rinex-majorproject1

August 26, 2024

1 MAJOR PROJECT 1

[ ]:
```
Diagnosis - KNN Classifier
```

[1]:
```
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

[9]:
```
df=pd.read_csv('C:
↪\\Users\\moham\\OneDrive\\Desktop\\Documents\\KNNAlgorithmDataset.csv')
```

[10]:
```
df.head()
```

[10]:

	id	diagnosis	radius_mean	texture_mean	perimeter_mean	area_mean	\
0	842302	M	17.99	10.38	122.80	1001.0	
1	842517	M	20.57	17.77	132.90	1326.0	
2	84300903	M	19.69	21.25	130.00	1203.0	
3	84348301	M	11.42	20.38	77.58	386.1	
4	84358402	M	20.29	14.34	135.10	1297.0	

	smoothness_mean	compactness_mean	concavity_mean	concave points_mean	\
0	0.11840	0.27760	0.3001	0.14710	
1	0.08474	0.07864	0.0869	0.07017	
2	0.10960	0.15990	0.1974	0.12790	
3	0.14250	0.28390	0.2414	0.10520	
4	0.10030	0.13280	0.1980	0.10430	

	…	radius_worst	texture_worst	perimeter_worst	area_worst	\
0	…	25.38	17.33	184.60	2019.0	
1	…	24.99	23.41	158.80	1956.0	
2	…	23.57	25.53	152.50	1709.0	
3	…	14.91	26.50	98.87	567.7	
4	…	22.54	16.67	152.20	1575.0	

	smoothness_worst	compactness_worst	concavity_worst	concave points_worst	\
0	0.1622	0.6656	0.7119	0.2654	

```
1            0.1238           0.1866            0.2416                0.1860
2            0.1444           0.4245            0.4504                0.2430
3            0.2098           0.8663            0.6869                0.2575
4            0.1374           0.2050            0.4000                0.1625

   symmetry_worst  fractal_dimension_worst
0          0.4601                  0.11890
1          0.2750                  0.08902
2          0.3613                  0.08758
3          0.6638                  0.17300
4          0.2364                  0.07678

[5 rows x 32 columns]
```

[11]:
```
print(df.columns)
```

```
Index(['id', 'diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst'],
      dtype='object')
```

[13]:
```
# Strip any whitespace from column names
df.columns = df.columns.str.strip()

# Safely drop the 'Unnamed: 32' column if it exists
df = df.drop('Unnamed: 32', axis=1, errors='ignore')

# Safely drop the 'id' column if it exists
df = df.drop('id', axis=1, errors='ignore')

# Display the first five rows of the DataFrame
df.head()
```

[13]:
```
  diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0         M        17.99         10.38          122.80     1001.0
1         M        20.57         17.77          132.90     1326.0
2         M        19.69         21.25          130.00     1203.0
3         M        11.42         20.38           77.58      386.1
4         M        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
```

```
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
0         0.2419  …         25.38          17.33           184.60
1         0.1812  …         24.99          23.41           158.80
2         0.2069  …         23.57          25.53           152.50
3         0.2597  …         14.91          26.50            98.87
4         0.1809  …         22.54          16.67           152.20

   area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   concave points_worst  symmetry_worst  fractal_dimension_worst
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678

[5 rows x 31 columns]
```

[14]:
```
df.columns
```

[14]:
```
Index(['diagnosis', 'radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst'],
      dtype='object')
```

[15]:
```
unique_classes = df['diagnosis'].unique()
print(f"Unique classes in 'diagnosis': {unique_classes}")
```

```
Unique classes in 'diagnosis': ['M' 'B']
```

[16]:
```
df['diagnosis'] = df['diagnosis'].replace({'M': 1, 'B': 0})
print(df.head())
```

