**BREAST CANCER DATA ANALYSIS OUTLINE**

**Chenyue Yang | Github: anny19951126anny**

**DATASET DESCRIPTION:**

Features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image. n the 3-dimensional space is that described in: [K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34].

**Attribute Information:**

1) ID number 2) Diagnosis (M = malignant, B = benign) 3-32)

**Ten real-valued features are computed for each cell nucleus:**

a) radius (mean of distances from center to points on the perimeter) b) texture (standard deviation of gray-scale values) c) perimeter d) area e) smoothness (local variation in radius lengths) f) compactness (perimeter^2 / area - 1.0) g) concavity (severity of concave portions of the contour) h) concave points (# of concave portions of the contour) i) symmetry j) fractal dimension ("coastline approximation" - 1)

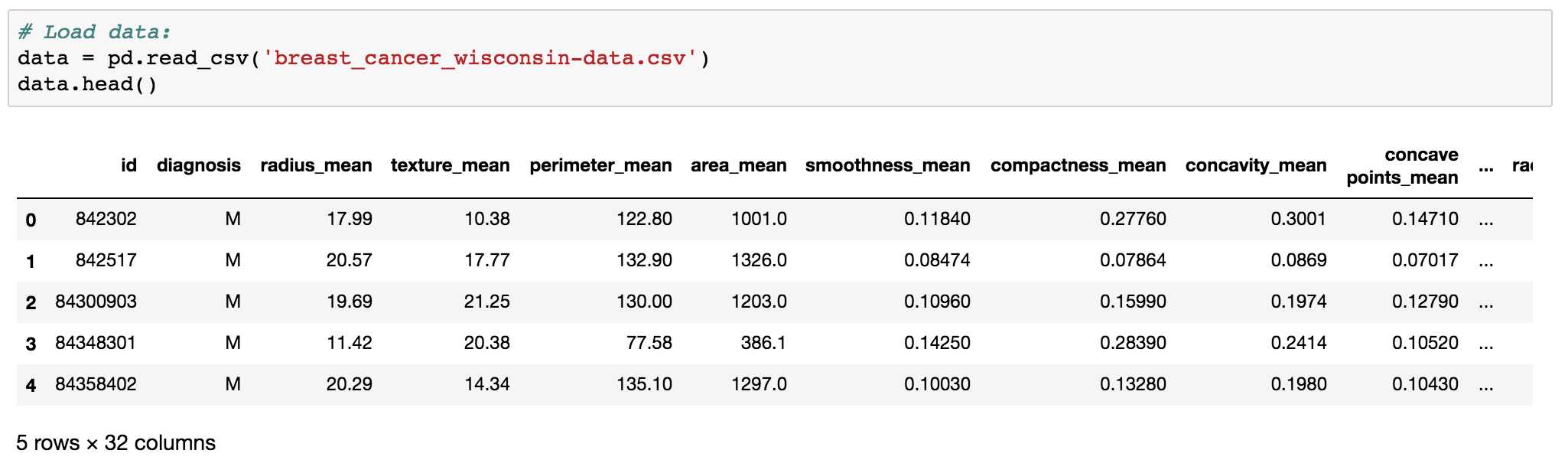
The **mean**, **standard** error and "**worst**" or largest (mean of the three largest values) of these features were computed for each image, resulting in 30 features. For instance, field 3 is Mean Radius, field 13 is Radius SE, field 23 is Worst Radius.

**Class distribution**: 357 benign, 212 malignant

**DATA PREPROCESSING:**

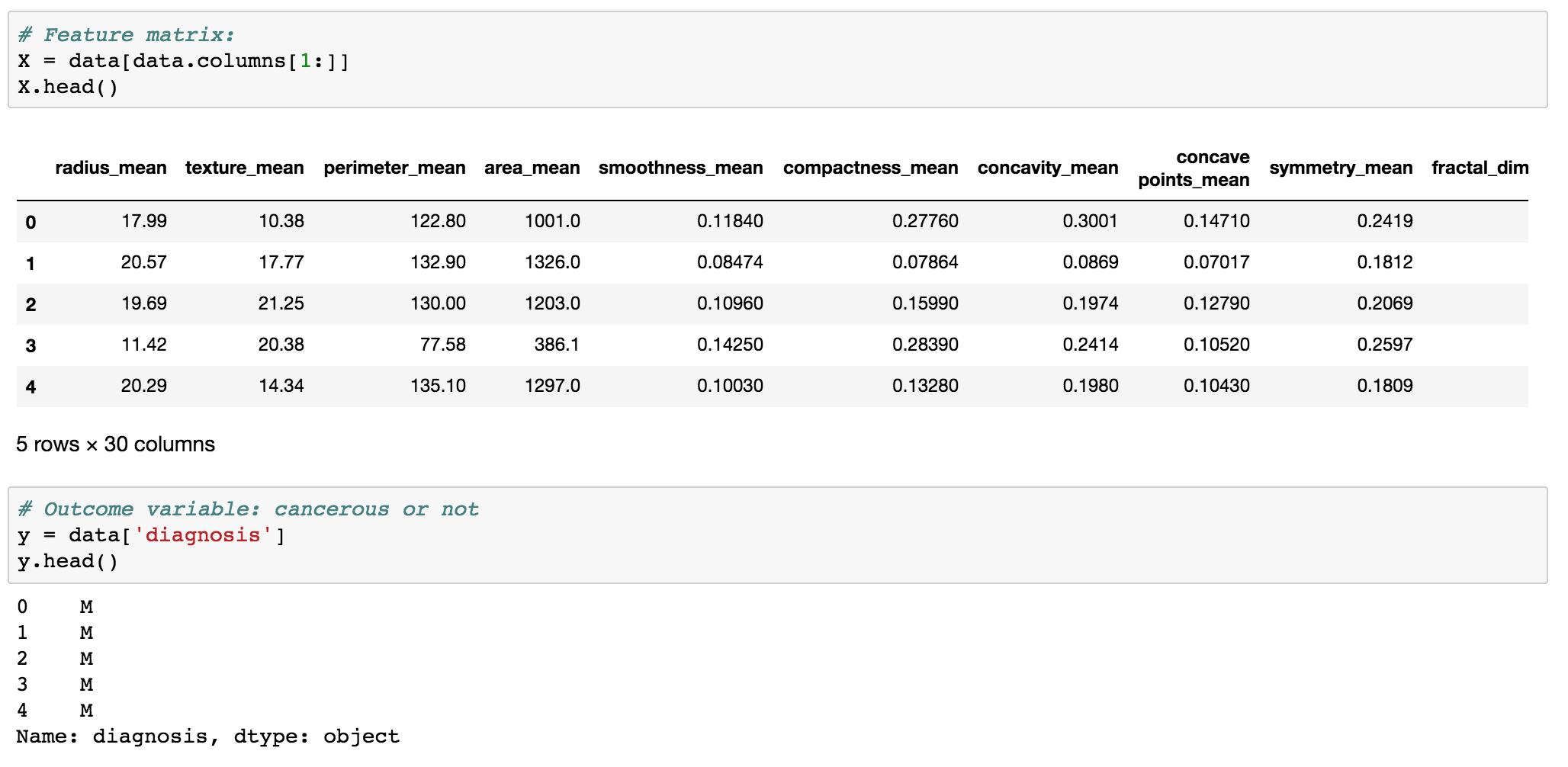
1. **Import dataset, reshape into desirable format, and inspect dataset**:

*Dataset inspection:*



*Dataset descriptive statistics:*

* The ranges of values for each variable are very different, may need standardization later.

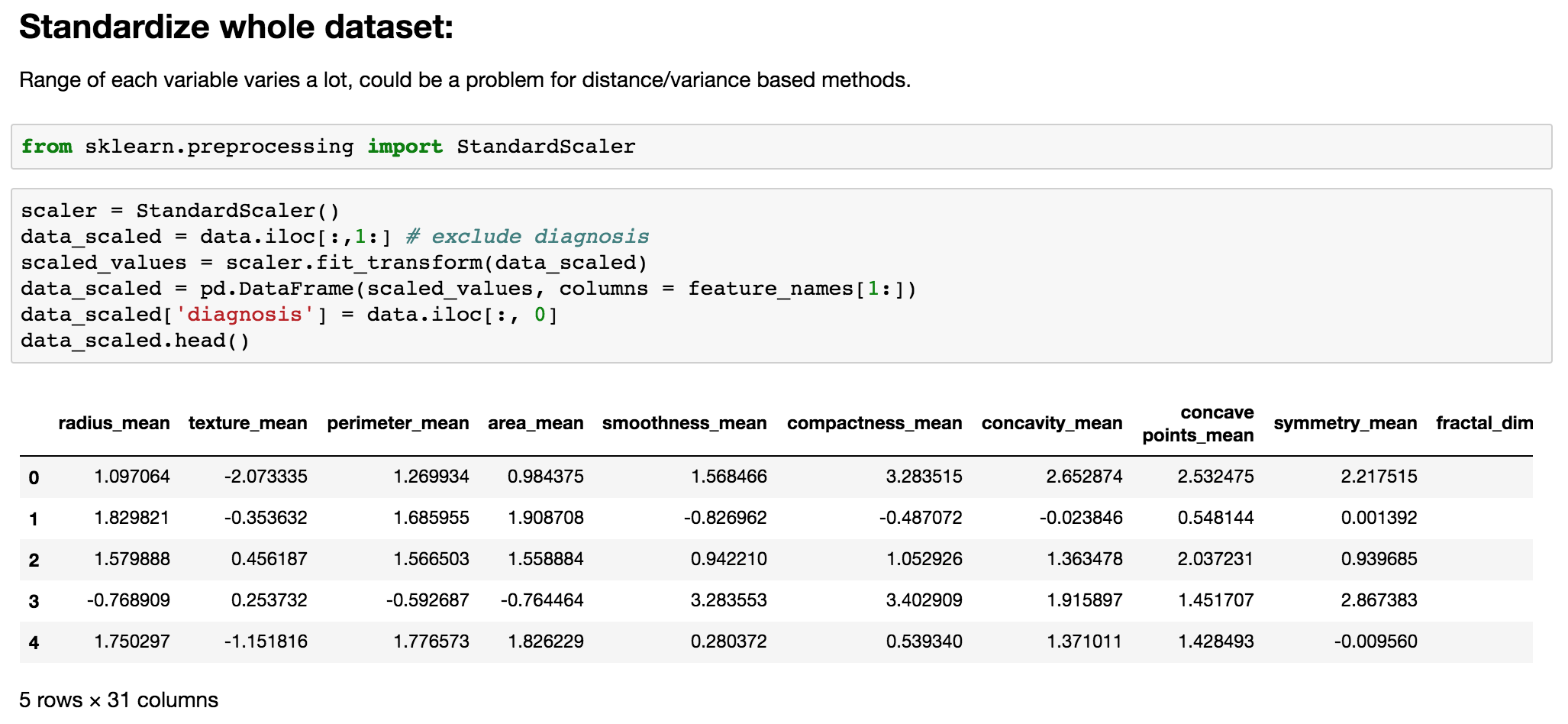
1. **Define feature and outcome variables:**

