Portfolio Week 2

COS40007 - Artificial Intelligence for Engineering

Studio 1-1 (12:30-2:30)

Hanh Nguyen Nguyen
103532674

Studio 1 Dataset Selection

The dataset chosen for this studio is the **Breast Cancer diagnosis**. The student is a Biomedical Engineering major and this dataset provides appropriate and relevant information regarding the major. The dataset provides access to real life cases, assisting students familiarize with the dataset they may encountered in the future.
Water quality is also considered since Biomedical and Environmental/Water Engineering are closely related or are important factors in biomedical aspects.

Heatmaps for Breast Cancer Diagnosis dataset

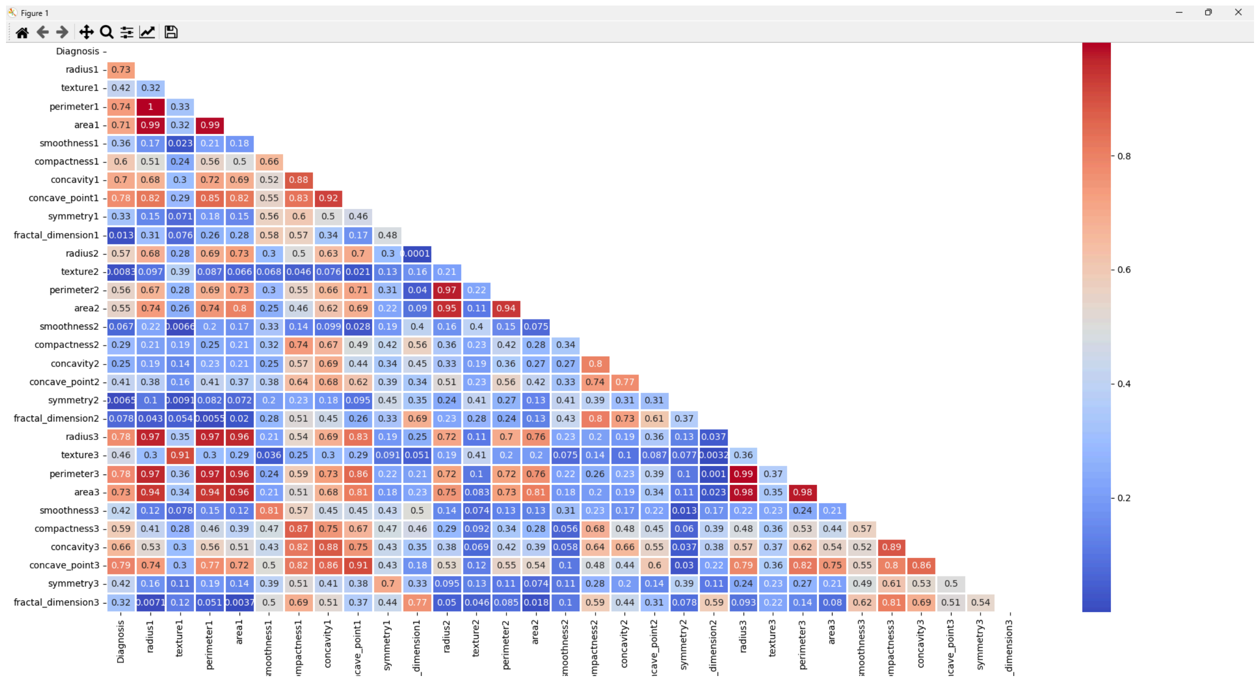


Figure 1. Overall Heatmap for Correlation to Diagnosis

Overall, the variables with a high (or significant enough to be considered) correlation value to diagnosis are: radius, perimeter, area, compactness, concavity, and concave point of the samples. The following heatmaps are added to provide better visualisation of the correlation between Diagnosis and other variables. The dataset has a base of the following 10 variables: radius, texture, perimeter, area, smoothness, compactness, concavity, concave point, symmetry, and fractual dimension - the division of the dataset is based on this list.

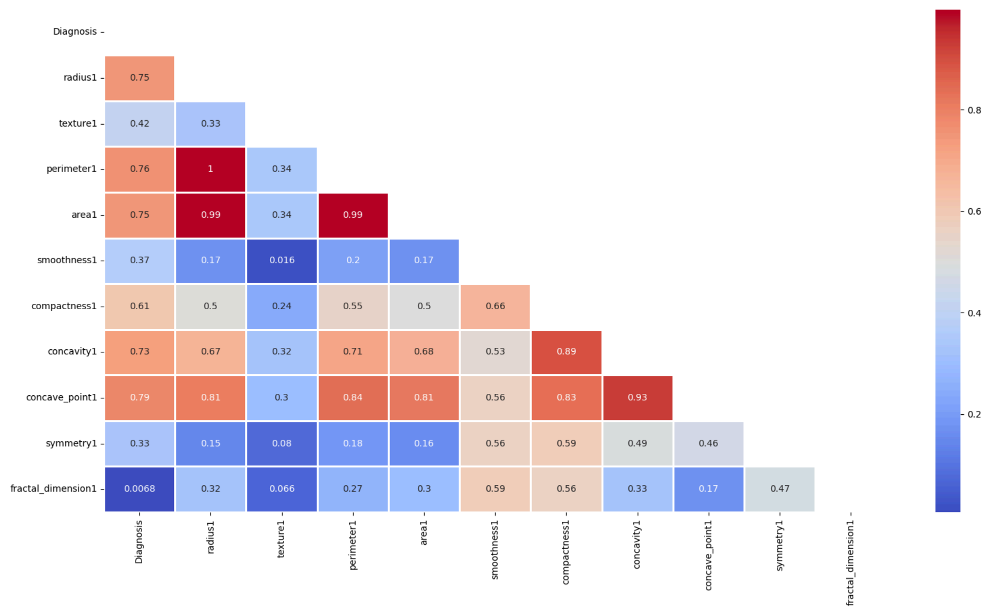


Figure 2. Heatmap for Correlation to Diagnosis - first subset of data (variables with 1 in it)