```
   diagnosis  radius_mean  texture_mean  perimeter_mean  area_mean  \
0          1        17.99         10.38          122.80     1001.0
1          1        20.57         17.77          132.90     1326.0
2          1        19.69         21.25          130.00     1203.0
3          1        11.42         20.38           77.58      386.1
4          1        20.29         14.34          135.10     1297.0

   smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
0          0.11840           0.27760          0.3001              0.14710
1          0.08474           0.07864          0.0869              0.07017
2          0.10960           0.15990          0.1974              0.12790
3          0.14250           0.28390          0.2414              0.10520
4          0.10030           0.13280          0.1980              0.10430

   symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
0         0.2419  …         25.38          17.33           184.60
1         0.1812  …         24.99          23.41           158.80
2         0.2069  …         23.57          25.53           152.50
3         0.2597  …         14.91          26.50            98.87
4         0.1809  …         22.54          16.67           152.20

   area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   concave points_worst  symmetry_worst  fractal_dimension_worst
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678

[5 rows x 31 columns]
```

[17]:
```
# Preparing feature set
X = df[['radius_mean', 'texture_mean', 'perimeter_mean',
       'area_mean', 'smoothness_mean', 'compactness_mean', 'concavity_mean',
       'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
       'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
       'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
       'fractal_dimension_se', 'radius_worst', 'texture_worst',
```

```
       'perimeter_worst', 'area_worst', 'smoothness_worst',
       'compactness_worst', 'concavity_worst', 'concave points_worst',
       'symmetry_worst', 'fractal_dimension_worst']]
Y = df[['diagnosis']]
print(X[0:5])
print(Y[0:5])
```

```
   radius_mean  texture_mean  perimeter_mean  area_mean  smoothness_mean  \
0        17.99         10.38          122.80     1001.0          0.11840
1        20.57         17.77          132.90     1326.0          0.08474
2        19.69         21.25          130.00     1203.0          0.10960
3        11.42         20.38           77.58      386.1          0.14250
4        20.29         14.34          135.10     1297.0          0.10030

   compactness_mean  concavity_mean  concave points_mean  symmetry_mean  \
0           0.27760          0.3001              0.14710         0.2419
1           0.07864          0.0869              0.07017         0.1812
2           0.15990          0.1974              0.12790         0.2069
3           0.28390          0.2414              0.10520         0.2597
4           0.13280          0.1980              0.10430         0.1809

   fractal_dimension_mean  …  radius_worst  texture_worst  perimeter_worst  \
0                 0.07871  …         25.38          17.33           184.60
1                 0.05667  …         24.99          23.41           158.80
2                 0.05999  …         23.57          25.53           152.50
3                 0.09744  …         14.91          26.50            98.87
4                 0.05883  …         22.54          16.67           152.20

   area_worst  smoothness_worst  compactness_worst  concavity_worst  \
0      2019.0            0.1622             0.6656           0.7119
1      1956.0            0.1238             0.1866           0.2416
2      1709.0            0.1444             0.4245           0.4504
3       567.7            0.2098             0.8663           0.6869
4      1575.0            0.1374             0.2050           0.4000

   concave points_worst  symmetry_worst  fractal_dimension_worst
0                0.2654          0.4601                  0.11890
1                0.1860          0.2750                  0.08902
2                0.2430          0.3613                  0.08758
3                0.2575          0.6638                  0.17300
4                0.1625          0.2364                  0.07678

[5 rows x 30 columns]
   diagnosis
0          1
1          1
2          1
```

```
3          1
4          1
```

[18]:
```
print(df.describe())
```

```
        diagnosis  radius_mean  texture_mean  perimeter_mean    area_mean  \
count  569.000000   569.000000    569.000000      569.000000   569.000000
mean     0.372583    14.127292     19.289649       91.969033   654.889104
std      0.483918     3.524049      4.301036       24.298981   351.914129
min      0.000000     6.981000      9.710000       43.790000   143.500000
25%      0.000000    11.700000     16.170000       75.170000   420.300000
50%      0.000000    13.370000     18.840000       86.240000   551.100000
75%      1.000000    15.780000     21.800000      104.100000   782.700000
max      1.000000    28.110000     39.280000      188.500000  2501.000000