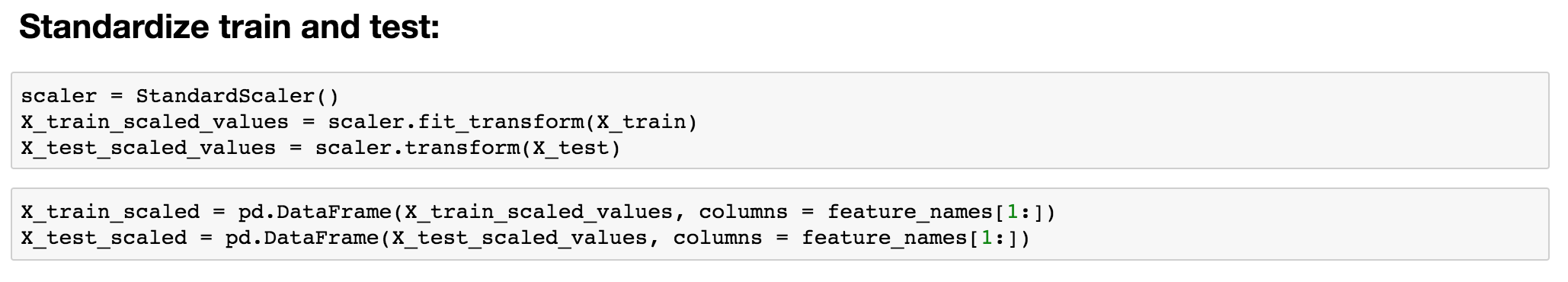
* Total of 30 features (dropped id); diagnosis is used as a binary outcome (Benign vs. Malignant)

1. **Split dataset into train and test:**
2. **Assess (and impute) missing values, and remove any bad columns:**

* *There’s no missing values in dataset, do not need missing imputations*

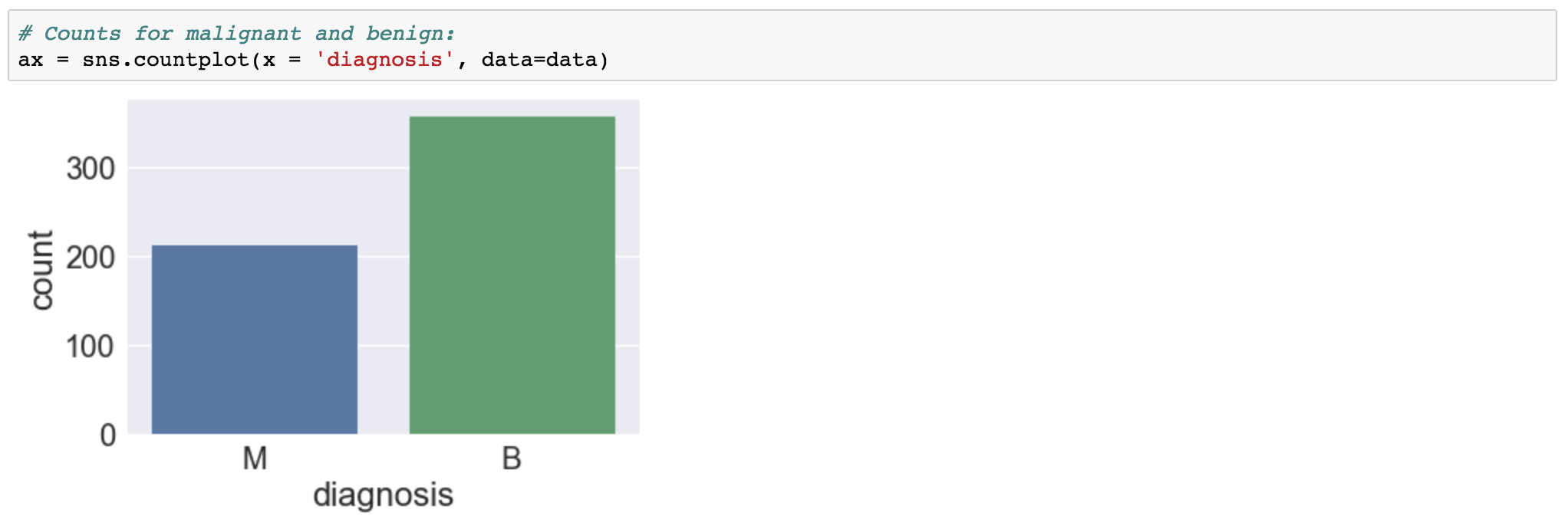
1. **Standardize dataset, train, and test data:**