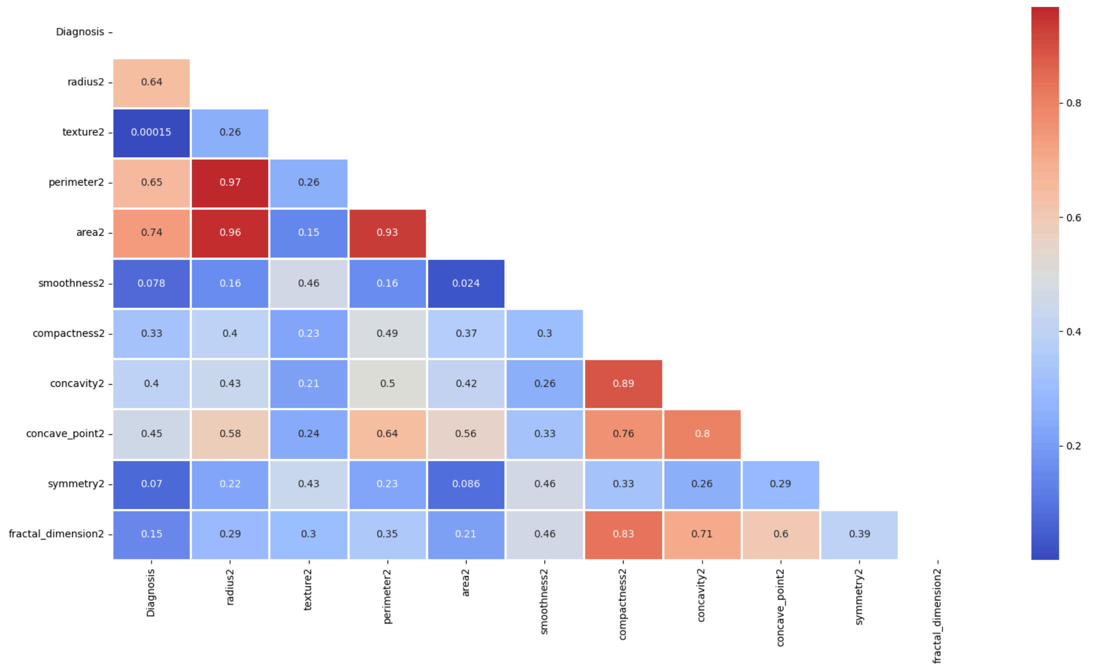


Figure 3. Heatmap for Correlation to Diagnosis - second subset of data (variables with 2 in it)

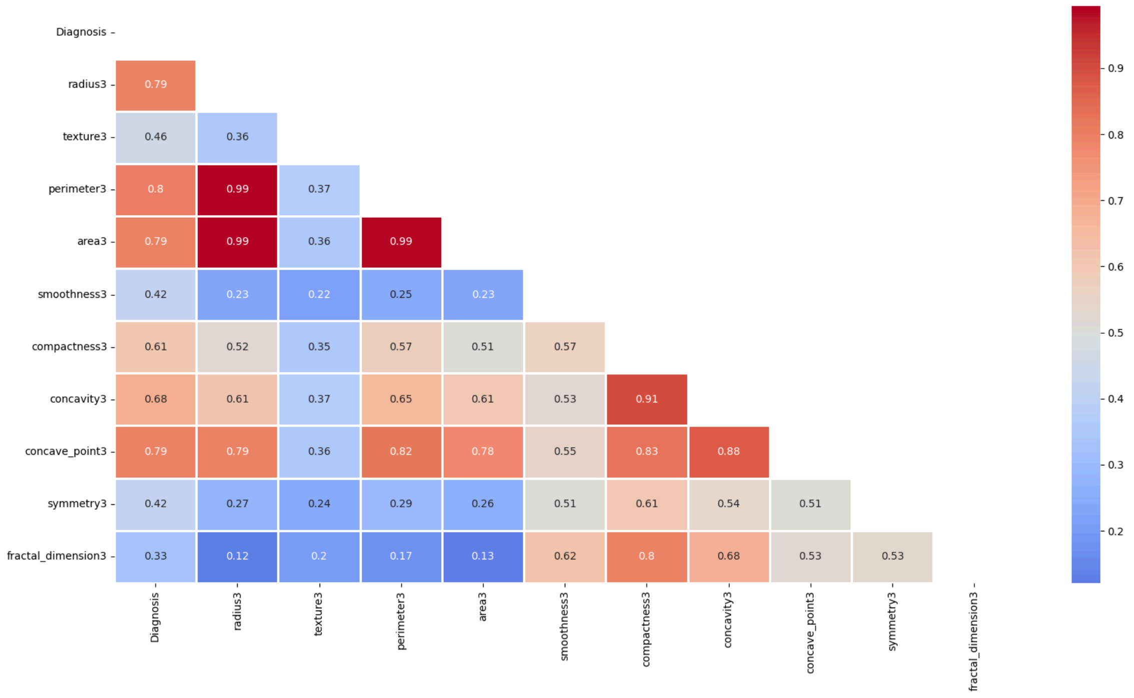


Figure 4. Heatmap for Correlation to Diagnosis - third subset of data (variables with 3 in it)

EDA (Exploratory Data Analysis) Summary

- Except for “Radius”, “Perimeter”, “Area”, “Compactness”, “Concavity”, and “Concave Point”, other variables have weak relationships with breast cancer “Diagnosis” feature and would not significantly contributed to making statistical decision (of correlation).
- Data set 1 and 3 have 6 variables that demonstrate very strong relationship with breast cancer “Diagnosis” feature, which are: Radius, Perimeter, Area, Compactness, Concavity, and Concave Point with the value range from 0.59 to 0.79
- Add all variables of the same type together (e.g. radius_sum = radius1 + radius2 + radius3, etc.) to see the overall correlation of that singular factor to the target variable of “Diagnosis”
- Repeated appearances of Radius, Perimeter, and Area strong correlations to Diagnosis are suggested to be anticipated since variables of Perimeter and Area depend on the Radius values significantly. If the Radius variables are strongly correlated to the Diagnosis results, Perimeter and Area are understandably inherited the same behaviour.

