Classify Malignant Cell from UCI Breast Cancer (Diagnosis) Dataset

Yet Another Revisit to the Classic Classification ML Model and the Alternative Approaches

Kai Zhao
Data Scientist
8/27/2019

Background - Breast Cancer

- About 1 in 8 U.S. women (about 12%) will develop invasive breast cancer over the course of her lifetime.
- For women in the U.S., breast cancer death rates are higher than those for any other cancer, besides lung cancer.
- Besides skin cancer, breast cancer is the most commonly diagnosed cancer among American women. In 2019, it's estimated that about 30% of newly diagnosed cancers in women will be breast cancers.

US Breast Cancer Statistics. Breastcancer.ort

Background - Diagnosis

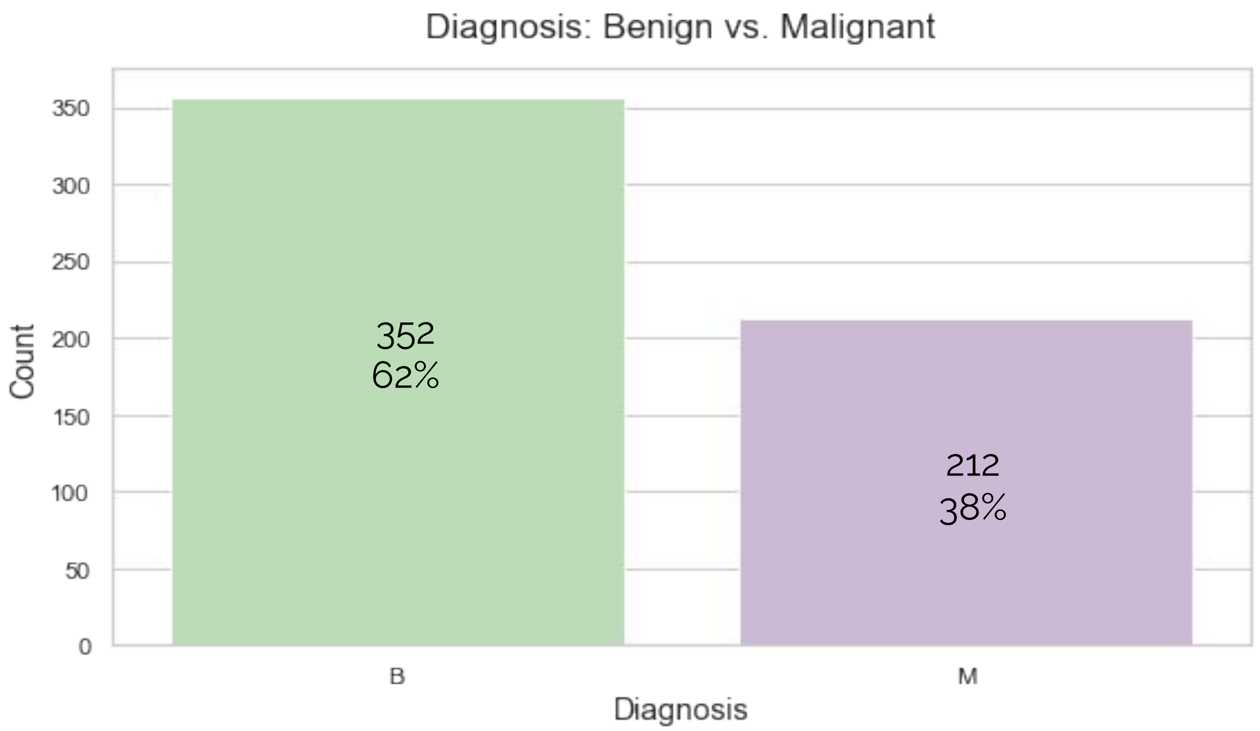
Diagnosis	Accuracy
Mammography	68% - 79%
Surgical biopsy	~100%
Fine Needle Aspiration (FNA)	65% - 98%

Table of Content

1. Explore the Data

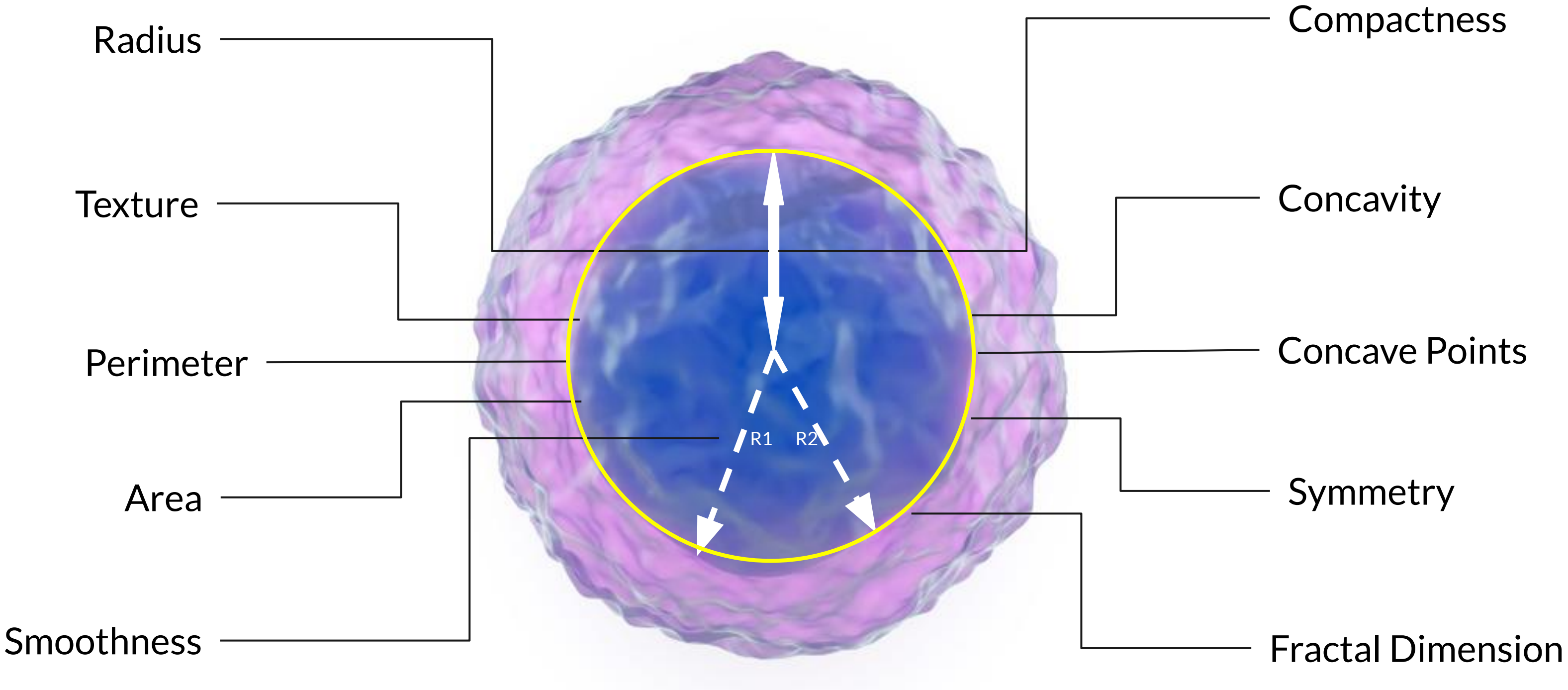
Class - Countplot

- Breast Cancer Diagnosis Dataset of 569 instances.
- Donnated by University of Wisconsin in 1995