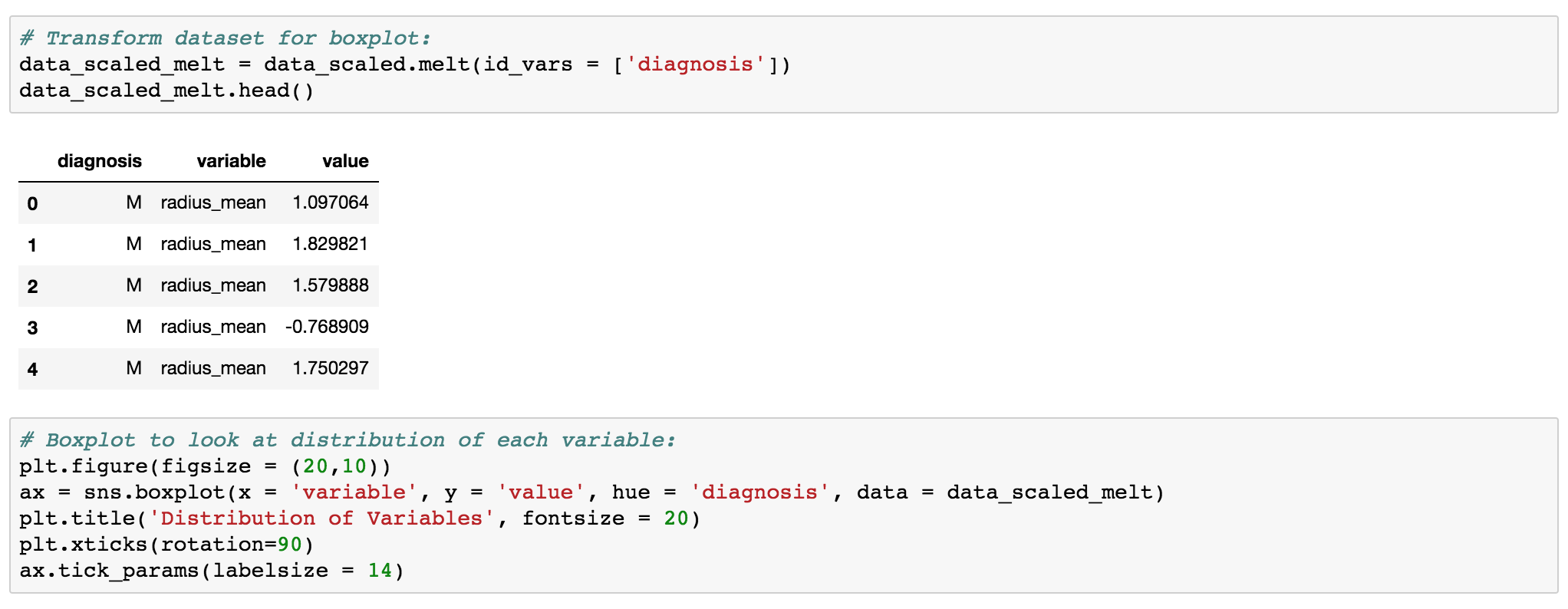


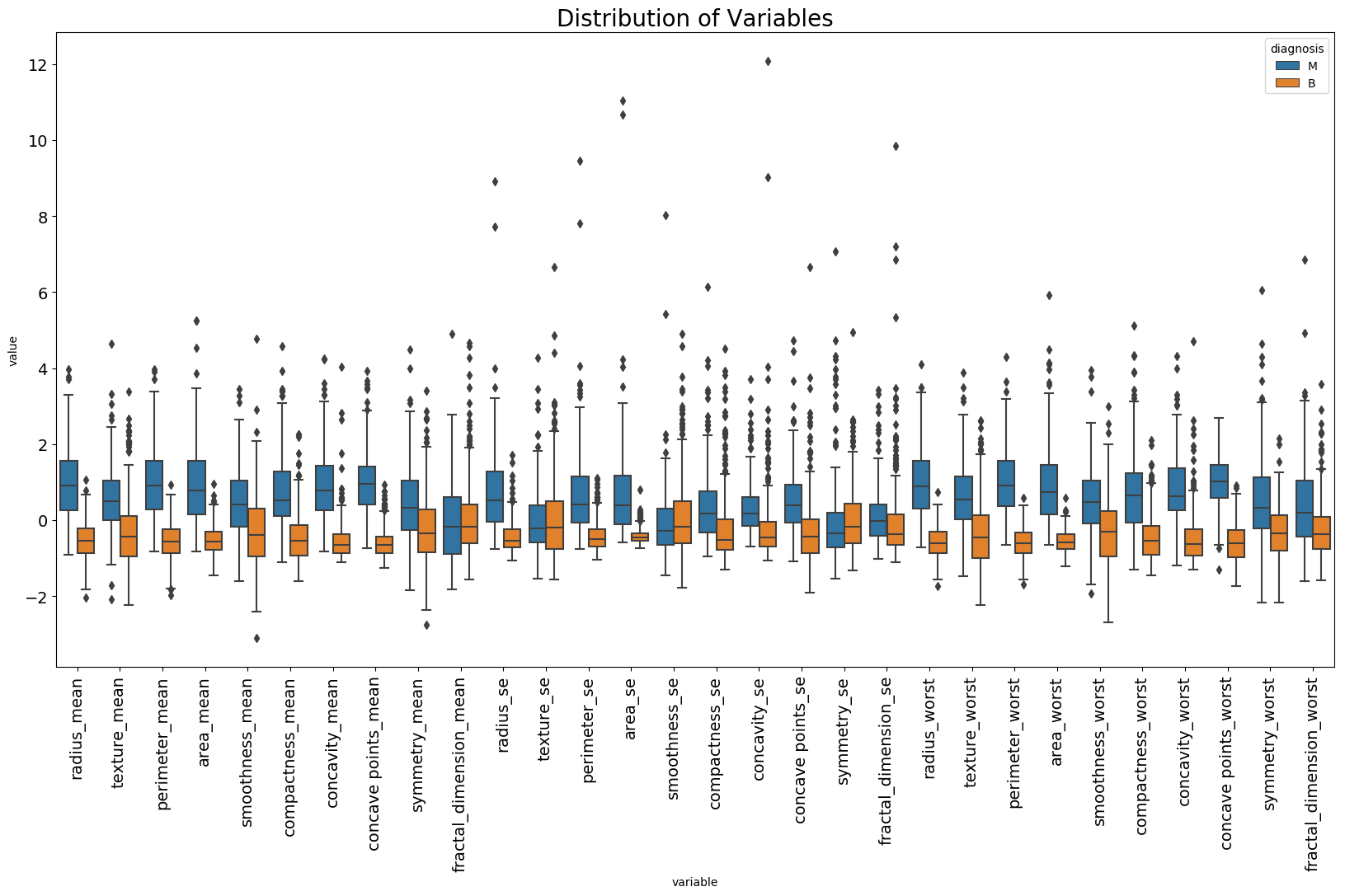


**EXPLORATORY DATA ANALYSIS & FEATURE SELECTION:**

1. **Visualize data:**

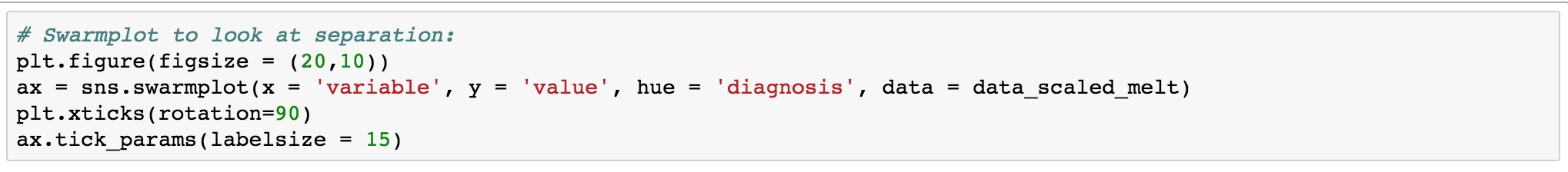
*Count plot is used to inspect distribution of benign vs. malignant:*

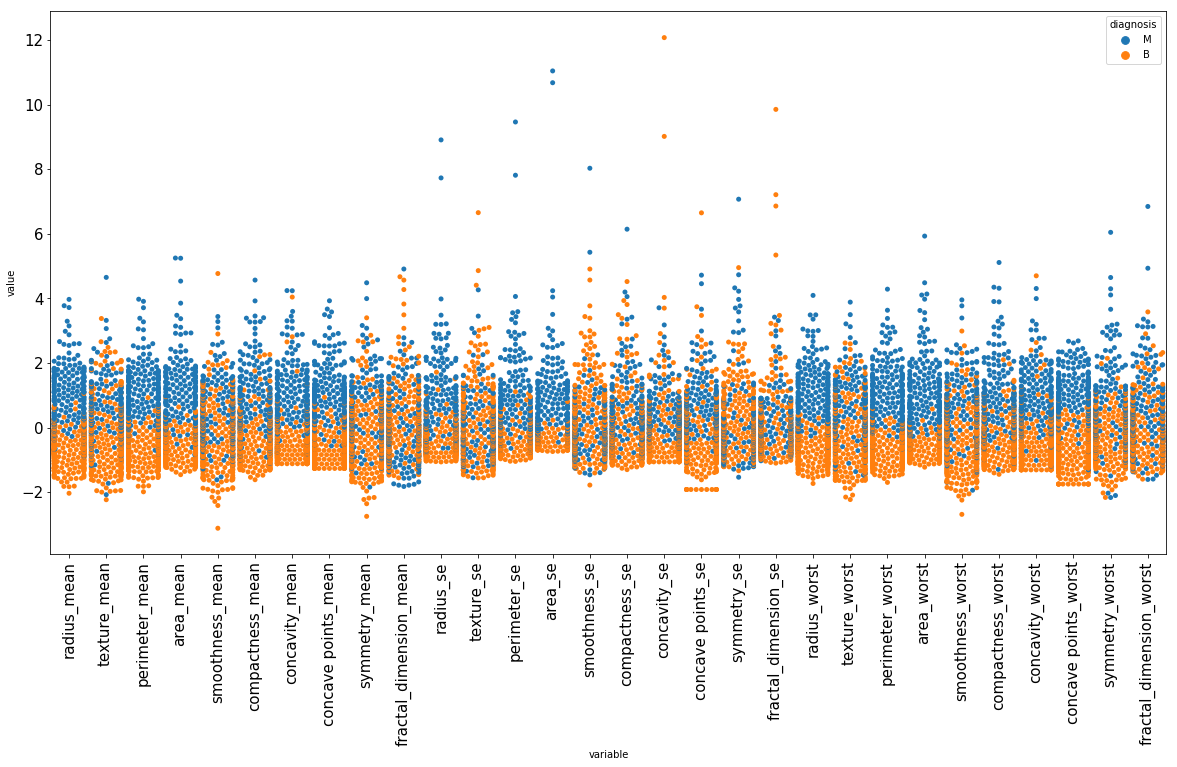
*Boxplot showing distribution of each predictor variable:*



* Lots of outliers for each variable.
* By inspection, mean value of variables for benign samples are lower than for malignant samples.

*Swarmplot is used to get a better idea of how much each variable is separated by benign vs. malignant. Variables that seem to be more associated with the separation could be included in classification model:*