       smoothness_mean  compactness_mean  concavity_mean  concave points_mean  \
count       569.000000        569.000000      569.000000           569.000000
mean          0.096360          0.104341        0.088799             0.048919
std           0.014064          0.052813        0.079720             0.038803
min           0.052630          0.019380        0.000000             0.000000
25%           0.086370          0.064920        0.029560             0.020310
50%           0.095870          0.092630        0.061540             0.033500
75%           0.105300          0.130400        0.130700             0.074000
max           0.163400          0.345400        0.426800             0.201200

       symmetry_mean  …  radius_worst  texture_worst  perimeter_worst  \
count     569.000000  …    569.000000     569.000000       569.000000
mean        0.181162  …     16.269190      25.677223       107.261213
std         0.027414  …      4.833242       6.146258        33.602542
min         0.106000  …      7.930000      12.020000        50.410000
25%         0.161900  …     13.010000      21.080000        84.110000
50%         0.179200  …     14.970000      25.410000        97.660000
75%         0.195700  …     18.790000      29.720000       125.400000
max         0.304000  …     36.040000      49.540000       251.200000

        area_worst  smoothness_worst  compactness_worst  concavity_worst  \
count   569.000000        569.000000         569.000000       569.000000
mean    880.583128          0.132369           0.254265         0.272188
std     569.356993          0.022832           0.157336         0.208624
min     185.200000          0.071170           0.027290         0.000000
25%     515.300000          0.116600           0.147200         0.114500
50%     686.500000          0.131300           0.211900         0.226700
75%    1084.000000          0.146000           0.339100         0.382900
max    4254.000000          0.222600           1.058000         1.252000

       concave points_worst  symmetry_worst  fractal_dimension_worst
count            569.000000      569.000000               569.000000
mean               0.114606        0.290076                 0.083946
```

```
std                 0.065732        0.061867                0.018061
min                 0.000000        0.156500                0.055040
25%                 0.064930        0.250400                0.071460
50%                 0.099930        0.282200                0.080040
75%                 0.161400        0.317900                0.092080
max                 0.291000        0.663800                0.207500