K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34 ures

1. Explore the Data

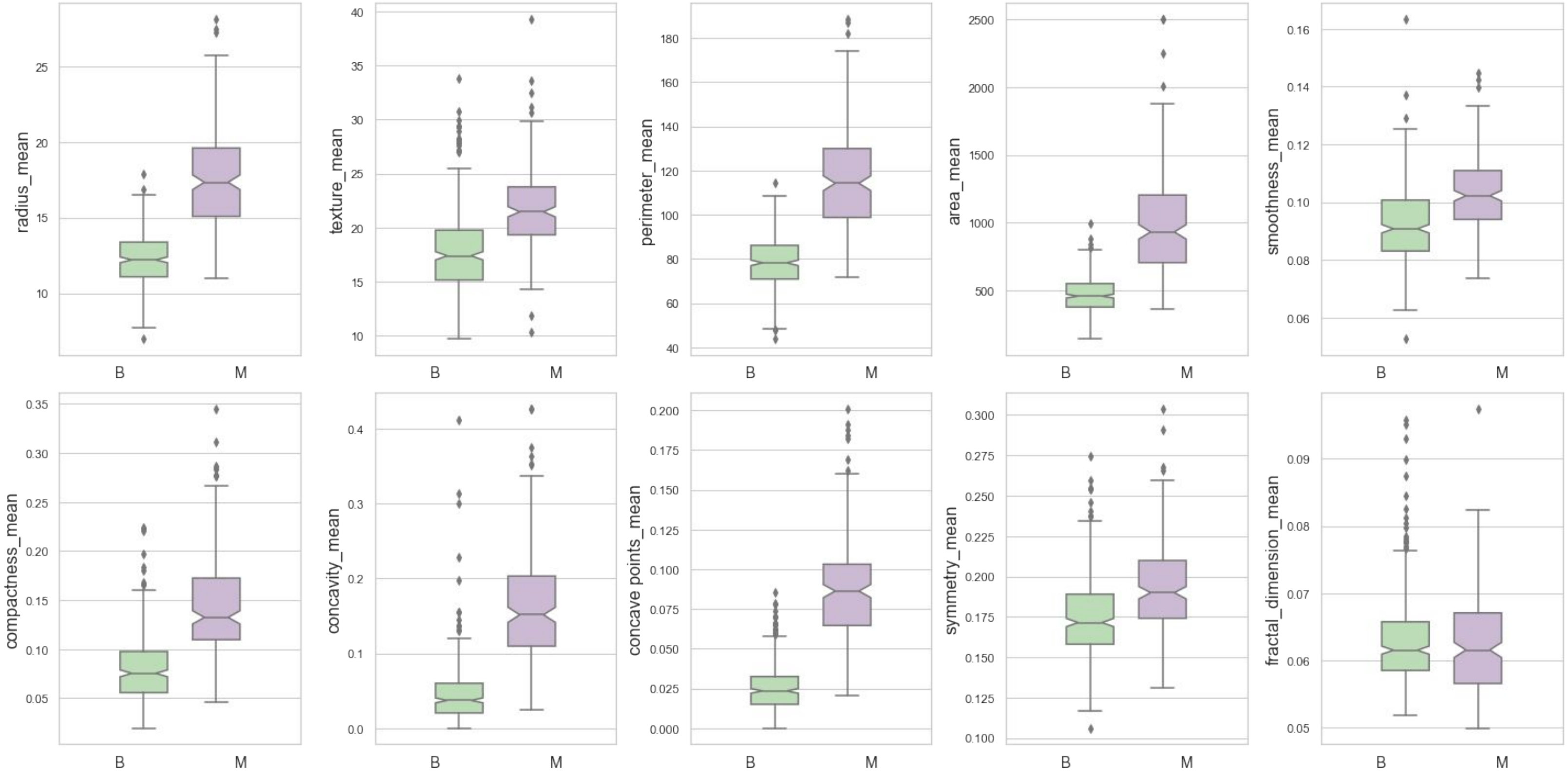


Mean, Standard Error, Worst - Total 30 Feats.

K. P. Bennett and O. L. Mangasarian: "Robust Linear Programming Discrimination of Two Linearly Inseparable Sets", Optimization Methods and Software 1, 1992, 23-34

1. Explore the Data

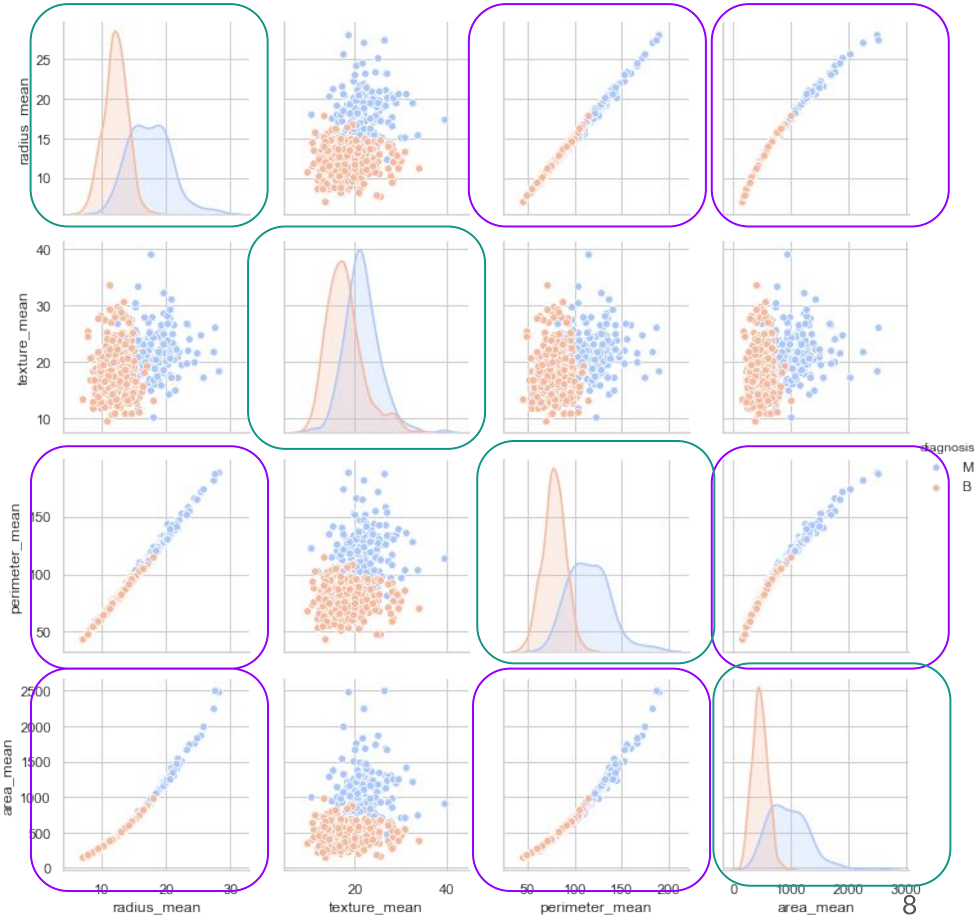
Cell Features - Boxplot (1D)



1. Explore the Data

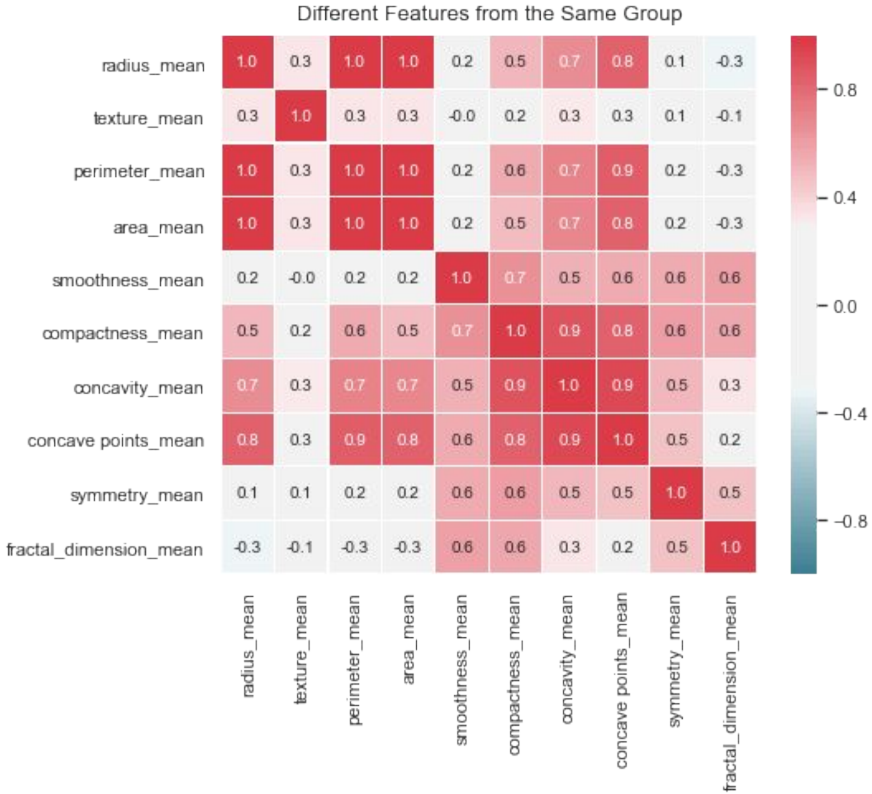
Features - Pairplot (2D)