Class Labelling for Targeted Variable

1) Distribution of target variable of Diagnosis (0: Benign & 1: Malignent)

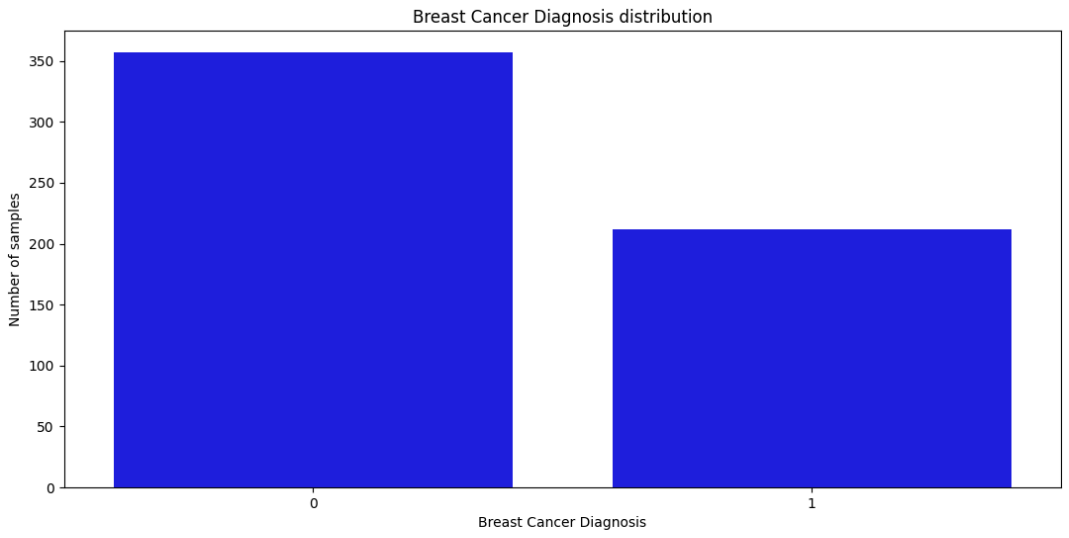


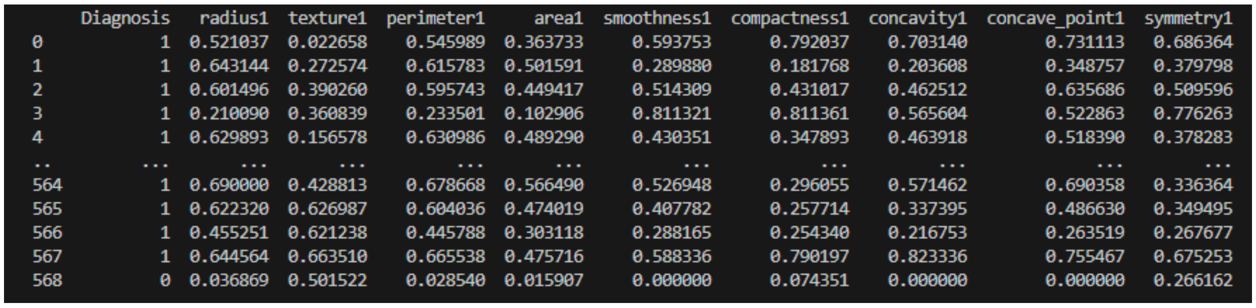
Figure 5. Distribution of Breast Cancer Diagnosis

- **Target variable:** Diagnosis
- **Predictors:** radius, texture, perimeter, area, smoothness, compactness, concavity, concave_point, symmetry, and fractual_dimension of all 3 sub-dataset

2) Distribution comparison

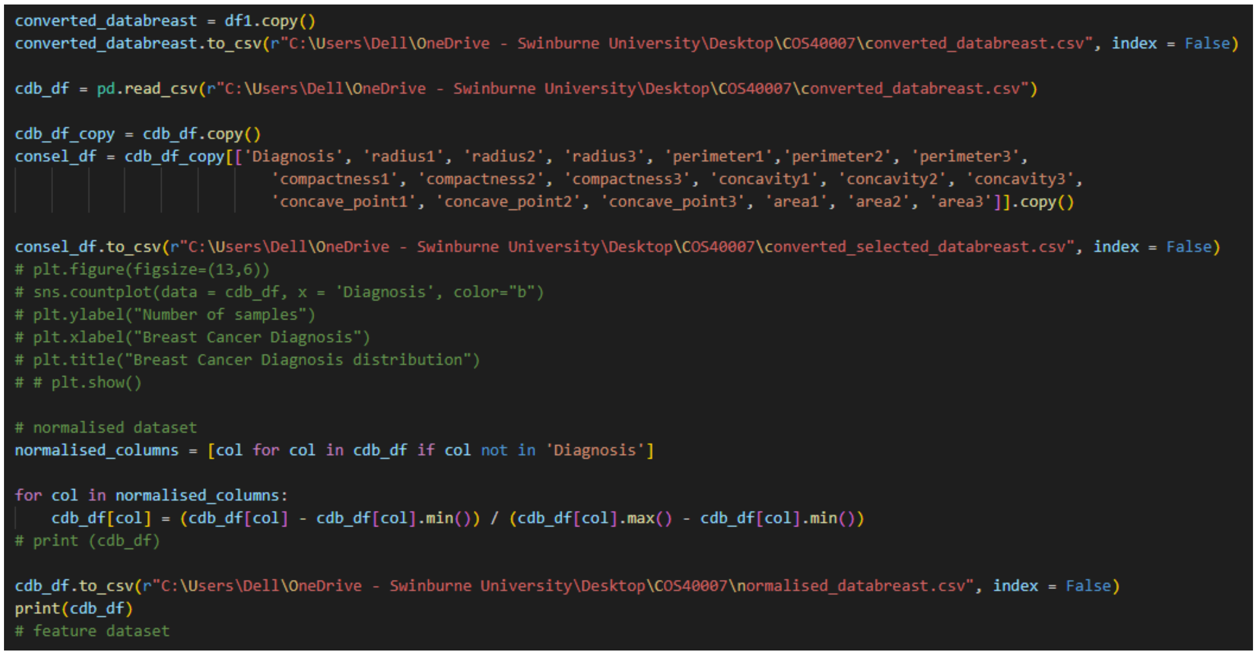
The difference between the diagnosis is roughly 150 samples which is under the significant difference mark of 200 samples. It is not nearly balanced but it is in the acceptable range of differences.