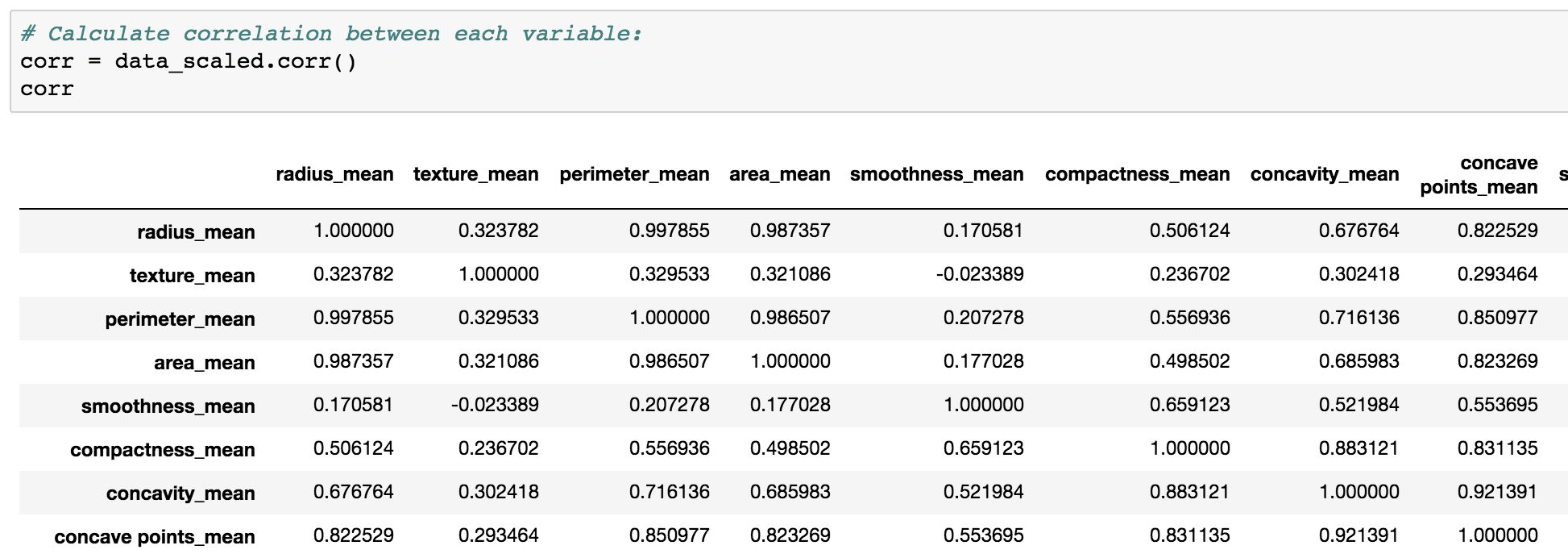


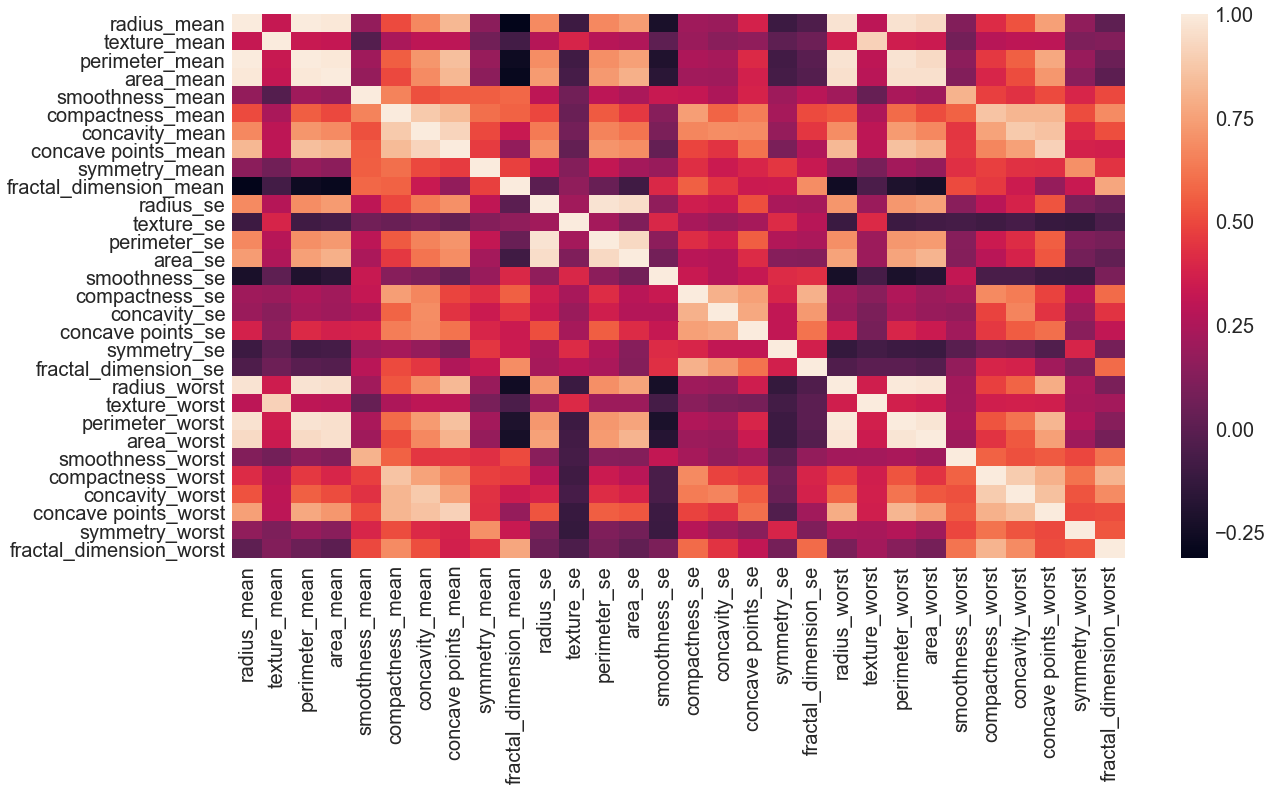
* Variables where B vs. M are relatively well-separated visually:

*{radius\_mean, perimeter\_mean, area\_mean, compactness\_mean, concavity\_mean, concave points\_mean, radius\_se, perimeter\_se, area\_se, radius\_worst, perimeter\_worst, area\_worst, concave points\_worst}*

* Could consider including these variables as features for classification model

1. **Assess and visualize multicollinearity between each variable through correlation matrix and heat map:**

*Correlation matrix:*

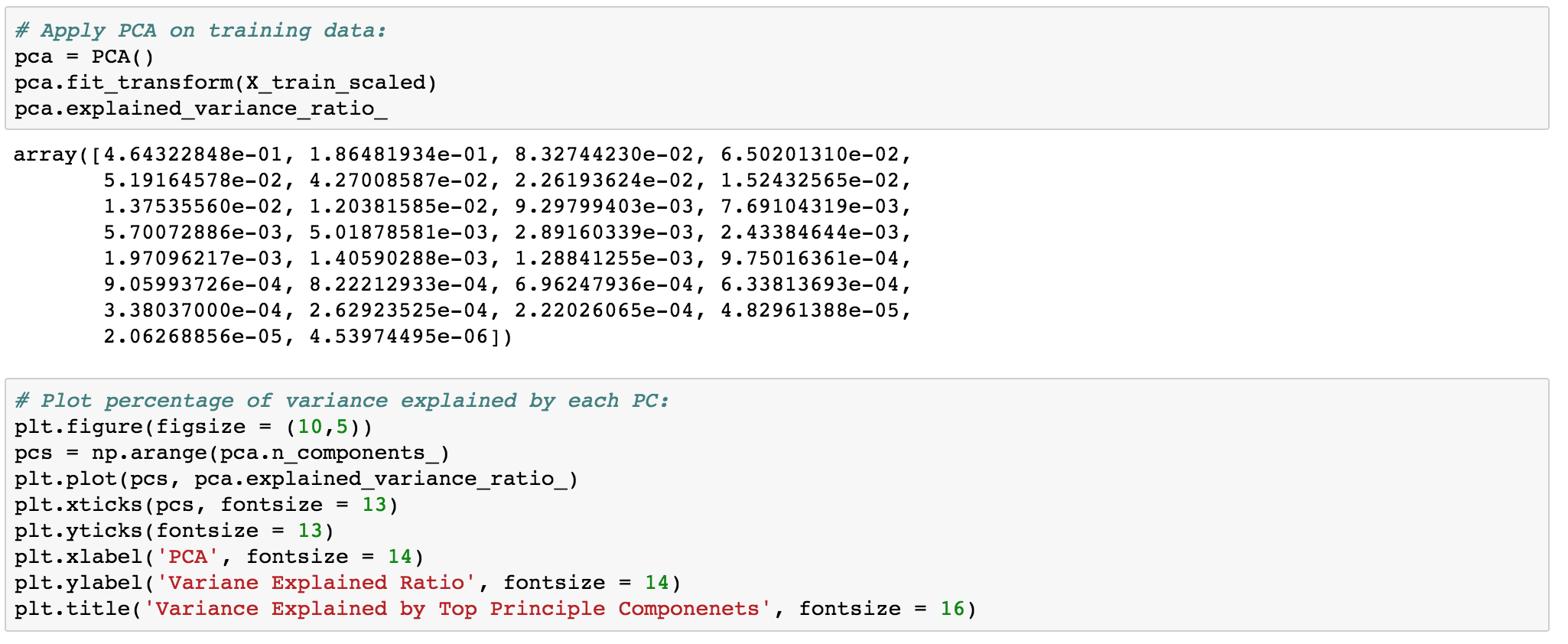


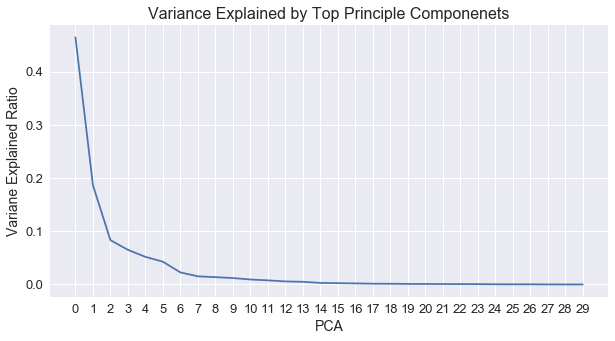
Highly correlated features groups:

* *{radius\_mean, perimeter\_mean, area\_mean, radius\_worst, perimeter\_worst, area\_worst, concave points\_mean, concave points\_worst} / {texture\_mean, texture\_worst} / {smoothness\_mean, smoothness\_worst} / {compactness\_mean, concavity\_mean, compactness\_worst, concavity\_worst, perimeter\_worst, radius\_worst } / {area\_se, area\_worst}*
* Lots of multicollinearity exist in dataset, could use various feature selection methods and PCA to address such problem.

1. **PCA to reduce dimensionality and multicollinearity and get a benchmark for expected performance of predictive models:**

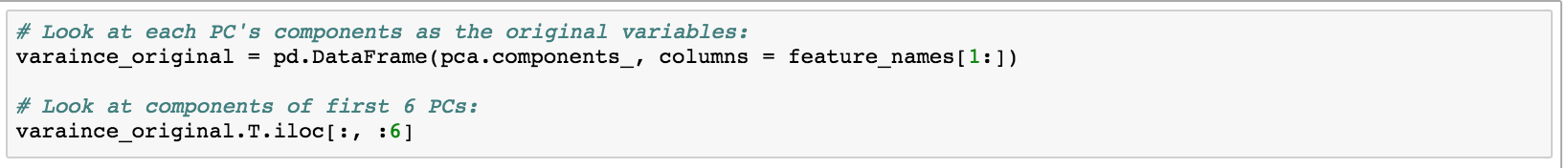
*Use percentage of variance explained to determine optimal number of PCs. This gives an idea of how many features to include in classification models. Look at which variables weight more in each PC that have most explained variance.*