- Feature Collinearity
- Class Overlap (2D not enough)
- Higher Dimension Needed



1. Explore the Data

Features - Heatmap



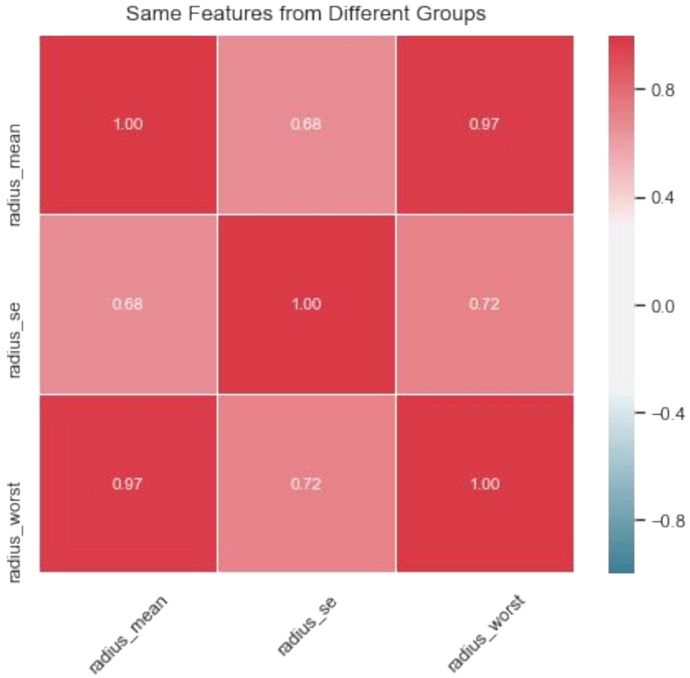
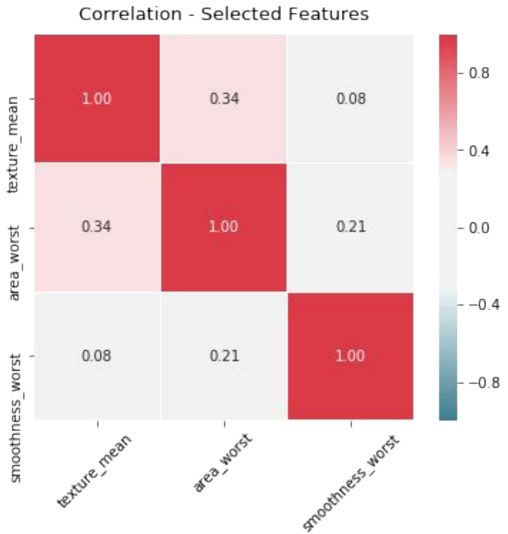


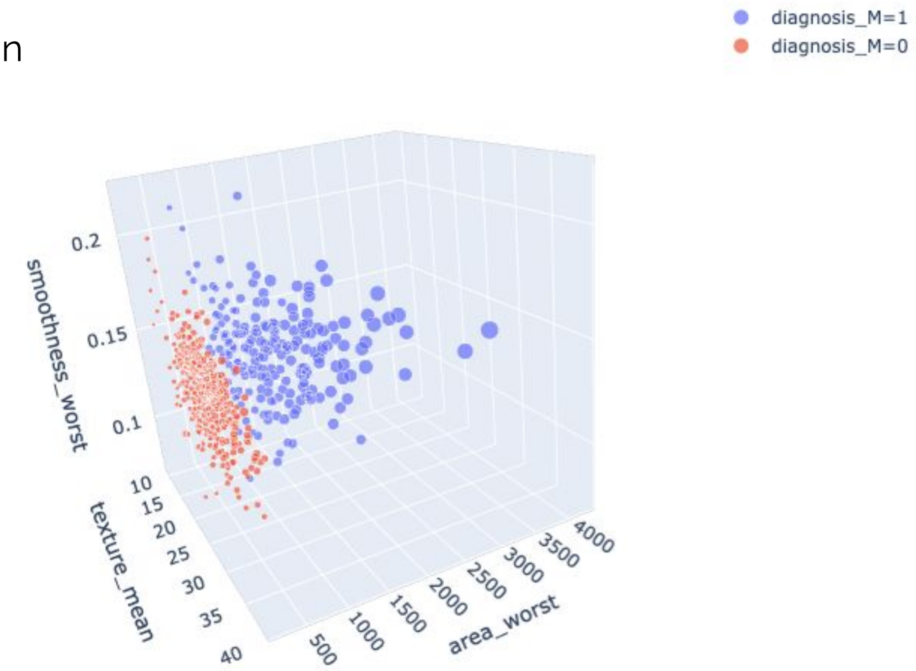
Table of Content

Review Paper & Select Model

Feature Selection Method - Feature Elimination

1. Texture_mean
2. Area_worst
3. Smoothness_worst





To Jupyter Notebook

2. Reproduce the 1994 Model

Review Paper & Select Model

Model Algorithm - Multisurface Method - Tree

(1) $$x^T w = \gamma,$$

Similar to linear kernel function

if and only if

(2) $$Aw \geq e\gamma + e, \; Bw \leq e\gamma - e.$$

Margine

(3)
$$\begin{array}{lrcl} \underset{w,\gamma,y,z}{\text{minimize}} & \dfrac{e^T y}{m} + \dfrac{e^T z}{k} & & \\ \text{subject to} & Aw + y & \geq & e\gamma + e \\ & Bw - z & \leq & e\gamma - e \\ & y, z & \geq & 0. \end{array}$$

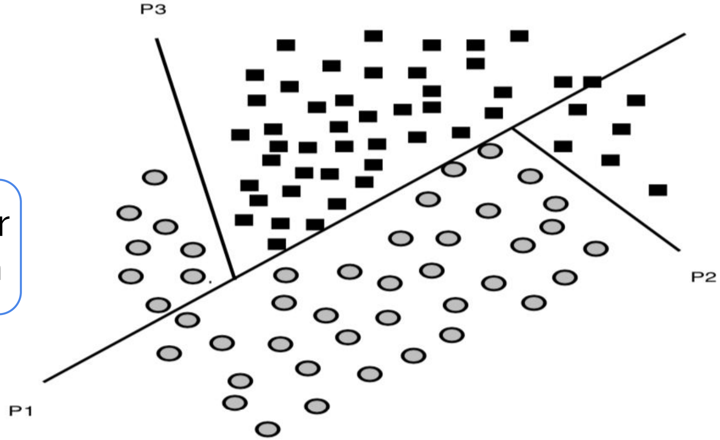


Figure 2: MSM-T separating planes.

My Approach:

Support Vector Machine

2. Reproduce the 1994 Model

Review Paper & Select Model

Target Metric:

97.5%

Based on

Cross-Validation

2. Reproduce the 1994 Model

Support Vector Classifier

```
# Instantiate pipeline
pipe = Pipeline([
    ('sc', StandardScaler()),
    ('svc', SVC(random_state=42, probability=True))
])

# Model parameters for GridSearch
param_grid = {
    'svc__kernel': ['rbf'],
    'svc__C': np.logspace(-3, 3, 7),
    'svc__gamma': np.logspace(-3, 3, 7)
}

# Instantiate GridSearch
search = GridSearchCV(pipe, param_grid, cv=5, verbose=1, n_jobs=-1)
```

2. Reproduce the 1994 Model

Model Outcome

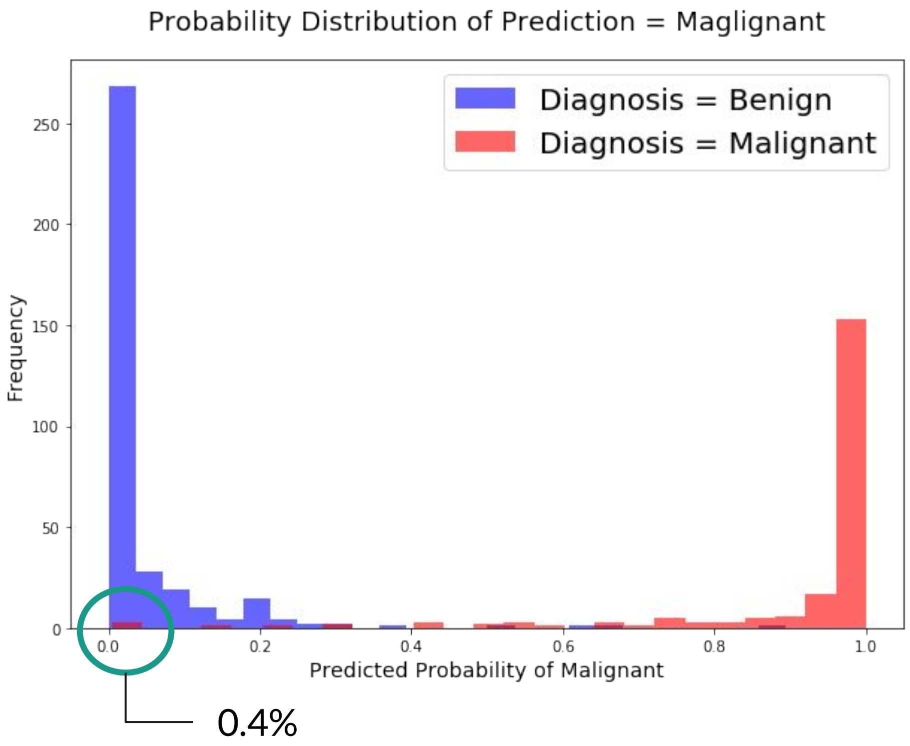
Achieved Metric:

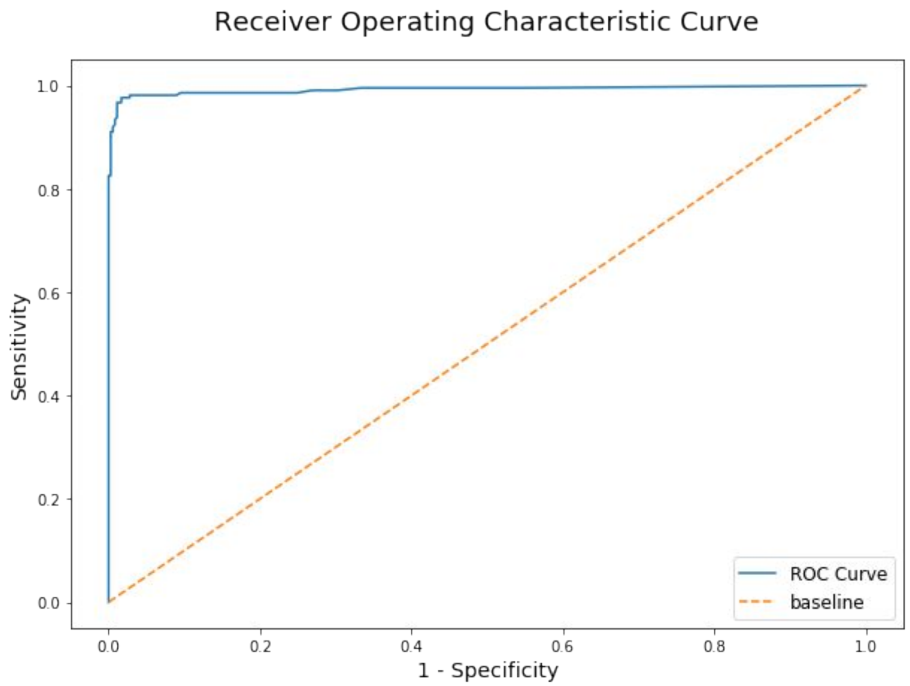
Metric	Score
CV Accuracy	97.5%
Train Recall	95.3%
Train F1-Score	96.7%

Confusion Matrix:

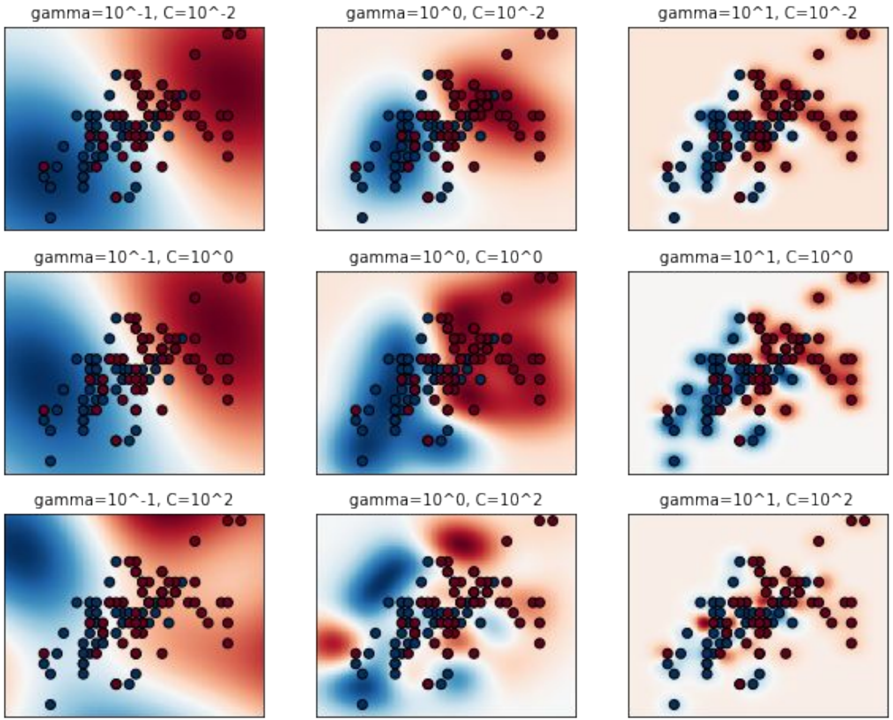
	Pred True	Pred False
Actual True	202	10
Actual False	4	353

Model Evaluation





Model Evaluation: C & Gamma



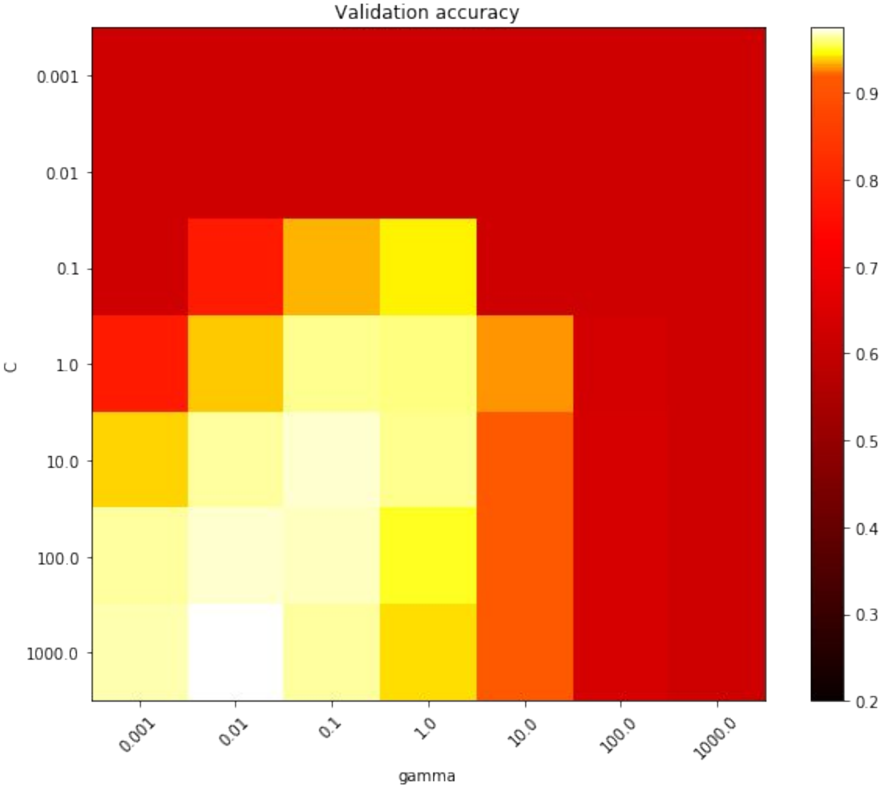
Gamma: the gamma parameter defines how far the influence of a single training example reaches, with low values meaning 'far' and high values meaning 'close'. The gamma parameters can be seen as the inverse of the radius of influence of samples selected by the model as support vectors.

C: the C parameter trades off correct classification of training examples against maximization of the decision function's margin. **- Regularization**

For larger values of C, a smaller margin will be accepted to correctly classify the training point.. A lower C will encourage a larger margin at the cost of training accuracy.

https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

Model Evaluation: C & Gamma



If gamma is too large, the radius of the area of influence of the support vectors only includes the support vector itself and no amount of regularization with C will be able to prevent overfitting.

When gamma is very small, the model is too constrained and cannot capture the complexity or "shape" of the data. The region of influence of any selected support vector would include the whole training set. The resulting model will behave similarly to a linear model with a set of hyperplanes that separate the centers of high density of any pair of two classes.

Good models can be found on a diagonal of C and gamma. Smooth models (lower gamma values) can be made more complex by increasing the importance of classifying each point correctly (larger C values) hence the diagonal of good performing models.