3) Normalisation of predictors



```
     Diagnosis   radius1  texture1  perimeter1     area1  smoothness1  compactness1  concavity1  concave_point1  symmetry1
0            1  0.521037  0.022658    0.545989  0.363733     0.593753      0.792037    0.703140        0.731113   0.686364
1            1  0.643144  0.272574    0.615783  0.501591     0.289880      0.181768    0.203608        0.348757   0.379798
2            1  0.601496  0.390260    0.595743  0.449417     0.514309      0.431017    0.462512        0.635686   0.509596
3            1  0.210090  0.360839    0.233501  0.102906     0.811321      0.811361    0.565604        0.522863   0.776263
4            1  0.629893  0.156578    0.630986  0.489290     0.430351      0.347893    0.463918        0.518390   0.378283
..         ...       ...       ...         ...       ...          ...           ...         ...             ...        ...
564          1  0.690000  0.428813    0.678668  0.566490     0.526948      0.296055    0.571462        0.690358   0.336364
565          1  0.622320  0.626987    0.604036  0.474019     0.407782      0.257714    0.337395        0.486630   0.349495
566          1  0.455251  0.621238    0.445788  0.303118     0.288165      0.254340    0.216753        0.263519   0.267677
567          1  0.644564  0.663510    0.665538  0.475716     0.588336      0.790197    0.823336        0.755467   0.675253
568          0  0.036869  0.501522    0.028540  0.015907     0.000000      0.074351    0.000000        0.000000   0.266162
```

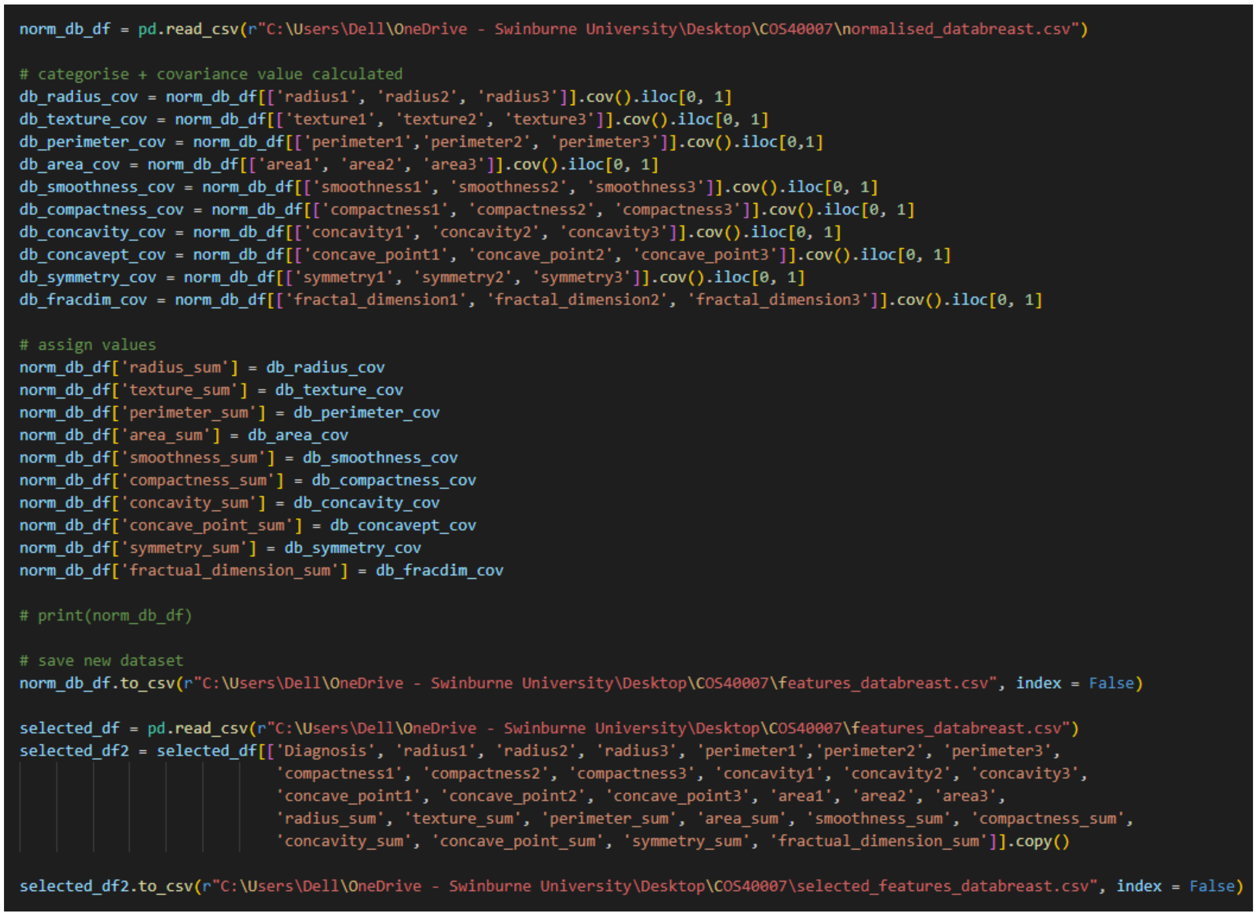
Figure 6. Some normalised predictors' values for Breast Cancer diagnosis



```
converted_databreast = df1.copy()
converted_databreast.to_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\converted_databreast.csv", index = False)

cdb_df = pd.read_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\converted_databreast.csv")

cdb_df_copy = cdb_df.copy()
consel_df = cdb_df_copy[['Diagnosis', 'radius1', 'radius2', 'radius3', 'perimeter1','perimeter2', 'perimeter3',
                         'compactness1', 'compactness2', 'compactness3', 'concavity1', 'concavity2', 'concavity3',
                         'concave_point1', 'concave_point2', 'concave_point3', 'area1', 'area2', 'area3']].copy()

consel_df.to_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\converted_selected_databreast.csv", index = False)
# plt.figure(figsize=(13,6))
# sns.countplot(data = cdb_df, x = 'Diagnosis', color="b")
# plt.ylabel("Number of samples")
# plt.xlabel("Breast Cancer Diagnosis")
# plt.title("Breast Cancer Diagnosis distribution")
# # plt.show()

# normalised dataset
normalised_columns = [col for col in cdb_df if col not in 'Diagnosis']

for col in normalised_columns:
    cdb_df[col] = (cdb_df[col] - cdb_df[col].min()) / (cdb_df[col].max() - cdb_df[col].min())
# print (cdb_df)

cdb_df.to_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\normalised_databreast.csv", index = False)
print(cdb_df)
# feature dataset
```