[8 rows x 31 columns]
```

[19]:
```
import matplotlib.pyplot as plt
from sklearn import preprocessing
# Preprocessing
X = preprocessing.StandardScaler().fit(X).transform(X.astype(float))
X[0:5]
```

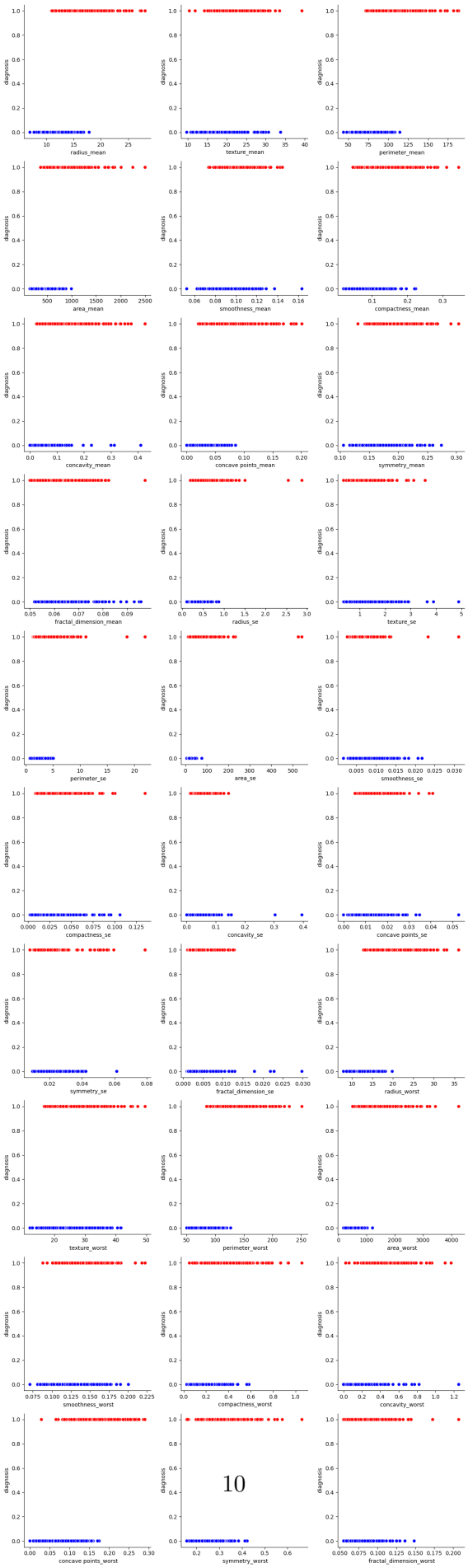
[19]:
```
array([[ 1.09706398e+00, -2.07333501e+00,  1.26993369e+00,
         9.84374905e-01,  1.56846633e+00,  3.28351467e+00,
         2.65287398e+00,  2.53247522e+00,  2.21751501e+00,
         2.25574689e+00,  2.48973393e+00, -5.65265059e-01,
         2.83303087e+00,  2.48757756e+00, -2.14001647e-01,
         1.31686157e+00,  7.24026158e-01,  6.60819941e-01,
         1.14875667e+00,  9.07083081e-01,  1.88668963e+00,
        -1.35929347e+00,  2.30360062e+00,  2.00123749e+00,
         1.30768627e+00,  2.61666502e+00,  2.10952635e+00,
         2.29607613e+00,  2.75062224e+00,  1.93701461e+00],
       [ 1.82982061e+00, -3.53632408e-01,  1.68595471e+00,
         1.90870825e+00, -8.26962447e-01, -4.87071673e-01,
        -2.38458552e-02,  5.48144156e-01,  1.39236330e-03,
        -8.68652457e-01,  4.99254601e-01, -8.76243603e-01,
         2.63326966e-01,  7.42401948e-01, -6.05350847e-01,
        -6.92926270e-01, -4.40780058e-01,  2.60162067e-01,
        -8.05450380e-01, -9.94437403e-02,  1.80592744e+00,
        -3.69203222e-01,  1.53512599e+00,  1.89048899e+00,
        -3.75611957e-01, -4.30444219e-01, -1.46748968e-01,
         1.08708430e+00, -2.43889668e-01,  2.81189987e-01],
       [ 1.57988811e+00,  4.56186952e-01,  1.56650313e+00,
         1.55888363e+00,  9.42210440e-01,  1.05292554e+00,
         1.36347845e+00,  2.03723076e+00,  9.39684817e-01,
        -3.98007910e-01,  1.22867595e+00, -7.80083377e-01,
         8.50928301e-01,  1.18133606e+00, -2.97005012e-01,
         8.14973504e-01,  2.13076435e-01,  1.42482747e+00,
         2.37035535e-01,  2.93559404e-01,  1.51187025e+00,
        -2.39743838e-02,  1.34747521e+00,  1.45628455e+00,
         5.27407405e-01,  1.08293217e+00,  8.54973944e-01,
         1.95500035e+00,  1.15225500e+00,  2.01391209e-01],
       [-7.68909287e-01,  2.53732112e-01, -5.92687167e-01,
        -7.64463792e-01,  3.28355348e+00,  3.40290899e+00,
```