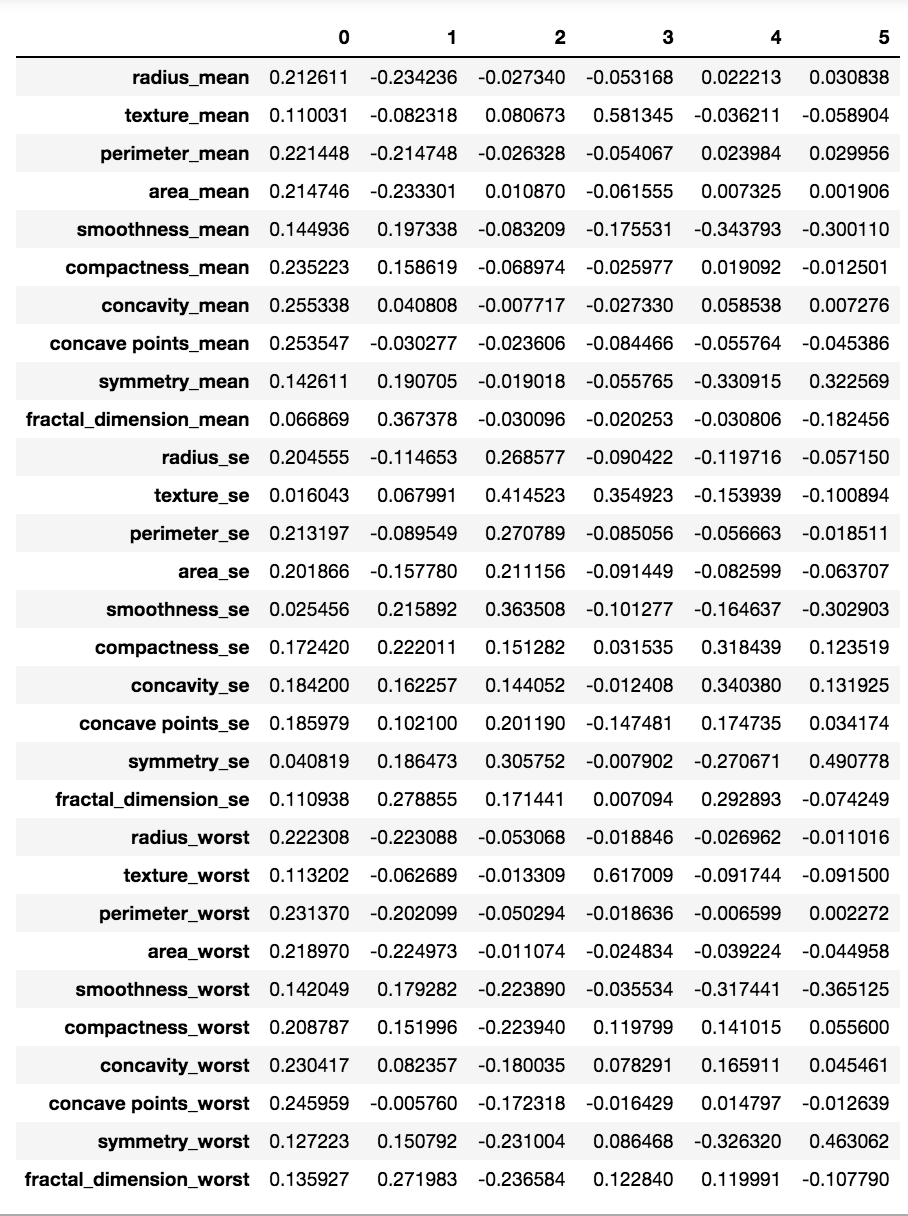


**

*Dimensionally reduce data to 2D to visualize dataset and to determine if clinical outcome associated with any sample subgroups (i.e. colors separated by each subgroup if any)*

* Significant decrease of variance explained ratio at PC = 2, PC = 6, (and PC = 14?).
* Could use the number of PCs as a reference of how many features to include in classification model.

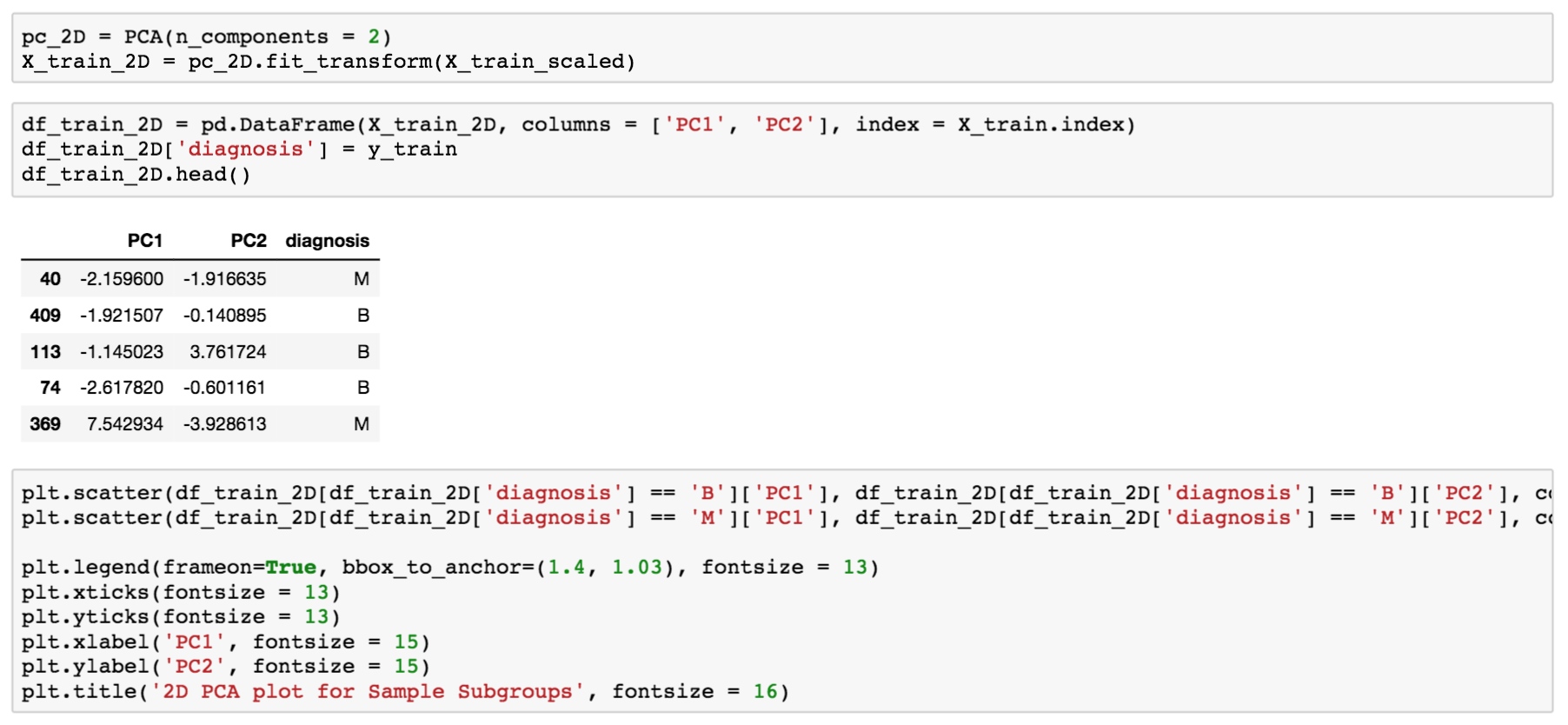
*Look at which original variable weight more in each PC:*

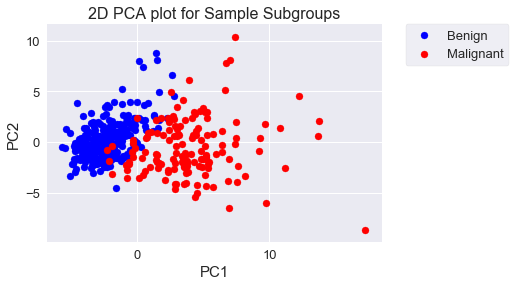


Look at which variable(s) have more impact in each PC. Could consider including these variables in the model:

* PC1: concavity\_mean, concavity points\_mean
* PC2: fractal\_dimension\_mean, fractal\_dimension\_worst
* PC3: texture\_se, smoothness\_se
* PC4: texture\_mean, texture\_worst
* PC5: smoothness\_mean, concavity\_se
* PC6: symmetry\_se, symmetry\_worst

*Dimensionally reduce data to 2D to:*

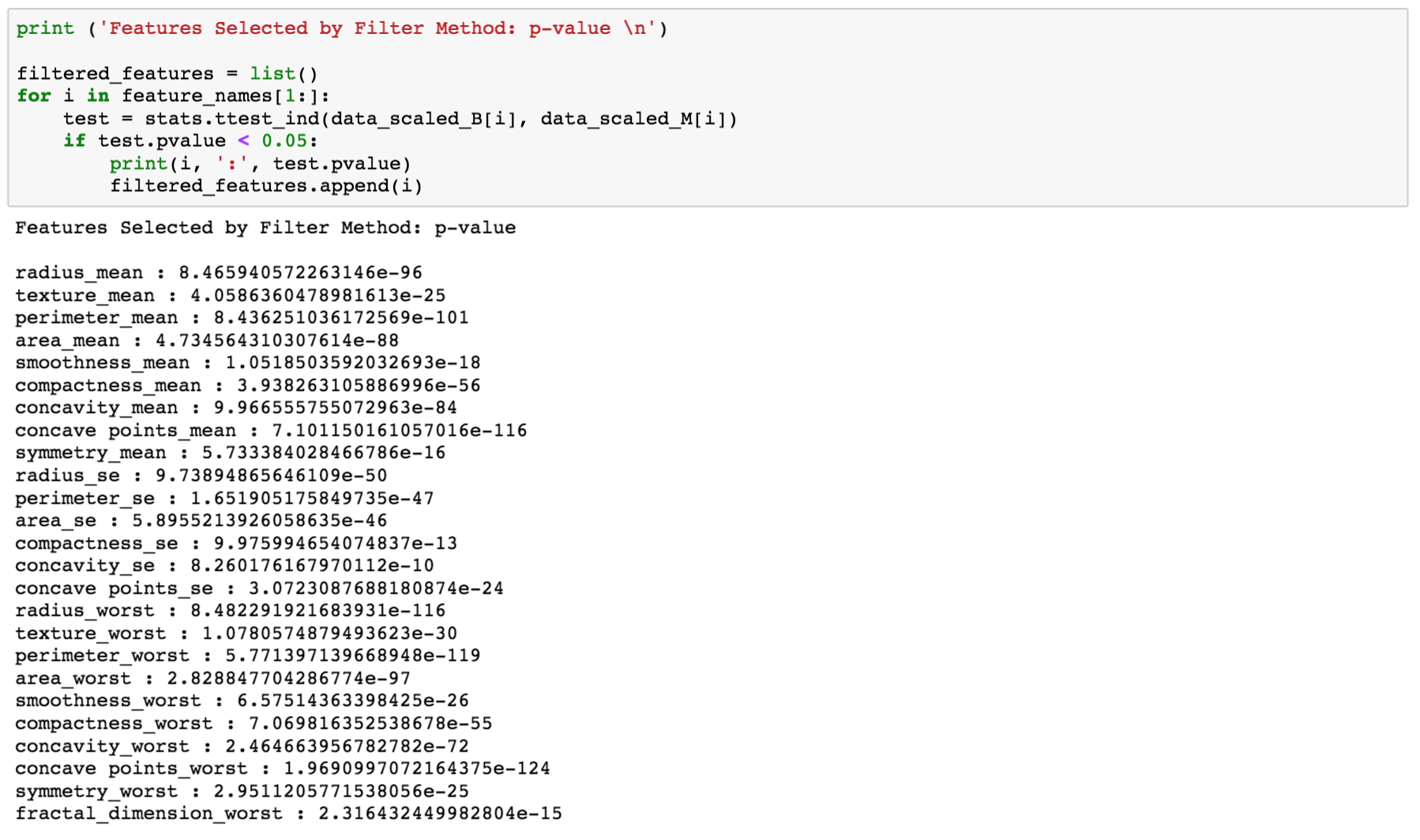
* *Visualize data points*
* *To identify any sample subgroups*
* *To determine if clinical outcome associated with any sample subgroups (i.e. colors separated by each subgroup if any)*



* Sample subgroups relatively well-determined by diagnosis status.