Figure 7. Normalisation source code

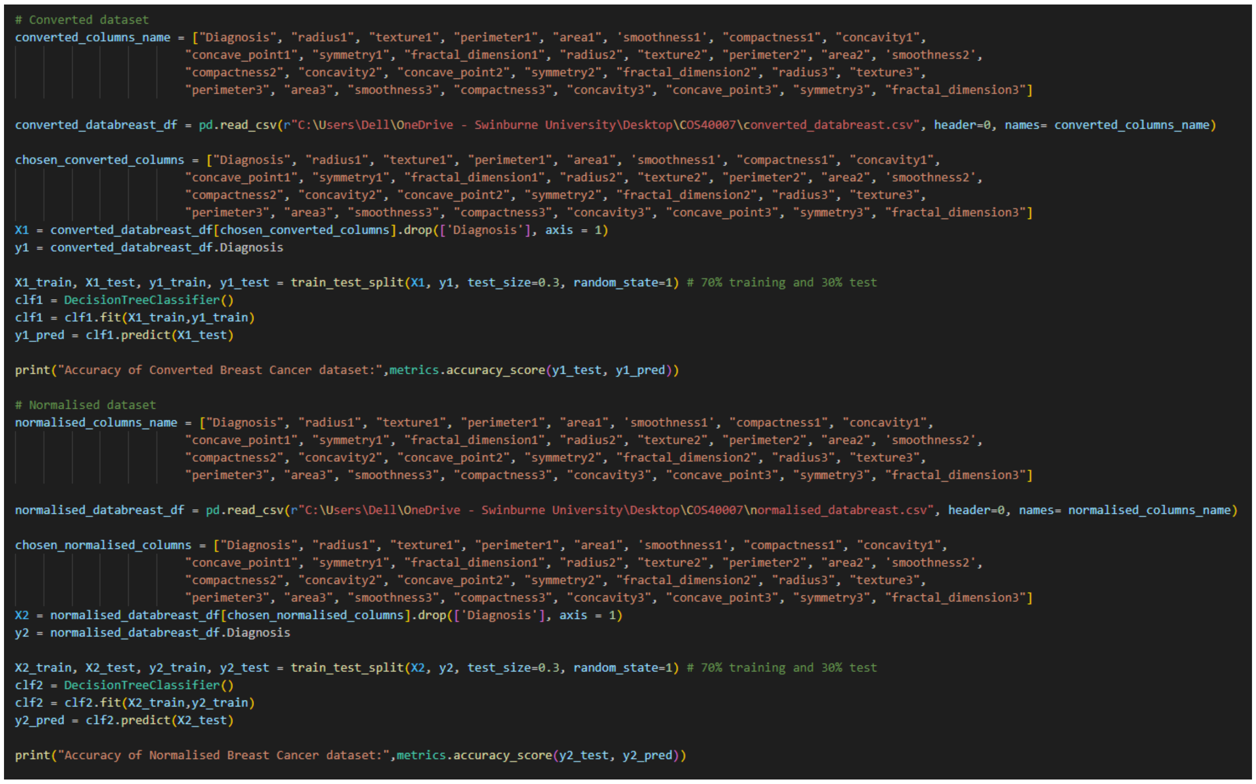
4) Feature Engineering and Feature Selections



```
norm_db_df = pd.read_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\normalised_databreast.csv")

# categorise + covariance value calculated
db_radius_cov = norm_db_df[['radius1', 'radius2', 'radius3']].cov().iloc[0, 1]
db_texture_cov = norm_db_df[['texture1', 'texture2', 'texture3']].cov().iloc[0, 1]
db_perimeter_cov = norm_db_df[['perimeter1','perimeter2', 'perimeter3']].cov().iloc[0,1]
db_area_cov = norm_db_df[['area1', 'area2', 'area3']].cov().iloc[0, 1]
db_smoothness_cov = norm_db_df[['smoothness1', 'smoothness2', 'smoothness3']].cov().iloc[0, 1]
db_compactness_cov = norm_db_df[['compactness1', 'compactness2', 'compactness3']].cov().iloc[0, 1]
db_concavity_cov = norm_db_df[['concavity1', 'concavity2', 'concavity3']].cov().iloc[0, 1]
db_concavept_cov = norm_db_df[['concave_point1', 'concave_point2', 'concave_point3']].cov().iloc[0, 1]
db_symmetry_cov = norm_db_df[['symmetry1', 'symmetry2', 'symmetry3']].cov().iloc[0, 1]
db_fracdim_cov = norm_db_df[['fractal_dimension1', 'fractal_dimension2', 'fractal_dimension3']].cov().iloc[0, 1]

# assign values
norm_db_df['radius_sum'] = db_radius_cov
norm_db_df['texture_sum'] = db_texture_cov
norm_db_df['perimeter_sum'] = db_perimeter_cov
norm_db_df['area_sum'] = db_area_cov
norm_db_df['smoothness_sum'] = db_smoothness_cov
norm_db_df['compactness_sum'] = db_compactness_cov
norm_db_df['concavity_sum'] = db_concavity_cov
norm_db_df['concave_point_sum'] = db_concavept_cov
norm_db_df['symmetry_sum'] = db_symmetry_cov
norm_db_df['fractual_dimension_sum'] = db_fracdim_cov

# print(norm_db_df)

# save new dataset
norm_db_df.to_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\features_databreast.csv", index = False)

selected_df = pd.read_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\features_databreast.csv")
selected_df2 = selected_df[['Diagnosis', 'radius1', 'radius2', 'radius3', 'perimeter1','perimeter2', 'perimeter3',
                            'compactness1', 'compactness2', 'compactness3', 'concavity1', 'concavity2', 'concavity3',
                            'concave_point1', 'concave_point2', 'concave_point3', 'area1', 'area2', 'area3',
                            'radius_sum', 'texture_sum', 'perimeter_sum', 'area_sum', 'smoothness_sum', 'compactness_sum',
                            'concavity_sum', 'concave_point_sum', 'symmetry_sum', 'fractual_dimension_sum']].copy()

selected_df2.to_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\selected_features_databreast.csv", index = False)
```