```
         1.91589718e+00,  1.45170736e+00,  2.86738293e+00,
         4.91091929e+00,  3.26373441e-01, -1.10409044e-01,
         2.86593405e-01, -2.88378148e-01,  6.89701660e-01,
         2.74428041e+00,  8.19518384e-01,  1.11500701e+00,
         4.73268037e+00,  2.04751088e+00, -2.81464464e-01,
         1.33984094e-01, -2.49939304e-01, -5.50021228e-01,
         3.39427470e+00,  3.89339743e+00,  1.98958826e+00,
         2.17578601e+00,  6.04604135e+00,  4.93501034e+00],
       [ 1.75029663e+00, -1.15181643e+00,  1.77657315e+00,
         1.82622928e+00,  2.80371830e-01,  5.39340452e-01,
         1.37101143e+00,  1.42849277e+00, -9.56046689e-03,
        -5.62449981e-01,  1.27054278e+00, -7.90243702e-01,
         1.27318941e+00,  1.19035676e+00,  1.48306716e+00,
        -4.85198799e-02,  8.28470780e-01,  1.14420474e+00,
        -3.61092272e-01,  4.99328134e-01,  1.29857524e+00,
        -1.46677038e+00,  1.33853946e+00,  1.22072425e+00,
         2.20556166e-01, -3.13394511e-01,  6.13178758e-01,
         7.29259257e-01, -8.68352984e-01, -3.97099619e-01]])
```

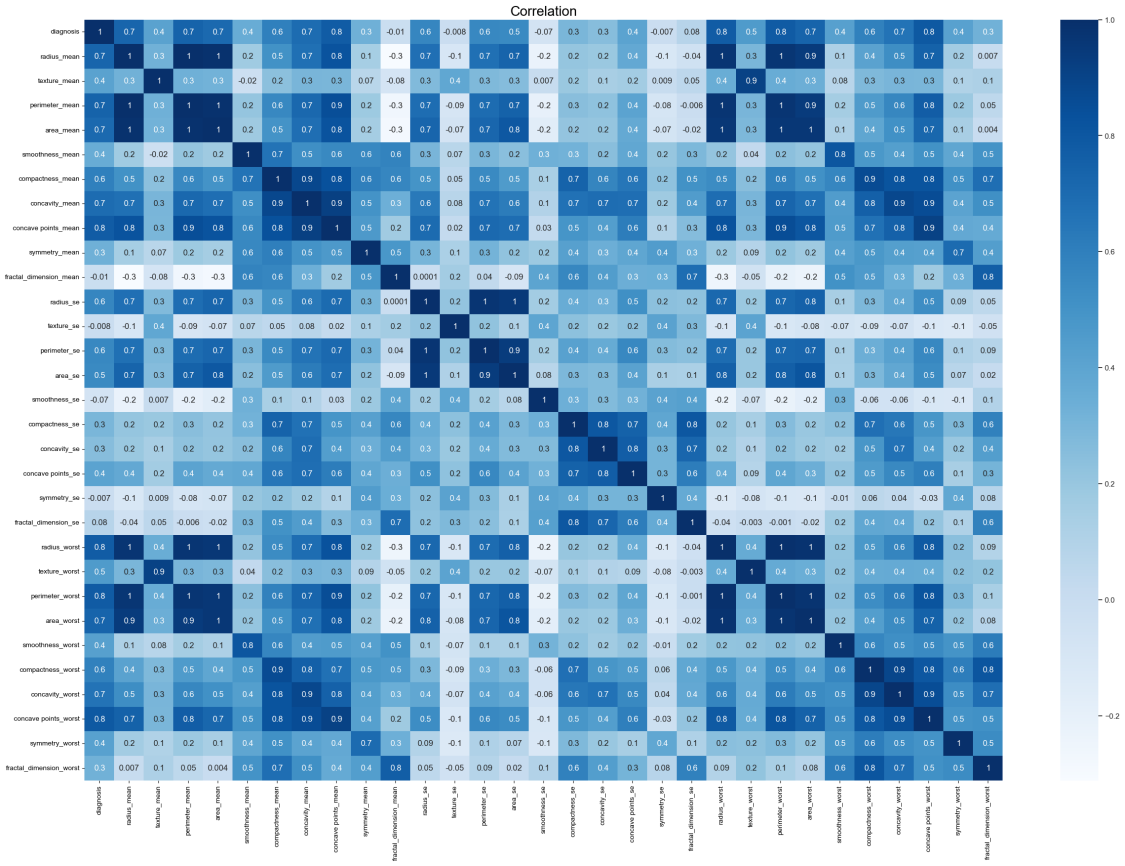
[20]:
```
import seaborn as sns
# Viz.
x_cols = ['radius_mean', 'texture_mean', 'perimeter_mean', 'area_mean',
          'smoothness_mean', 'compactness_mean', 'concavity_mean',
          'concave points_mean', 'symmetry_mean', 'fractal_dimension_mean',
          'radius_se', 'texture_se', 'perimeter_se', 'area_se', 'smoothness_se',
          'compactness_se', 'concavity_se', 'concave points_se', 'symmetry_se',
          'fractal_dimension_se', 'radius_worst', 'texture_worst',
          'perimeter_worst', 'area_worst', 'smoothness_worst',
          'compactness_worst', 'concavity_worst', 'concave points_worst',
          'symmetry_worst', 'fractal_dimension_worst']