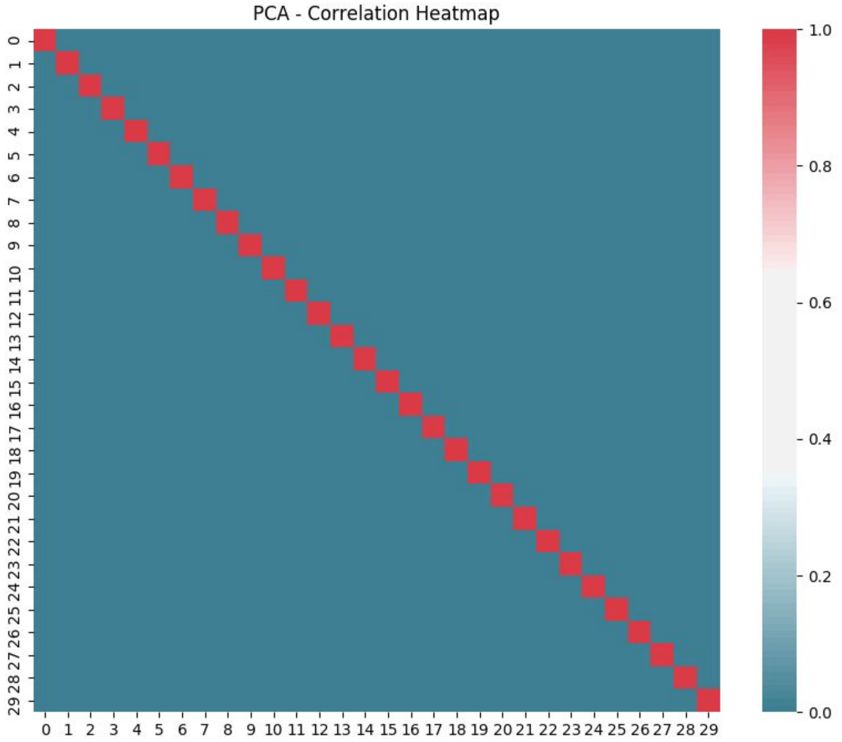
If possible, lower C is preferred for faster model.

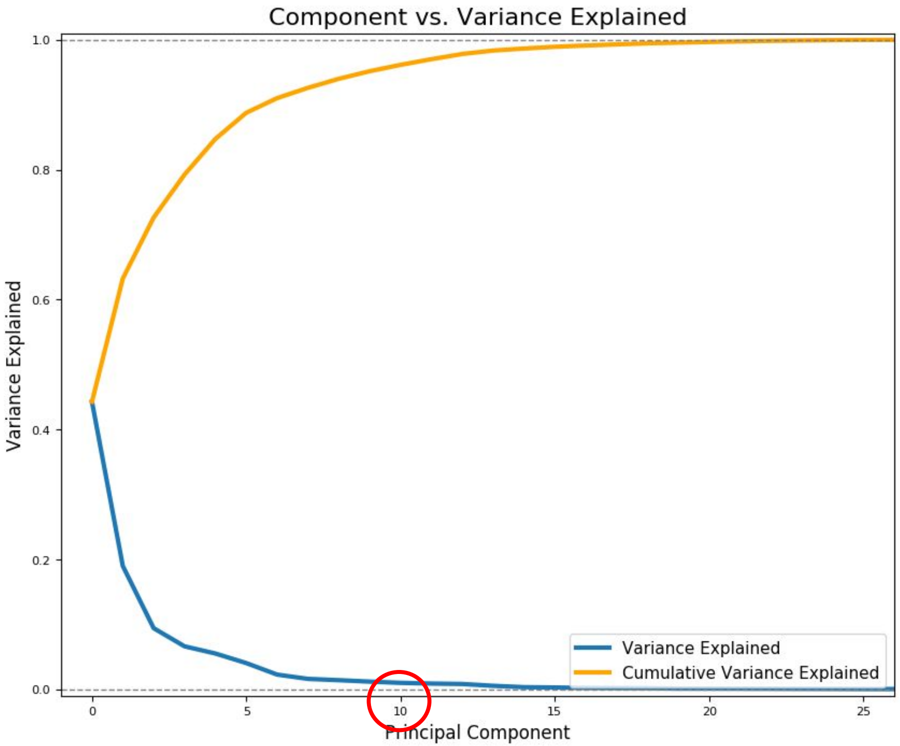
https://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