1. **Filter Method for feature selection** **based on Chi-square test:**

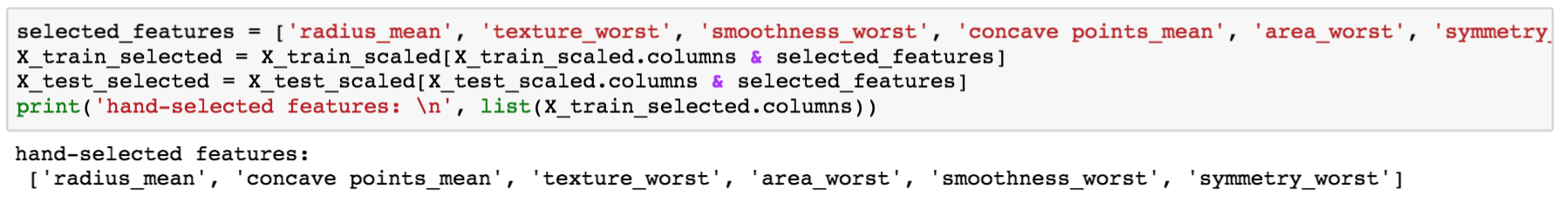
* Initial selection of features. Select features with significant t-test scores
* However, filter method only select features based on statistical score of feature by outcome, it does not handle multicollinearity and includes all correlated features if they are important

****

* A total of 25 features selected by filter methods.

1. **Combine observation from boxplot, swarmplot, multicollinearity, PCA, and filter selection. Hand-select the following features considering:**

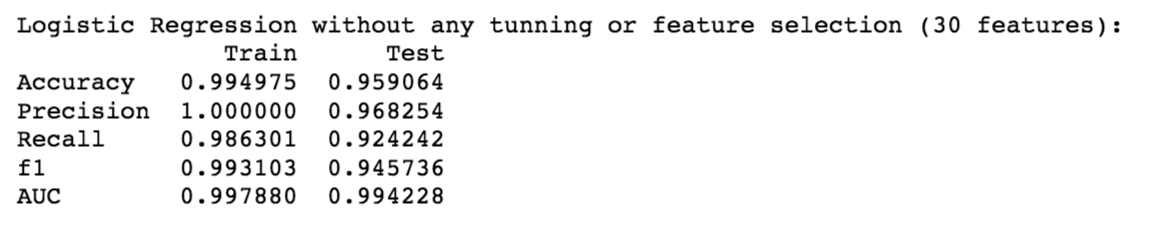
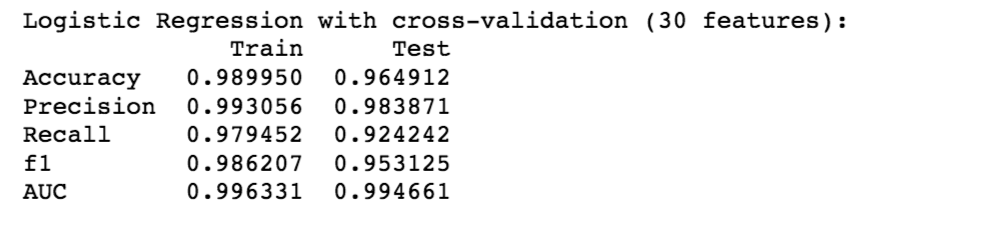
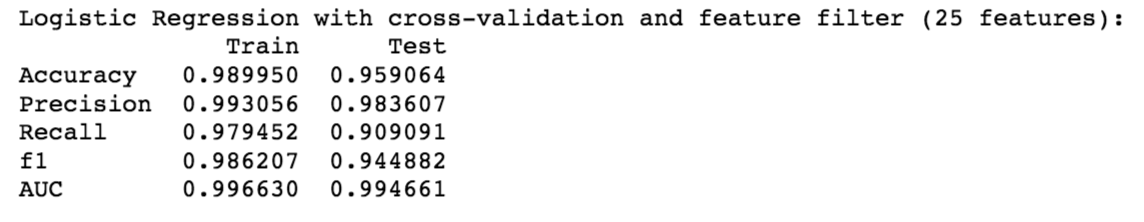
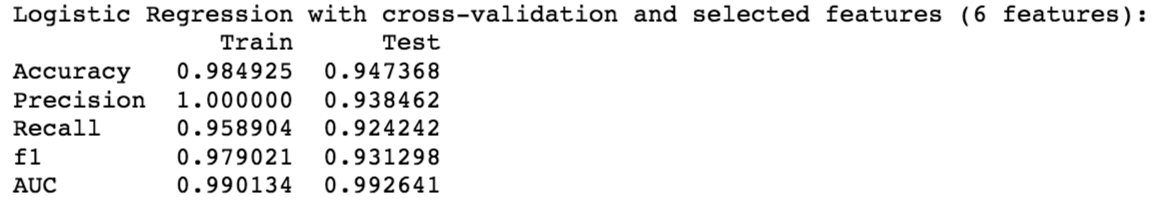
* Association with outcome variable
* Variance explained
* Multicollinearity

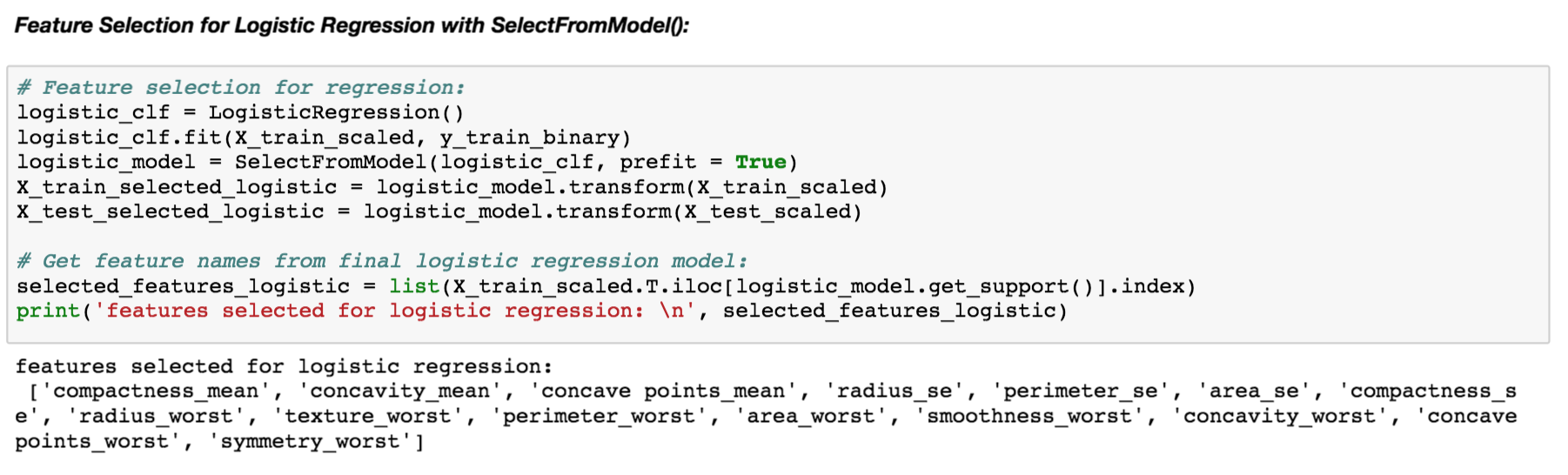


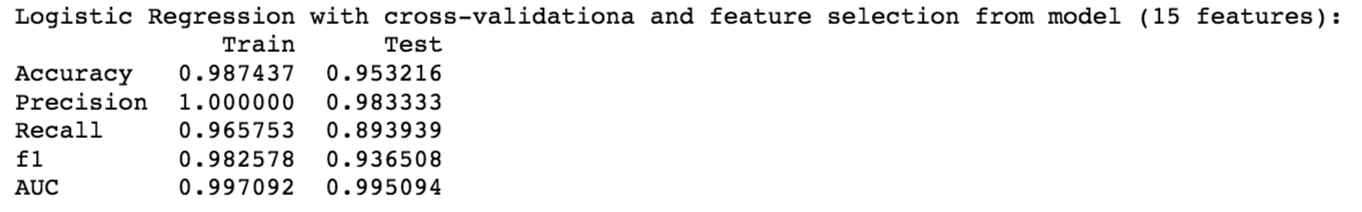
* Features selected: *{radius\_mean, concave\_points\_mean, texture\_worst, area\_worst, smoothness\_worst, symmetry\_worst}*

**CLASSIFICATION MODELS:**

1. **Model 1 - Logistic Regression**:

* From EDA, we know multicollinearity exist between features selected above. Use regularization on logistic regression to reduce multicollinearity and overfitting. Use cross-validation to search for best regularization parameter c.
* *Performance no tuning original (30 features):*
* *Performance with cross validation & grid search original (30 features):*
* *Performance with cross validation & grid search filtered (25 features):*
* *Performance with cross validation & grid search hand selected (6 features):*