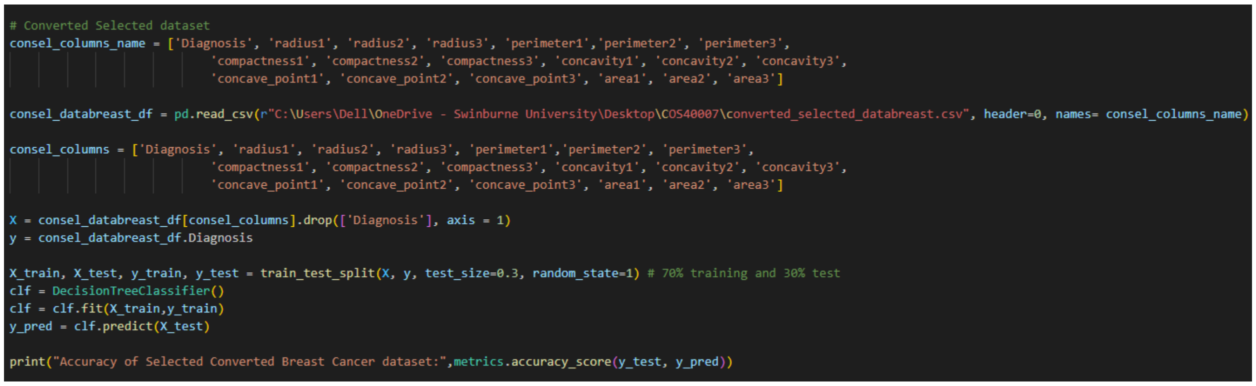
Figure 8. Feature Additions source code

5) Training and decision tree model development



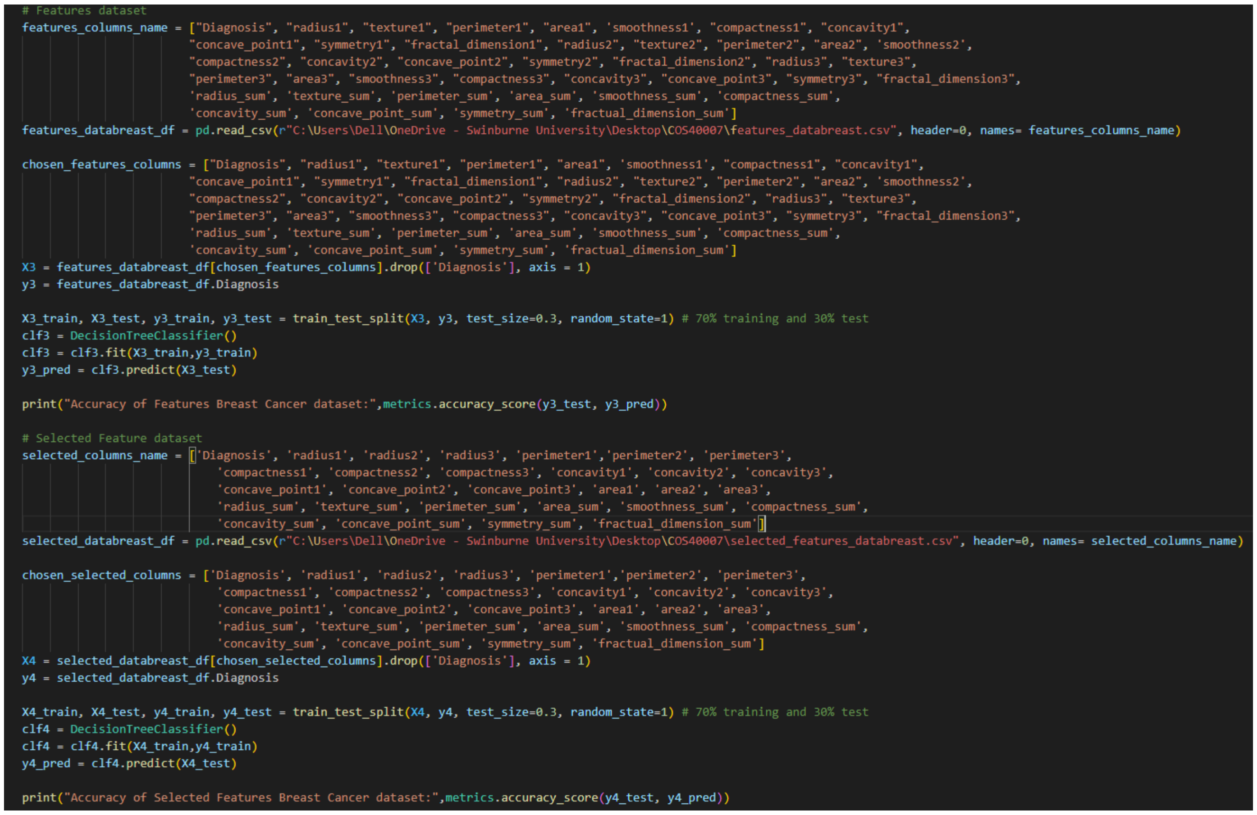
```python
# Converted dataset
converted_columns_name = ["Diagnosis", "radius1", "texture1", "perimeter1", "area1", 'smoothness1', "compactness1", "concavity1",
                          "concave_point1", "symmetry1", "fractal_dimension1", "radius2", "texture2", "perimeter2", "area2", 'smoothness2',
                          "compactness2", "concavity2", "concave_point2", "symmetry2", "fractal_dimension2", "radius3", "texture3",
                          "perimeter3", "area3", "smoothness3", "compactness3", "concavity3", "concave_point3", "symmetry3", "fractal_dimension3"]

converted_databreast_df = pd.read_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\converted_databreast.csv", header=0, names= converted_columns_name)

chosen_converted_columns = ["Diagnosis", "radius1", "texture1", "perimeter1", "area1", 'smoothness1', "compactness1", "concavity1",
                          "concave_point1", "symmetry1", "fractal_dimension1", "radius2", "texture2", "perimeter2", "area2", 'smoothness2',
                          "compactness2", "concavity2", "concave_point2", "symmetry2", "fractal_dimension2", "radius3", "texture3",
                          "perimeter3", "area3", "smoothness3", "compactness3", "concavity3", "concave_point3", "symmetry3", "fractal_dimension3"]
X1 = converted_databreast_df[chosen_converted_columns].drop(['Diagnosis'], axis = 1)
y1 = converted_databreast_df.Diagnosis

X1_train, X1_test, y1_train, y1_test = train_test_split(X1, y1, test_size=0.3, random_state=1) # 70% training and 30% test
clf1 = DecisionTreeClassifier()
clf1 = clf1.fit(X1_train,y1_train)
y1_pred = clf1.predict(X1_test)

print("Accuracy of Converted Breast Cancer dataset:",metrics.accuracy_score(y1_test, y1_pred))

# Normalised dataset
normalised_columns_name = ["Diagnosis", "radius1", "texture1", "perimeter1", "area1", 'smoothness1', "compactness1", "concavity1",
                          "concave_point1", "symmetry1", "fractal_dimension1", "radius2", "texture2", "perimeter2", "area2", 'smoothness2',
                          "compactness2", "concavity2", "concave_point2", "symmetry2", "fractal_dimension2", "radius3", "texture3",
                          "perimeter3", "area3", "smoothness3", "compactness3", "concavity3", "concave_point3", "symmetry3", "fractal_dimension3"]