melted_df = df.melt(id_vars="diagnosis", value_vars=x_cols)

g = sns.FacetGrid(melted_df, col="variable", col_wrap=3, sharex=False,␣
 ↪sharey=False, height=4)

def scatter_with_hue(data, **kws):
    sns.scatterplot(data=data, x="value", y="diagnosis", hue="diagnosis",␣
 ↪palette={0: "blue", 1: "red"}, legend=False, **kws)

g.map_dataframe(scatter_with_hue)


for ax in g.axes.flat:
    xlabel = ax.get_title().split('=')[1]
    ax.set_xlabel(xlabel)
    ax.set_title('')
```

```
plt.subplots_adjust(hspace=0.8)

g.tight_layout()
plt.show()
```

```
C:\Users\moham\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:
The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```


diagnosis
radius_mean
texture_mean
perimeter_mean
area_mean
smoothness_mean
compactness_mean
concavity_mean
concave points_mean
symmetry_mean
fractal_dimension_mean
radius_se
texture_se
perimeter_se
area_se
smoothness_se
compactness_se
concavity_se
concave points_se
symmetry_se
fractal_dimension_se
radius_worst
texture_worst
perimeter_worst
area_worst
smoothness_worst
compactness_worst
concavity_worst
concave points_worst
symmetry_worst
fractal_dimension_worst

[21]:
```
plt.figure(figsize=(30,20))
plt.title("Correlation", fontsize=20)
sns.set_theme(style="white")
corr = df.corr()
heatmap = sns.heatmap(corr, annot=True, cmap="Blues", fmt='.1g')
```



[22]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split( X, Y, test_size=0.2,␣
  ↪random_state=4)
print ('Train set:', X_train.shape,  y_train.shape)
print ('Test set:', X_test.shape,  y_test.shape)
```

```
Train set: (455, 30) (455, 1)
Test set: (114, 30) (114, 1)
```

[25]:
```
import numpy as np
from sklearn.neighbors import KNeighborsClassifier
from sklearn import metrics
```

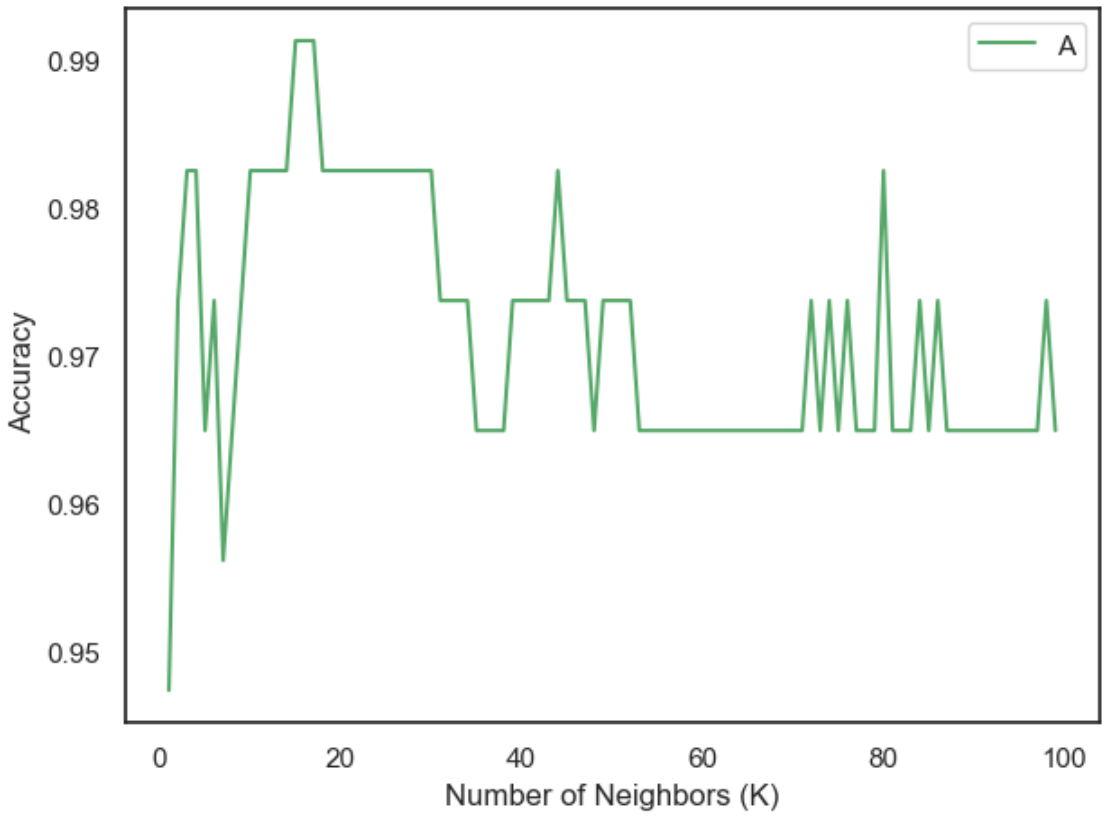
```
Ks = 100
mean_acc = np.zeros((Ks-1))