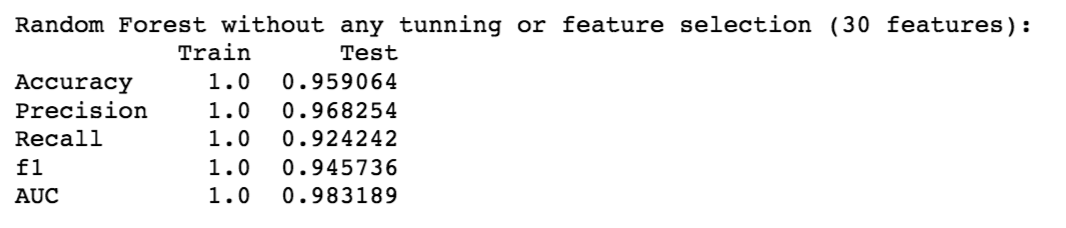
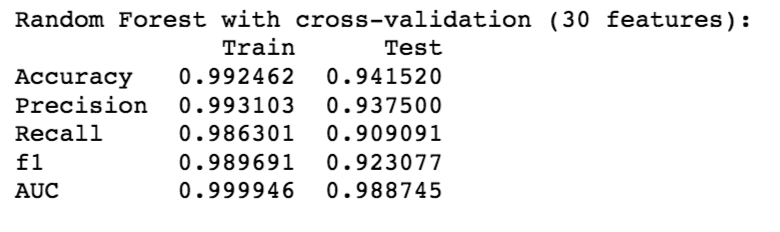
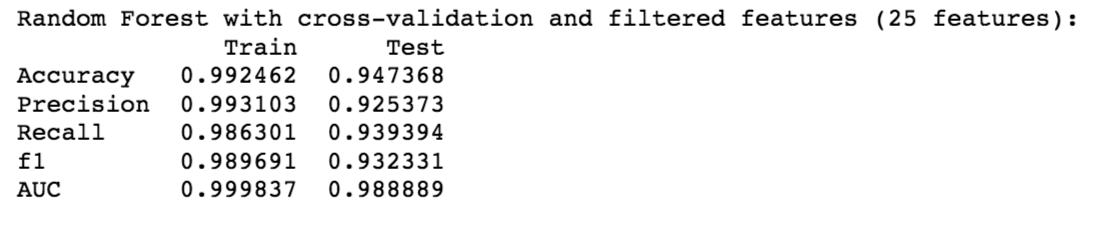
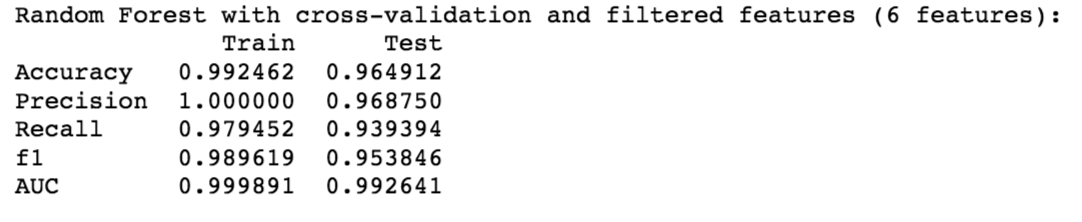


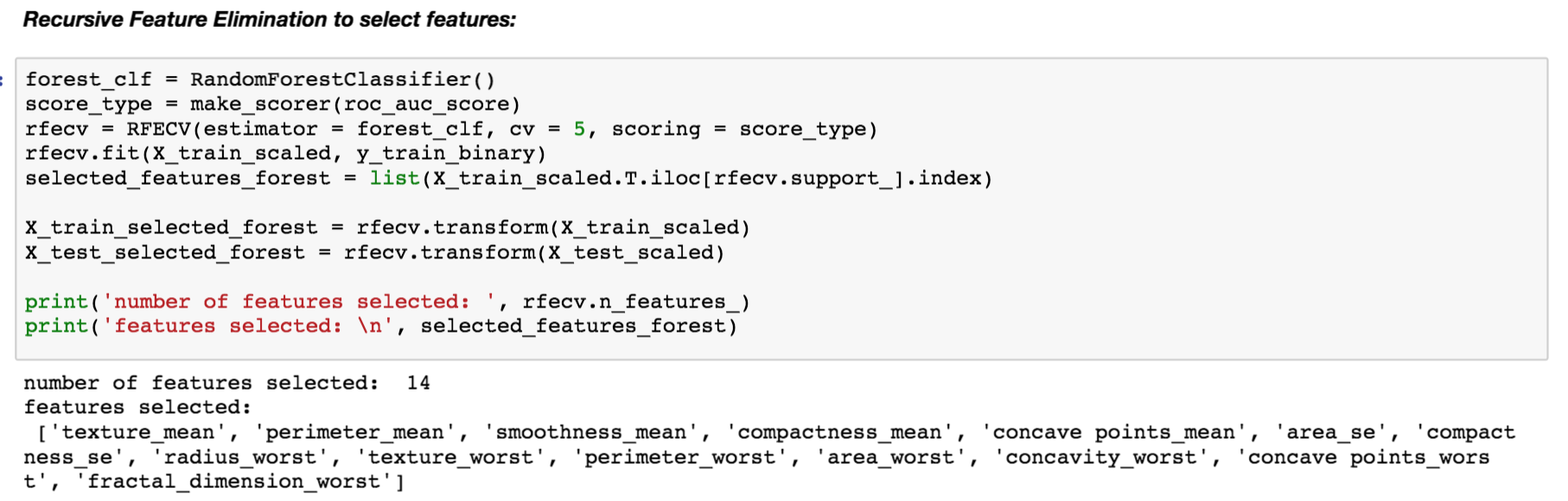
* *Performance with cross validation & grid search model selected (15 features):*
* Pros vs. Cons of logistic regression:

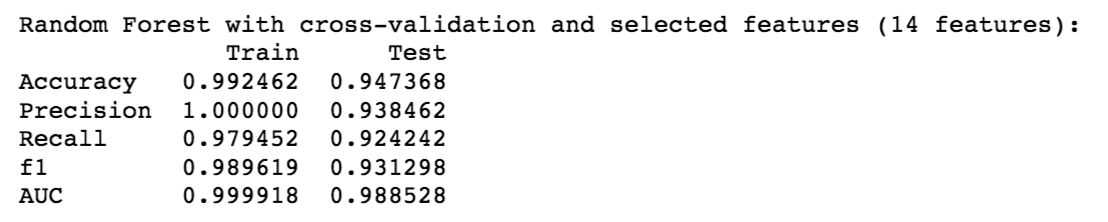
(-) prone to overfitting, (-) prone to outliers, (+) feature importance interpretable (standardized).

* It can be seen from the above results, logistic regression had very good performance on both normalized training and testing datasets. Tuning regularization parameter C slightly reduced overfitting on training data. However, further feature selections did not improve model.

1. **Model 2 - Random Forest**:

* *Performance no tuning original (30 features):*
* *Performance with cross validation & grid search original (30 features):*
* *Performance with cross validation & grid search filtered (25 features):*
* *Performance with cross validation & grid search hand selected (6 features):*
* *Performance with cross validation & grid search model selected (14 features):*