normalised_databreast_df = pd.read_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\normalised_databreast.csv", header=0, names= normalised_columns_name)

chosen_normalised_columns = ["Diagnosis", "radius1", "texture1", "perimeter1", "area1", 'smoothness1', "compactness1", "concavity1",
                          "concave_point1", "symmetry1", "fractal_dimension1", "radius2", "texture2", "perimeter2", "area2", 'smoothness2',
                          "compactness2", "concavity2", "concave_point2", "symmetry2", "fractal_dimension2", "radius3", "texture3",
                          "perimeter3", "area3", "smoothness3", "compactness3", "concavity3", "concave_point3", "symmetry3", "fractal_dimension3"]
X2 = normalised_databreast_df[chosen_normalised_columns].drop(['Diagnosis'], axis = 1)
y2 = normalised_databreast_df.Diagnosis

X2_train, X2_test, y2_train, y2_test = train_test_split(X2, y2, test_size=0.3, random_state=1) # 70% training and 30% test
clf2 = DecisionTreeClassifier()
clf2 = clf2.fit(X2_train,y2_train)
y2_pred = clf2.predict(X2_test)

print("Accuracy of Normalised Breast Cancer dataset:",metrics.accuracy_score(y2_test, y2_pred))
```

Figure 9. Converted and Normalised dataset training and decision tree model development source code



```python
# Converted Selected dataset
consel_columns_name = ['Diagnosis', 'radius1', 'radius2', 'radius3', 'perimeter1','perimeter2', 'perimeter3',
                       'compactness1', 'compactness2', 'compactness3', 'concavity1', 'concavity2', 'concavity3',
                       'concave_point1', 'concave_point2', 'concave_point3', 'area1', 'area2', 'area3']

consel_databreast_df = pd.read_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\converted_selected_databreast.csv", header=0, names= consel_columns_name)

consel_columns = ['Diagnosis', 'radius1', 'radius2', 'radius3', 'perimeter1','perimeter2', 'perimeter3',
                       'compactness1', 'compactness2', 'compactness3', 'concavity1', 'concavity2', 'concavity3',
                       'concave_point1', 'concave_point2', 'concave_point3', 'area1', 'area2', 'area3']

X = consel_databreast_df[consel_columns].drop(['Diagnosis'], axis = 1)
y = consel_databreast_df.Diagnosis

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1) # 70% training and 30% test
clf = DecisionTreeClassifier()
clf = clf.fit(X_train,y_train)
y_pred = clf.predict(X_test)

print("Accuracy of Selected Converted Breast Cancer dataset:",metrics.accuracy_score(y_test, y_pred))
```

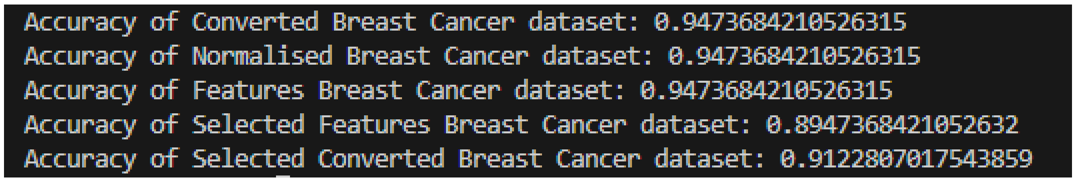
Figure 10. Converted Selected dataset training and decision tree model development source code



```python
# Features dataset
features_columns_name = ["Diagnosis", "radius1", "texture1", "perimeter1", "area1", 'smoothness1', "compactness1", "concavity1",
                        "concave_point1", "symmetry1", "fractal_dimension1", "radius2", "texture2", "perimeter2", "area2", 'smoothness2',
                        "compactness2", "concavity2", "concave_point2", "symmetry2", "fractal_dimension2", "radius3", "texture3",
                        "perimeter3", "area3", "smoothness3", "compactness3", "concavity3", "concave_point3", "symmetry3", "fractal_dimension3",
                        'radius_sum', 'texture_sum', 'perimeter_sum', 'area_sum', 'smoothness_sum', 'compactness_sum',
                        'concavity_sum', 'concave_point_sum', 'symmetry_sum', 'fractual_dimension_sum']
features_databreast_df = pd.read_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\features_databreast.csv", header=0, names= features_columns_name)

chosen_features_columns = ["Diagnosis", "radius1", "texture1", "perimeter1", "area1", 'smoothness1', "compactness1", "concavity1",
                        "concave_point1", "symmetry1", "fractal_dimension1", "radius2", "texture2", "perimeter2", "area2", 'smoothness2',
                        "compactness2", "concavity2", "concave_point2", "symmetry2", "fractal_dimension2", "radius3", "texture3",
                        "perimeter3", "area3", "smoothness3", "compactness3", "concavity3", "concave_point3", "symmetry3", "fractal_dimension3",
                        'radius_sum', 'texture_sum', 'perimeter_sum', 'area_sum', 'smoothness_sum', 'compactness_sum',
                        'concavity_sum', 'concave_point_sum', 'symmetry_sum', 'fractual_dimension_sum']
X3 = features_databreast_df[chosen_features_columns].drop(['Diagnosis'], axis = 1)
y3 = features_databreast_df.Diagnosis

X3_train, X3_test, y3_train, y3_test = train_test_split(X3, y3, test_size=0.3, random_state=1) # 70% training and 30% test
clf3 = DecisionTreeClassifier()
clf3 = clf3.fit(X3_train,y3_train)
y3_pred = clf3.predict(X3_test)

print("Accuracy of Features Breast Cancer dataset:",metrics.accuracy_score(y3_test, y3_pred))

# Selected Feature dataset
selected_columns_name = ['Diagnosis', 'radius1', 'radius2', 'radius3', 'perimeter1','perimeter2', 'perimeter3',
                            'compactness1', 'compactness2', 'compactness3', 'concavity1', 'concavity2', 'concavity3',
                            'concave_point1', 'concave_point2', 'concave_point3', 'area1', 'area2', 'area3',
                            'radius_sum', 'texture_sum', 'perimeter_sum', 'area_sum', 'smoothness_sum', 'compactness_sum',
                            'concavity_sum', 'concave_point_sum', 'symmetry_sum', 'fractual_dimension_sum']
selected_databreast_df = pd.read_csv(r"C:\Users\Dell\OneDrive - Swinburne University\Desktop\COS40007\selected_features_databreast.csv", header=0, names= selected_columns_name)

chosen_selected_columns = ['Diagnosis', 'radius1', 'radius2', 'radius3', 'perimeter1','perimeter2', 'perimeter3',
                            'compactness1', 'compactness2', 'compactness3', 'concavity1', 'concavity2', 'concavity3',
                            'concave_point1', 'concave_point2', 'concave_point3', 'area1', 'area2', 'area3',
                            'radius_sum', 'texture_sum', 'perimeter_sum', 'area_sum', 'smoothness_sum', 'compactness_sum',
                            'concavity_sum', 'concave_point_sum', 'symmetry_sum', 'fractual_dimension_sum']
X4 = selected_databreast_df[chosen_selected_columns].drop(['Diagnosis'], axis = 1)
y4 = selected_databreast_df.Diagnosis

X4_train, X4_test, y4_train, y4_test = train_test_split(X4, y4, test_size=0.3, random_state=1) # 70% training and 30% test
clf4 = DecisionTreeClassifier()
clf4 = clf4.fit(X4_train,y4_train)
y4_pred = clf4.predict(X4_test)

print("Accuracy of Selected Features Breast Cancer dataset:",metrics.accuracy_score(y4_test, y4_pred))
```

Figure 11. Features and Selected Features dataset training and decision tree model development source code

6) Final Comparison Table



```
Accuracy of Converted Breast Cancer dataset: 0.9473684210526315
Accuracy of Normalised Breast Cancer dataset: 0.9473684210526315
Accuracy of Features Breast Cancer dataset: 0.9473684210526315
Accuracy of Selected Features Breast Cancer dataset: 0.8947368421052632
Accuracy of Selected Converted Breast Cancer dataset: 0.9122807017543859
```

Figure 12. FIRST time accuracy after finalising the testing models.

Model 1: converted_databreast.csv (all features no normalisation and no composite features)
Model 2: normalised_databreast.csv (all features with normalisation, no composite features)
Model 3: features_databreast.csv (all features with normalisation + composite features)
Model 4: selected_feature_databreast.csv (selected features with normalisation)
Model 5: selected_converted_concrete.csv (selected feature, no normalisation)

Model 1	Model 2	Model 3	Model 4	Model 5
94.74%	94.74%	94.74%	89.47%	91.22%

Table 1. Model Accuracy Comparison table

Summary
The model with the highest accuracy in this case are model 1, 2, and 3 and the model with the lowest accuracy is model 4. The difference is roughly 5-6% and when run the program at least 10 times, the lowest accuracy produced is roughly around 88-89%.

Appendix

- Source code
- Datasets