 8 5.000 7 4 3 4**

**34 1074610 2 1 1 2 2 1.000 3 1 1 2**

**35 1075123 3 1 2 1 2 1.000 2 1 1 2**

**36 1079304 2 1 1 1 2 1.000 2 1 1 2**

**37 1080185 10 10 10 8 6 1.000 8 9 1 4**

**38 1081791 6 2 1 1 1 1.000 7 1 1 2**

**39 1084584 5 4 4 9 2 10.000 5 6 1 4**

**40 1091262 2 5 3 3 6 7.000 7 5 1 4**

**41 1096800 6 6 6 9 6 3.545 7 8 1 2**

**42 1099510 10 4 3 1 3 3.000 6 5 2 4**

**43 1100524 6 10 10 2 8 10.000 7 3 3 4**

**44 1102573 5 6 5 6 10 1.000 3 1 1 4**

**45 1103608 10 10 10 4 8 1.000 8 10 1 4**

**46 1103722 1 1 1 1 2 1.000 2 1 2 2**

**47 1105257 3 7 7 4 4 9.000 4 8 1 4**

**48 1105524 1 1 1 1 2 1.000 2 1 1 2**

**49 1106095 4 1 1 3 2 1.000 3 1 1 2**

**50 1106829 7 8 7 2 4 8.000 3 8 2 4**

**51 1108370 9 5 8 1 2 3.000 2 1 5 4**

**52 1108449 5 3 3 4 2 4.000 3 4 1 4**

**53 1110102 10 3 6 2 3 5.000 4 10 2 4**

**54 1110503 5 5 5 8 10 8.000 7 3 7 4**

**55 1110524 10 5 5 6 8 8.000 7 1 1 4**

**56 1111249 10 6 6 3 4 5.000 3 6 1 4**

**57 1112209 8 10 10 1 3 6.000 3 9 1 4**

**58 1113038 8 2 4 1 5 1.000 5 4 4 4**

**59 1113483 5 2 3 1 6 10.000 5 1 1 4**

**60 1113906 9 5 5 2 2 2.000 5 1 1 4**

**61 1115282 5 3 5 5 3 3.000 4 10 1 4**

**62 1115293 1 1 1 1 2 2.000 2 1 1 2**

**63 1116116 9 10 10 1 10 8.000 3 3 1 4**

**64 1116132 6 3 4 1 5 2.000 3 9 1 4**

**65 1116192 1 1 1 1 2 1.000 2 1 1 2**

**66 1116998 10 4 2 1 3 2.000 4 3 10 4**

**67 1117152 4 1 1 1 2 1.000 3 1 1 2**

**68 1118039 5 3 4 1 8 10.000 4 9 1 4**

**69 1120559 8 3 8 3 4 9.000 8 9 8 4**

**70 1121732 1 1 1 1 2 1.000 3 2 1 2**

**71 1121919 5 1 3 1 2 1.000 2 1 1 2**

**72 1123061 6 10 2 8 10 2.000 7 8 10 4**

**73 1124651 1 3 3 2 2 1.000 7 2 1 2**

**74 1125035 9 4 5 10 6 10.000 4 8 1 4**

**75 1126417 10 6 4 1 3 4.000 3 2 3 4**

**76 1131294 1 1 2 1 2 2.000 4 2 1 2**

**77 1132347 1 1 4 1 2 1.000 2 1 1 2**

**78 1133041 5 3 1 2 2 1.000 2 1 1 2**

**79 1133136 3 1 1 1 2 3.000 3 1 1 2**

**80 1136142 2 1 1 1 3 1.000 2 1 1 2**

**81 1137156 2 2 2 1 1 1.000 7 1 1 2**

**82 1143978 4 1 1 2 2 1.000 2 1 1 2**

**83 1143978 5 2 1 1 2 1.000 3 1 1 2**

**84 1147044 3 1 1 1 2 2.000 7 1 1 2**

**85 1147699 3 5 7 8 8 9.000 7 10 7 4**

**86 1147748 5 10 6 1 10 4.000 4 10 10 4**

**87 1148278 3 3 6 4 5 8.000 4 4 1 4**

**88 1148873 3 6 6 6 5 10.000 6 8 3 4**

**89 1152331 4 1 1 1 2 1.000 3 1 1 2**

**90 1155546 2 1 1 2 3 1.000 2 1 1 2**

**# Display frequency table of "Class" vs. F6**

**print(pd.crosstab(data['Class'], data['F6']))**

**F6 1 10 2 3 4 5 6 7 8 9 ?**

**Class**

**2 387 3 21 14 6 10 0 1 2 0 14**

**4 15 129 9 14 13 20 4 7 19 9 2**

**#1.4 Create scatter matrix of F1 to F6**

**pd.plotting.scatter\_matrix(data[['F1', 'F2', 'F3', 'F4', 'F5', 'F6']])**

**plt.show()**

**Chart

Description automatically generated**

**# Create histogram of F7 to F9**

**data[['F7', 'F8', 'F9']].hist()**

**plt.show()**

**# Create box plot of F7 to F9**

**data [['F7', 'F8', 'F9']].boxplot()**

**plt.show()**

**Chart, box and whisker chart

Description automatically generated**

**#2.1 delete all objects in Python environment**

**del data**

**# 2.2 read in the CSV file.**

**data = pd.read\_csv('/Users/Owner/Desktop/Spring 2023/CS 513 KDD/breast-cancer-wisconsin.csv')**

**# Remove any rows with missing values**

**data.dropna(inplace=True)**