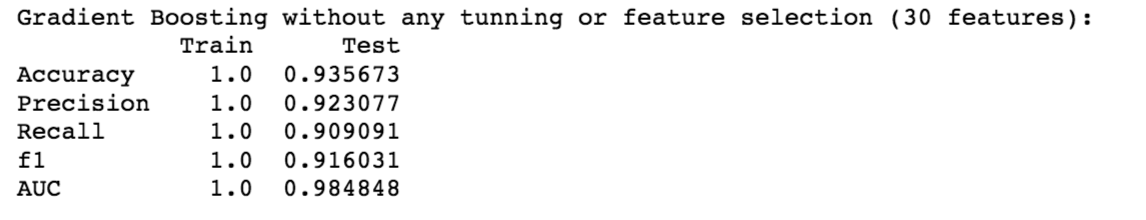
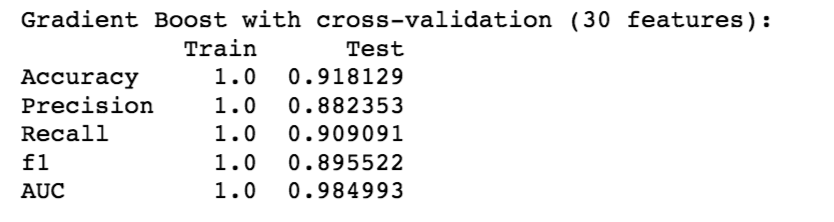
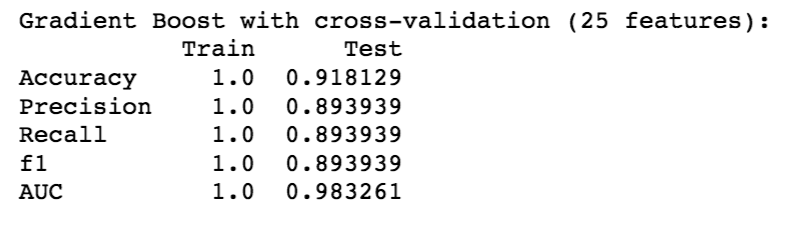
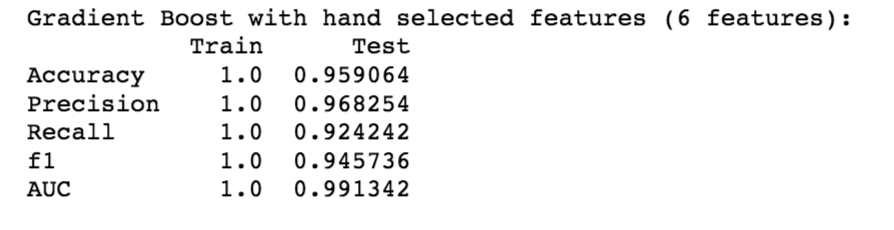
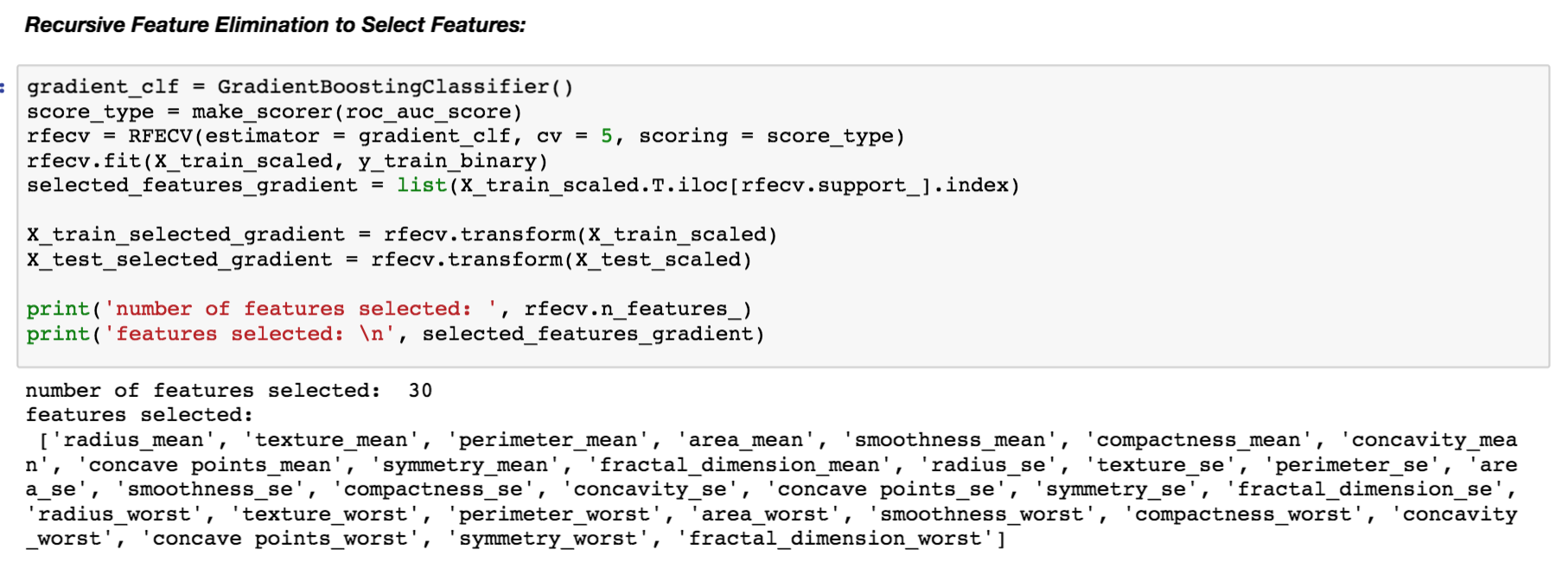


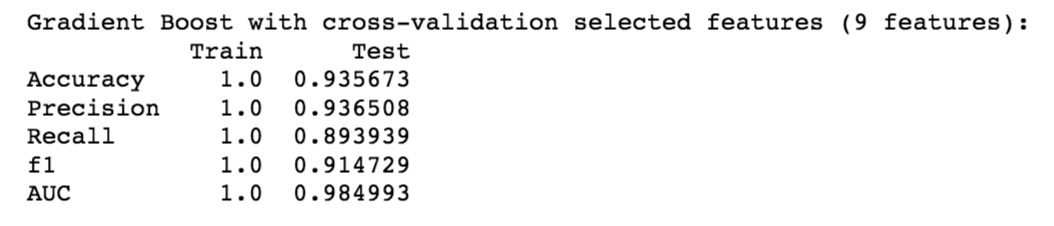
* Pros vs. Cons of random forest:

(+) not prone to multicollinearity, (+) no linearity assumptions, (+) robust to high-dimension data, (+) can directly output feature importance, can visualize feature importance, (-) slow training, may still over-fit

* Tuning did not improve model much. But reducing features hand-selected 6 features improved accuracy on test data.

1. **Gradient Boosting Tree**:

* *Performance no tuning original (30 features):*
* *Performance with cross validation & grid search original (30 features):*
* *Performance with cross validation & grid search filtered (25 features):*
* *Performance with cross validation & grid search hand selected (6 features):*
* *Performance with cross validation & grid search model selected (14 features):*

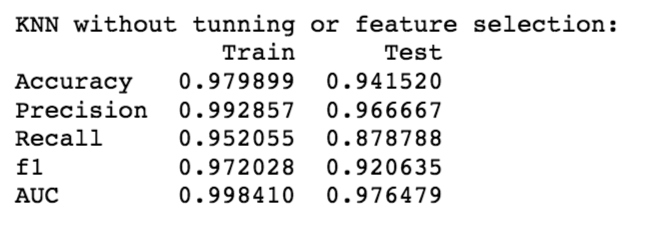
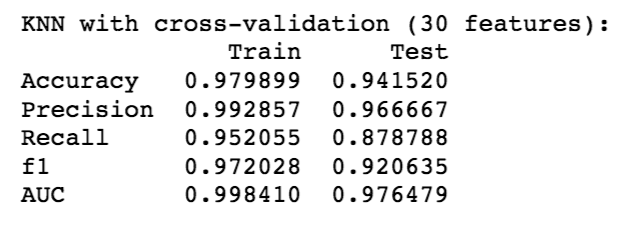
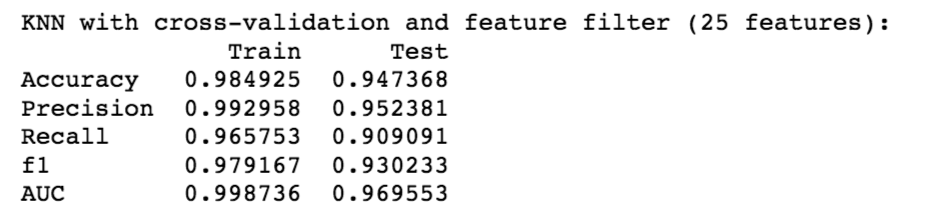
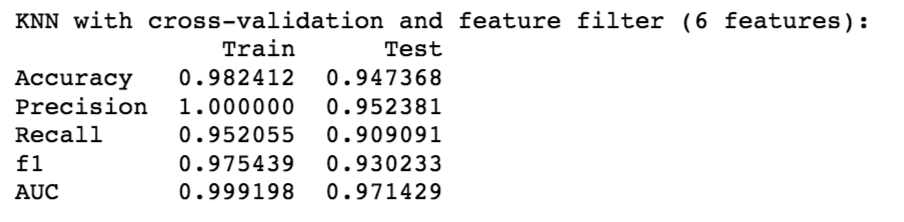


* Pros vs. Cons of gradient boosting tree:

(+) no linearity assumptions, (+) easy tuning, (-) performance influenced by base learner, (-) performance influenced by outliers

* Very good performance on training set, but overfitting problem present. Performance on testing data improve after reducing features to 6.

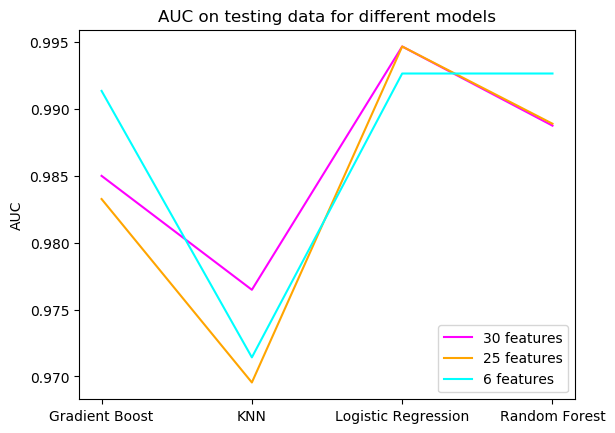
1. **KNN**: Cross-validation to find optimal k (selected features)

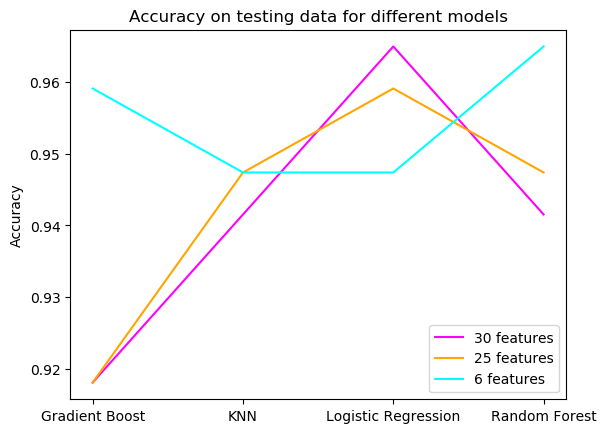
* *Performance no tuning original (30 features):*
* *Performance with cross validation & grid search original (30 features):*
* *Performance with cross validation & grid search filtered (25 features):*
* *Performance with cross validation & grid search hand selected (6 features):*
* Pros vs. Cons of KNN: (+) no linearity assumptions, (+) fast train, (-) curse of dimensionality
* Very similar performance with/without tuning and feature selection.

**COMPARE MODEL PERFORMANCE:**

Compare model performance on sets of features selected through:

1. Filter methods: features with top chi-sq scores with outcome variables (do not for sure reduce multicollinearity)
2. Embedded methods: regularization (reduce multicollinearity in logistic regression)
3. Wrapper methods: recursive feature elimination to determine optimal number of features (reduce multicollinearity)

* Plot AUC and Accuracy for each model for each set of features to visualize model performance:



* No model is universally better on this dataset.
* Generally, regularized logistic regression have better performance with more features. Tree-based methods have better performance with less features. KNN has worst AUC for all feature sets.
* Note the ranges of y axis on above plots are small, meaning all model have similar performance. - Nevertheless, all models achieved >95% accuracy and >0.90 AUC on predicting breast cancer for this dataset after tuning and feature selection.