for n in range(1, Ks):
    # Train the model and predict
    neigh = KNeighborsClassifier(n_neighbors=n).fit(X_train, y_train.values.
 ↪ravel())
    yhat = neigh.predict(X_test)
    mean_acc[n-1] = metrics.accuracy_score(y_test, yhat)

# Display the mean accuracy scores for different values of K
mean_acc
```

[25]:
```
array([0.94736842, 0.97368421, 0.98245614, 0.98245614, 0.96491228,
       0.97368421, 0.95614035, 0.96491228, 0.97368421, 0.98245614,
       0.98245614, 0.98245614, 0.98245614, 0.98245614, 0.99122807,
       0.99122807, 0.99122807, 0.98245614, 0.98245614, 0.98245614,
       0.98245614, 0.98245614, 0.98245614, 0.98245614, 0.98245614,
       0.98245614, 0.98245614, 0.98245614, 0.98245614, 0.98245614,
       0.97368421, 0.97368421, 0.97368421, 0.97368421, 0.96491228,
       0.96491228, 0.96491228, 0.96491228, 0.97368421, 0.97368421,
       0.97368421, 0.97368421, 0.97368421, 0.98245614, 0.97368421,
       0.97368421, 0.97368421, 0.96491228, 0.97368421, 0.97368421,
       0.97368421, 0.97368421, 0.96491228, 0.96491228, 0.96491228,
       0.96491228, 0.96491228, 0.96491228, 0.96491228, 0.96491228,
       0.96491228, 0.96491228, 0.96491228, 0.96491228, 0.96491228,
       0.96491228, 0.96491228, 0.96491228, 0.96491228, 0.96491228,
       0.96491228, 0.97368421, 0.96491228, 0.97368421, 0.96491228,
       0.97368421, 0.96491228, 0.96491228, 0.96491228, 0.98245614,
       0.96491228, 0.96491228, 0.96491228, 0.97368421, 0.96491228,
       0.97368421, 0.96491228, 0.96491228, 0.96491228, 0.96491228,
       0.96491228, 0.96491228, 0.96491228, 0.96491228, 0.96491228,
       0.96491228, 0.96491228, 0.97368421, 0.96491228])
```

[26]:
```
print( "The best accuracy was with", mean_acc.max(), "with k=", mean_acc.
 ↪argmax()+1)
```