Table of Content

3. Remodel with Alternative Approaches

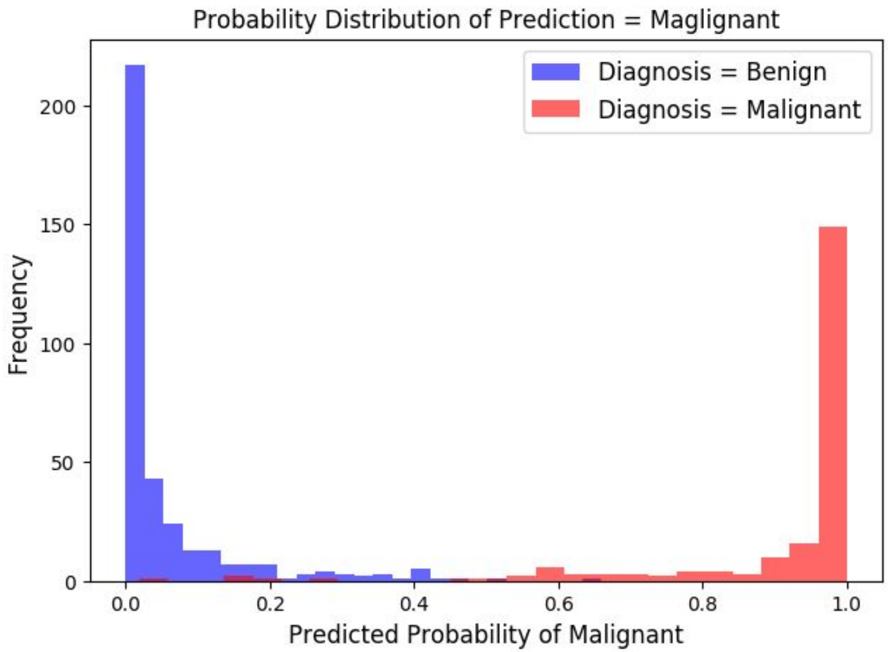
All features & Preprocess with PCA

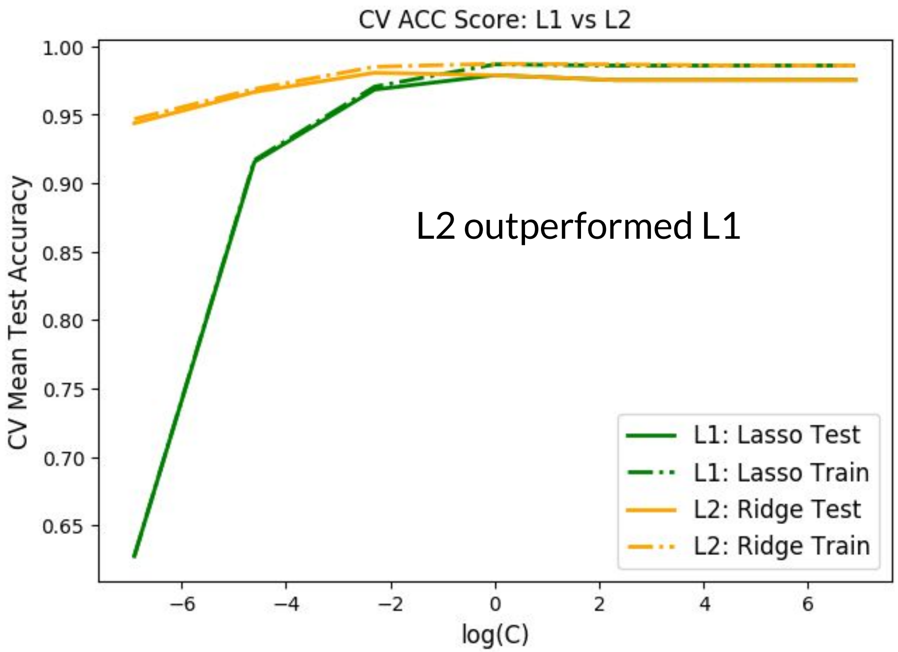




3. Remodel with Alternative Approaches

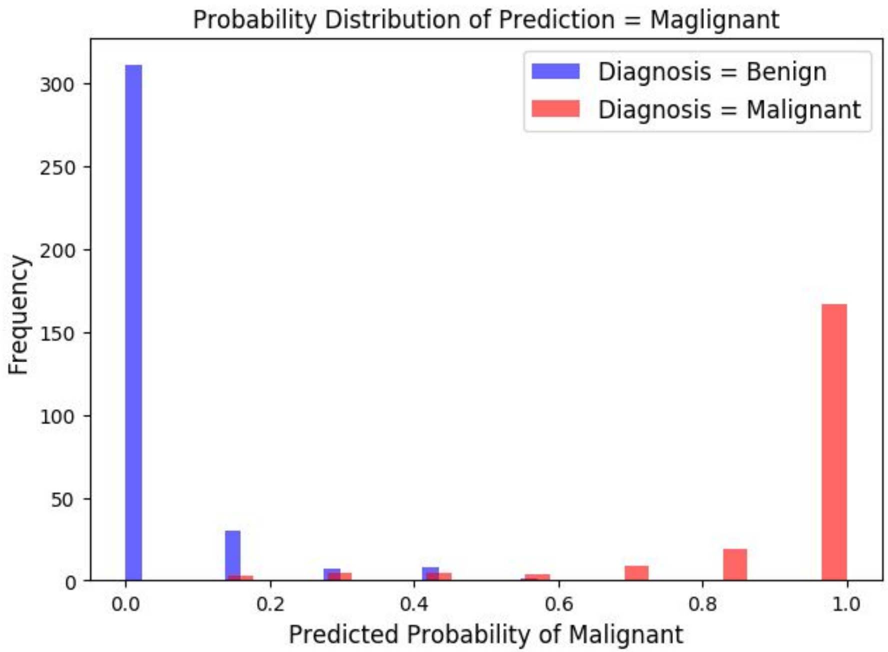
Logistic Regression

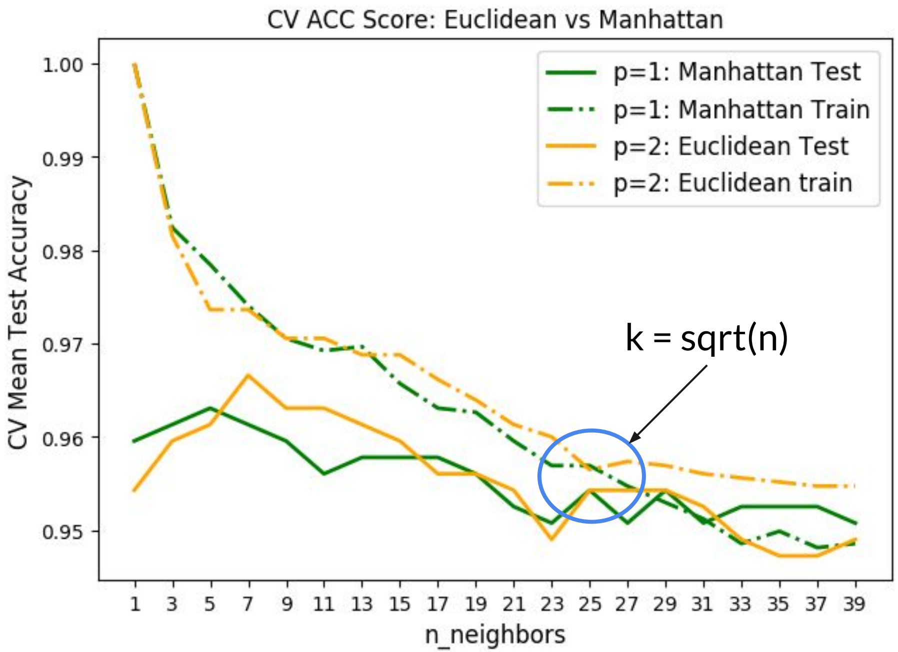




3. Remodel with Alternative Approaches

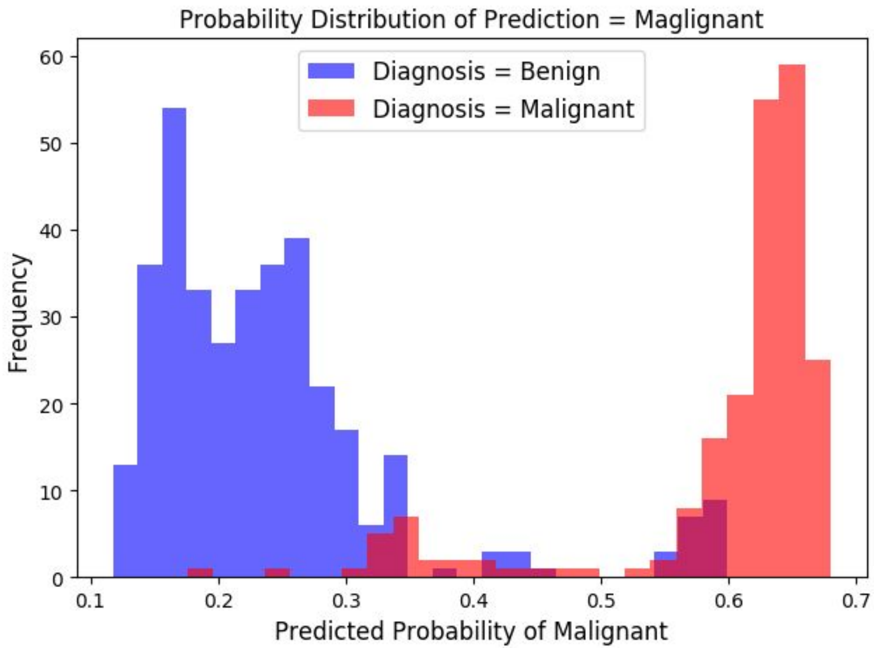
K-Nearest Neighbors

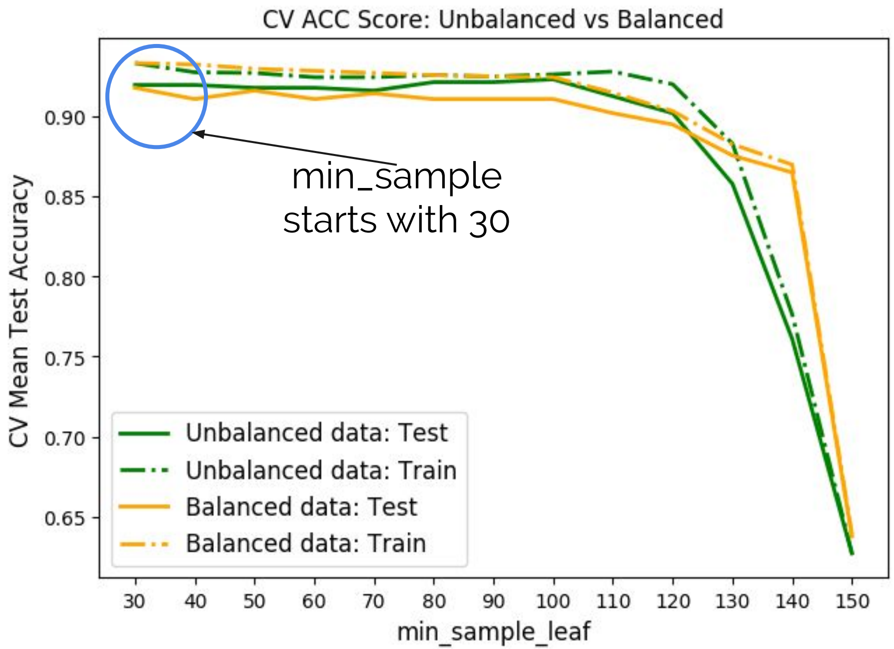




3. Remodel with Alternative Approaches

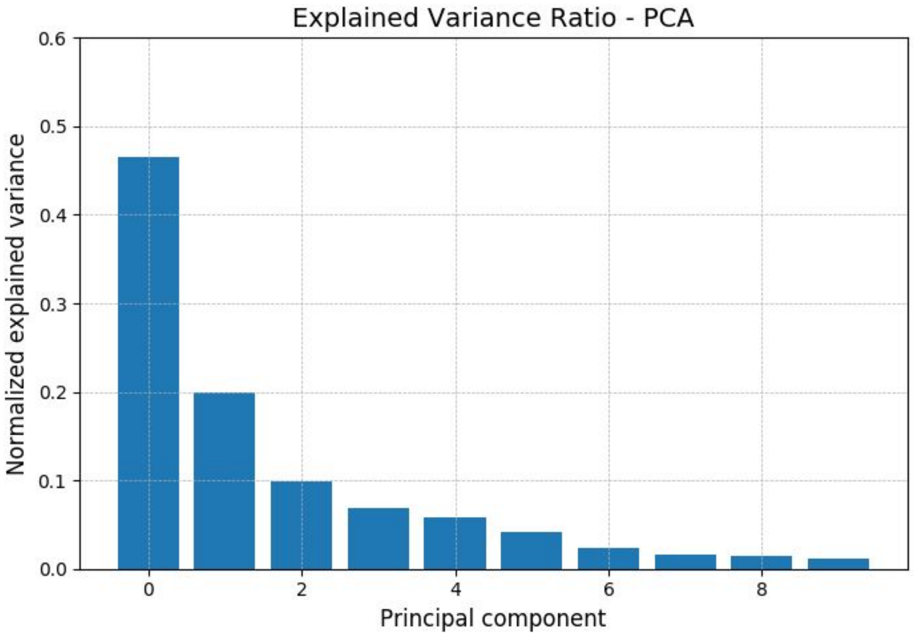
Random Forest

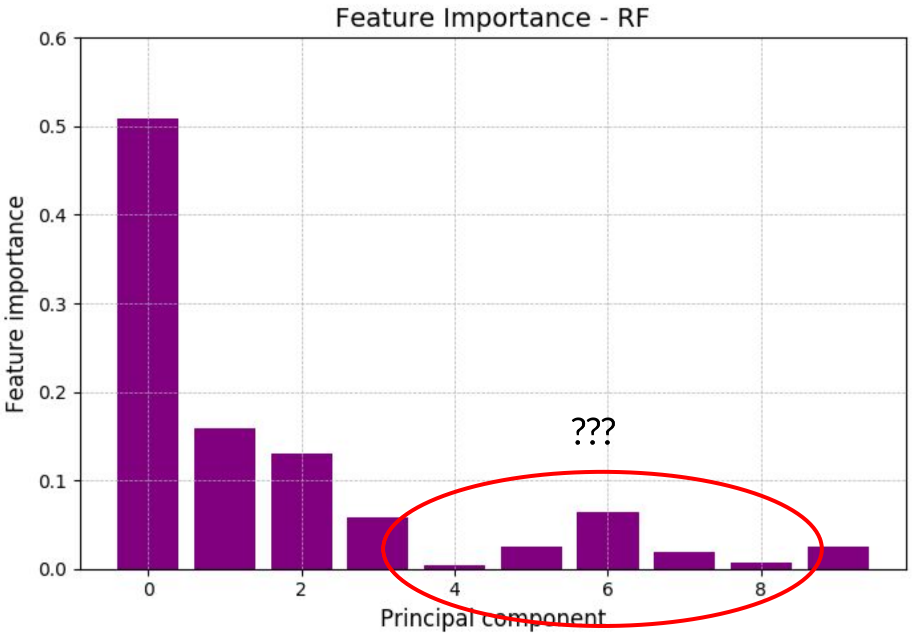




3. Remodel with Alternative Approaches

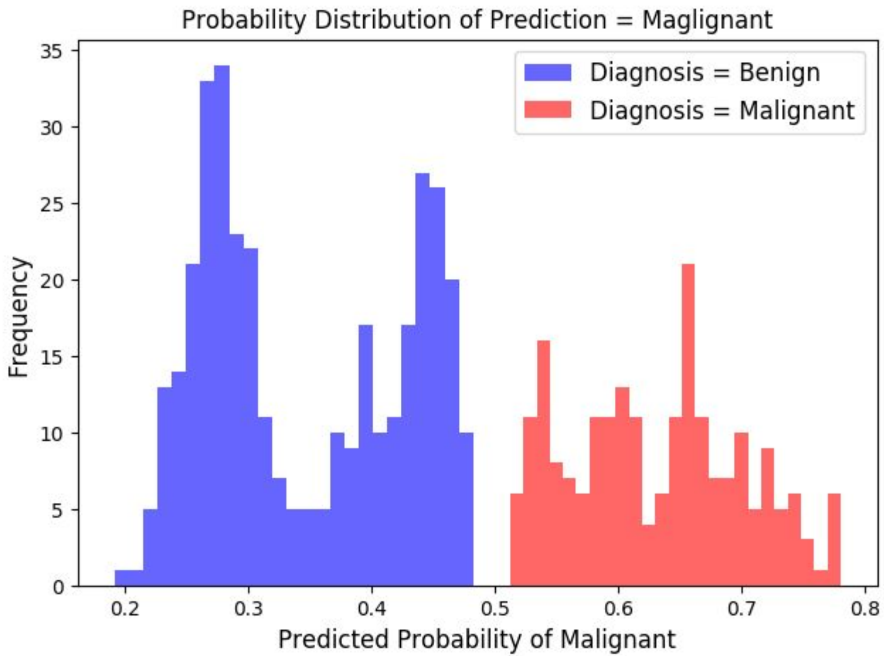
Random Forest: Feature Importance

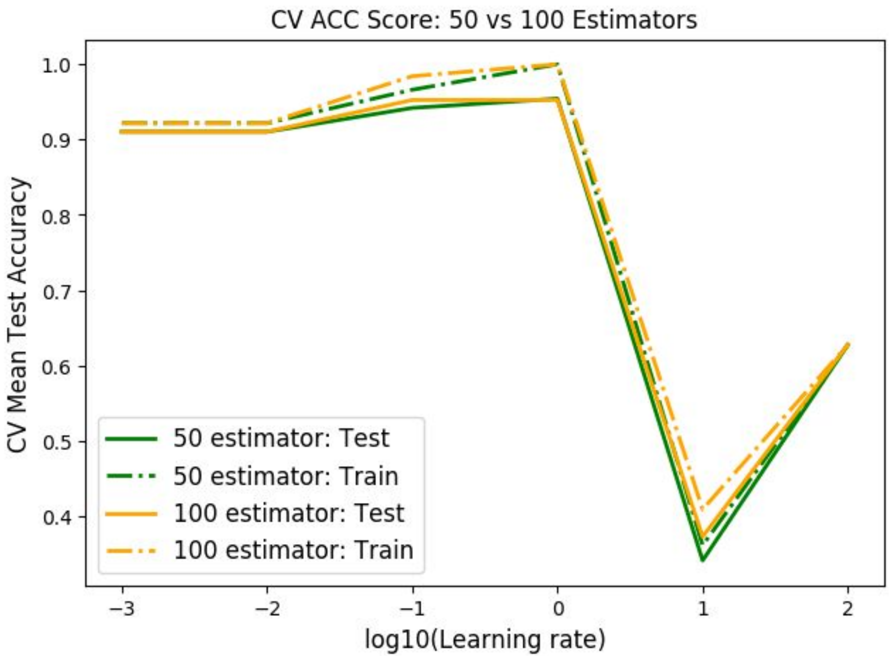




3. Remodel with Alternative Approaches

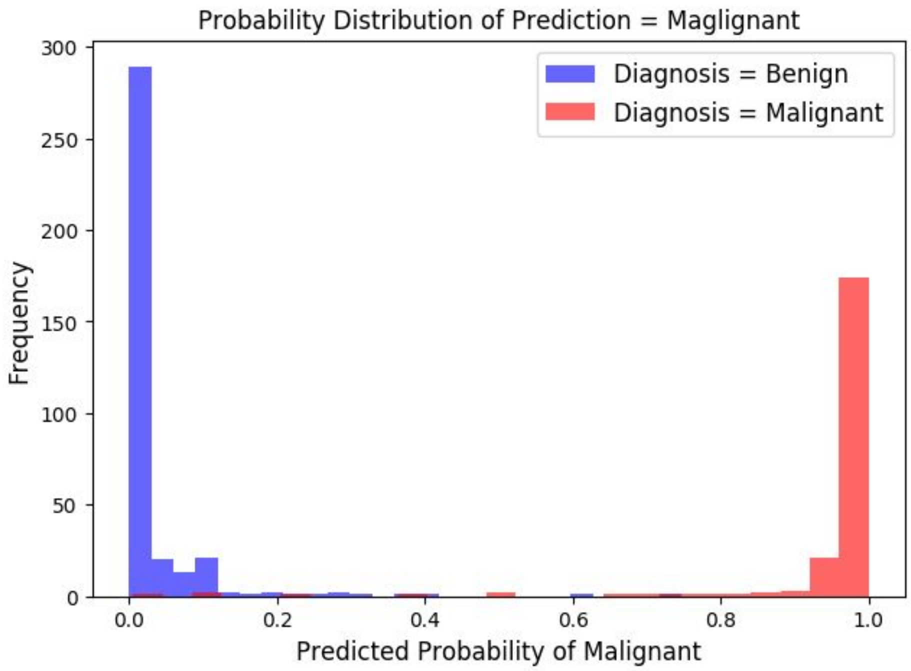
Adaboost

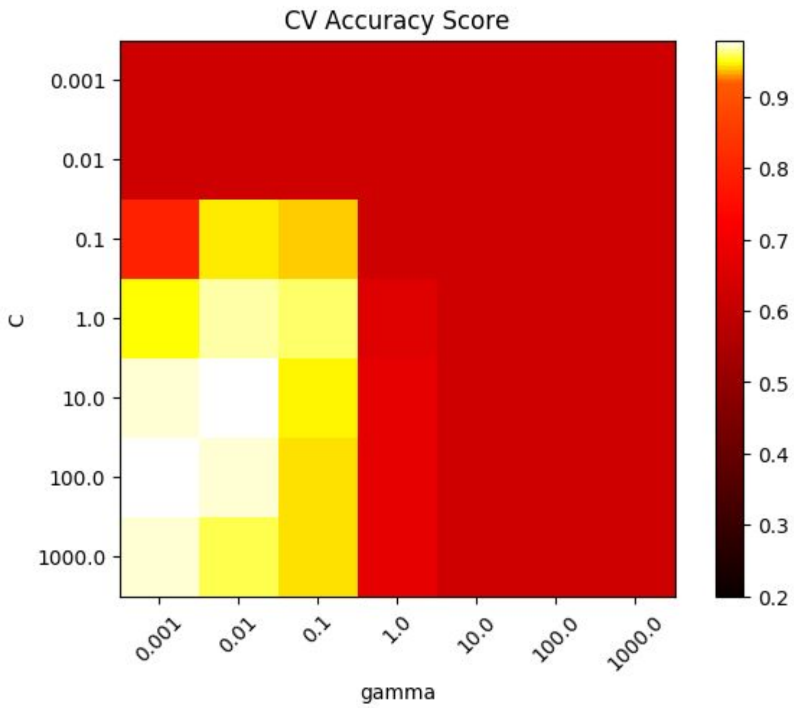




3. Remodel with Alternative Approaches

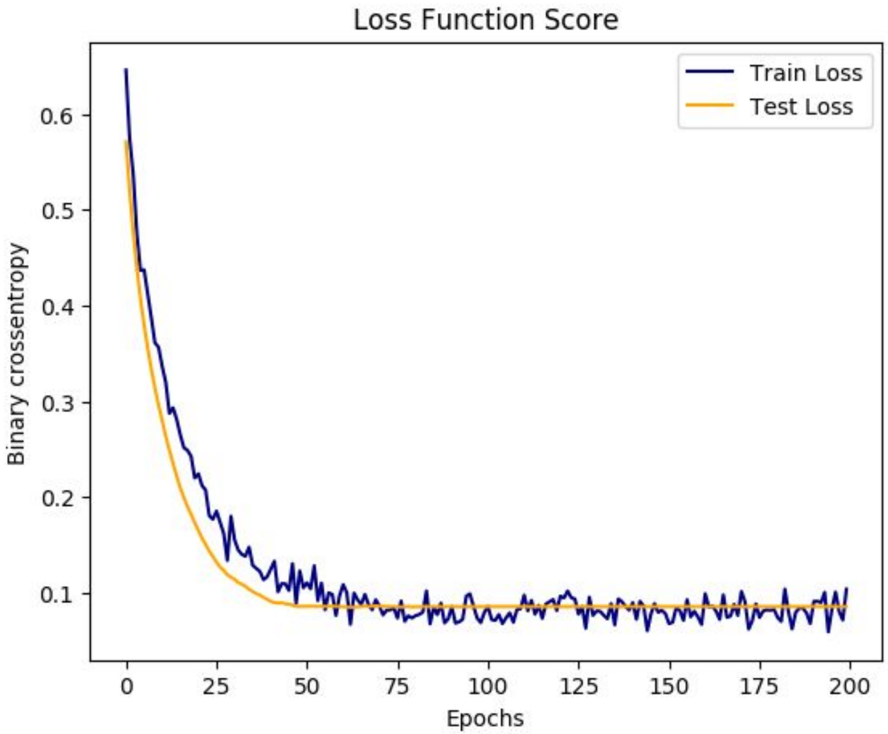
Support Vector Classifier

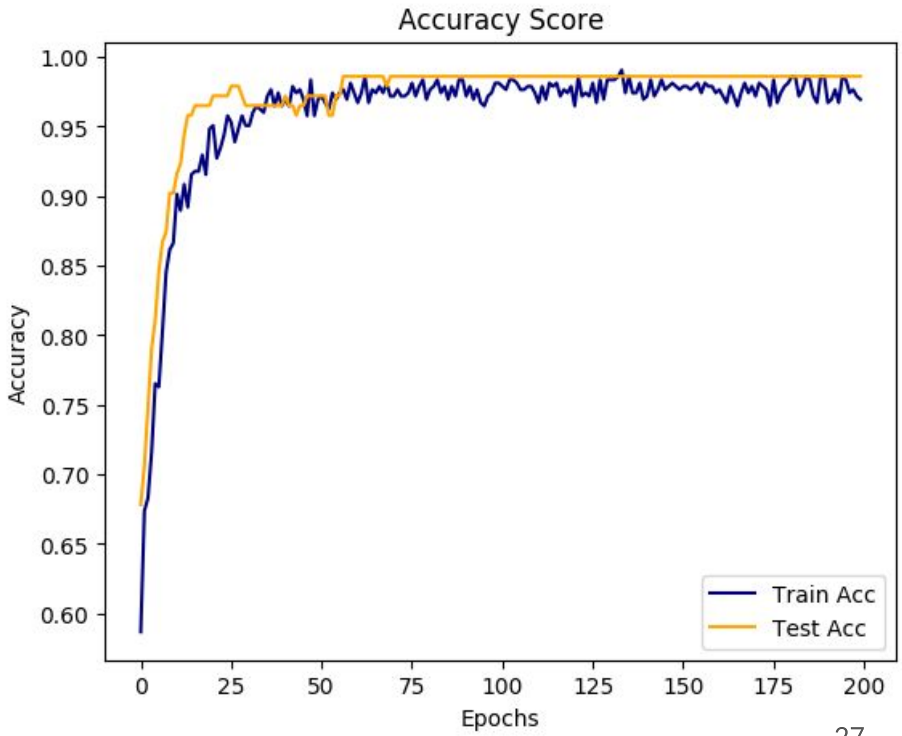




3. Remodel with Alternative Approaches

Neural Network





Summary

Metric	Paper	LR	KNN	RF	AB	SVC	NN
CV ACC	97.5%	98.1%	96.7%	92.3%	95.4%	97.9%	98.6%
Train F1	96.7%	98.1%	96.6%	89.5%	100%	97.9%	99.8%
Train FN	10	6	13	25	0	7	1
ID 297	0.4%	14.0%	14.3%	23.9%	51.3%	0.6%	60.5%

Conclusion & Next Step

Conclusion

1. We were able to reproduce the legacy model accuracy of 97.5% was SVC with proper hyperparameter tuning.
2. PCA works well with predictors with collinearity.
3. Neural Network outperforms all other model based on cross-validation accuracy but at greater computational cost.
4. Feature extraction provides model with more information at the cost of the interpretability of the features.

Nest Step:

1. Feature engineering.
2. Model using more recent data/gene data.
3. Breast cancer prognosis.
4. Image Processing - Cancer Cell Locator.

Table of Content

1. Explore the Data
    a. Class
    b. Features
2. Reproduce the Legacy 1994 Model
    a. Review Paper & Select Model
    b. Compare Outcome & Visualize Results
    c. Evaluate Model
3. Remodel with Alternative Approaches
    a. Feature: Elimination vs. Extraction
    b. Algorithms:
        i. Logistic Regression
        ii. K-Nearest Neighbors
        iii. Random Forest
        iv. Adaboost
        v. Support Vector Machine
        vi. Neural Network
4. Appendix: Avoid the Hidden Trap

4. Avoid the Hidden Trap

Can you find the issue of the following model?

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    stratify=y,
                                                    random_state=42)

sc = StandardScaler()
X_train_sc = sc.fit_transform(X_train)
X_test_sc = sc.fit_transform(X_test)

svc = SVC()

param = {
    'C': [0.1, 1, 10],
    'gamma': [0.1, 1, 10]
}

search = GridSearchCV(svc, param_grid=param, cv=5, verbose=1, n_jobs=-1)
search.fit(X_train, y_train)
```

Fitting train data transformer before GridSearchCV will leads to biased CV score.

4. Avoid the Hidden Trap When Tuning Model with CV

Let’s look for issues

```
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    stratify=y,
                                                    random_state=42)

pipe = Pipeline([
    ('sc', StandardScaler()),
    ('svc', SVC())
])

param = {
    'svc__C': [0.1, 1, 10],
    'svc__gamma': [0.1, 1, 10]
}

search = GridSearchCV(pipe, param_grid=param, cv=5, verbose=1, n_jobs=-1)
search.fit(X_train, y_train)
```

Solution: always put transformers in Pipeline!

Thank You!