```
The best accuracy was with 0.9912280701754386 with k= 15
```

[27]:
```
plt.plot(range(1,Ks),mean_acc,'g')
plt.legend(('Accuracy '))
plt.ylabel('Accuracy ')
plt.xlabel('Number of Neighbors (K)')
plt.tight_layout()
plt.show()
```



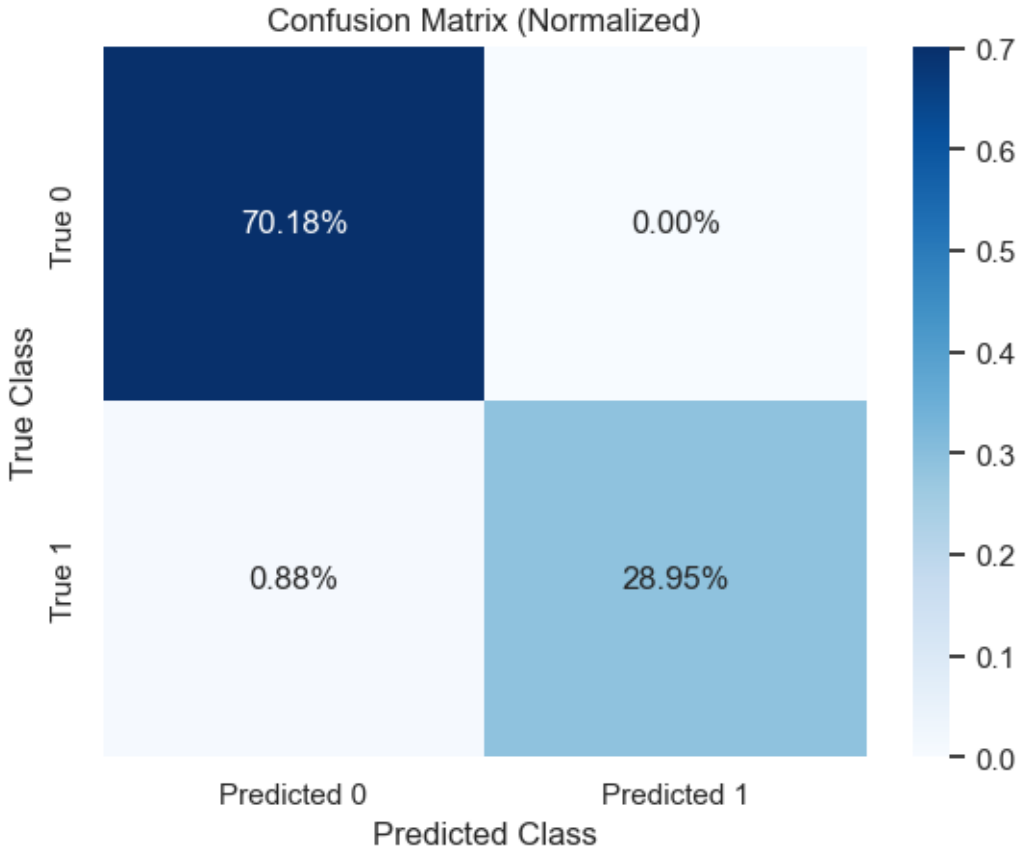
[28]:
```
final_k = 15
neigh = KNeighborsClassifier(n_neighbors = final_k).fit(X_train,y_train.values.
  ↪ravel())
yhat=neigh.predict(X_test)
```

[29]:
```
jaccard_score = metrics.jaccard_score(y_test, yhat)
f1_score = metrics.f1_score(y_test, yhat)
print(f"Jaccard score: {jaccard_score}\nF1_Score: {f1_score}")
```

```
Jaccard score: 0.9705882352941176
F1_Score: 0.9850746268656716
```

[30]:
```
cf_matrix = metrics.confusion_matrix(y_test, yhat)

normalized_cf_matrix = cf_matrix / np.sum(cf_matrix)

sns.heatmap(normalized_cf_matrix, annot=True, fmt='.2%', cmap='Blues',
            xticklabels=['Predicted ' + str(label) for label in np.
  ↪unique(yhat)],
            yticklabels=['True ' + str(label) for label in np.unique(y_test)])
```

```
plt.xlabel('Predicted Class')
plt.ylabel('True Class')

plt.title('Confusion Matrix (Normalized)')
plt.show()
```



[ ]: