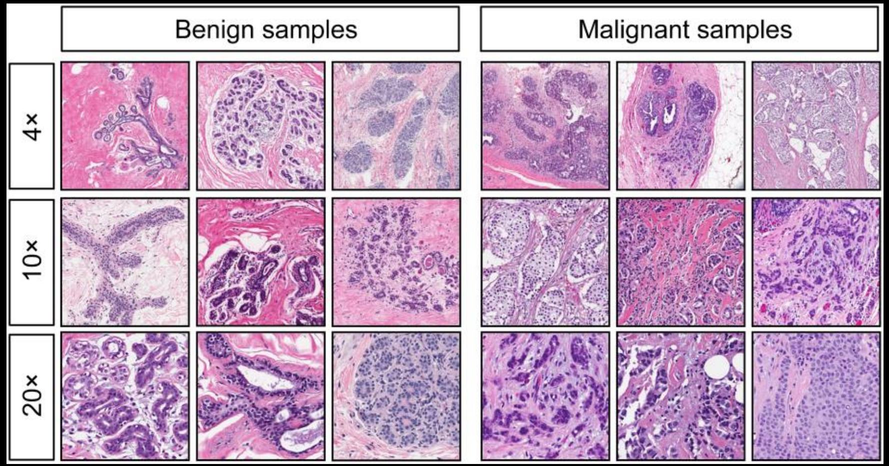
APPLICATION OF STATISTICAL LEARNING ALGORITHMS TO BREAST CANCER DIAGNOSTICS

BREAST CANCER

- Second most common cancer among women in the US
- Early diagnostics plays a crucial role in the success of the treatment

OUR GOALS

- Classification of tumors based on the tissue samples as benign and malignant
- Improving the efficiency of diagnostics
- Getting insight into which characteristics of the tissue are most indicative for the diagnosis



BREAST CANCER WISCONSIN DATA SET

- Made available by UCI Machine Learning Repository
- 569 samples of tissue: 357 benign and 212 malignant sample
- Split the data 50/50 into training and testing data.

PREDICTORS

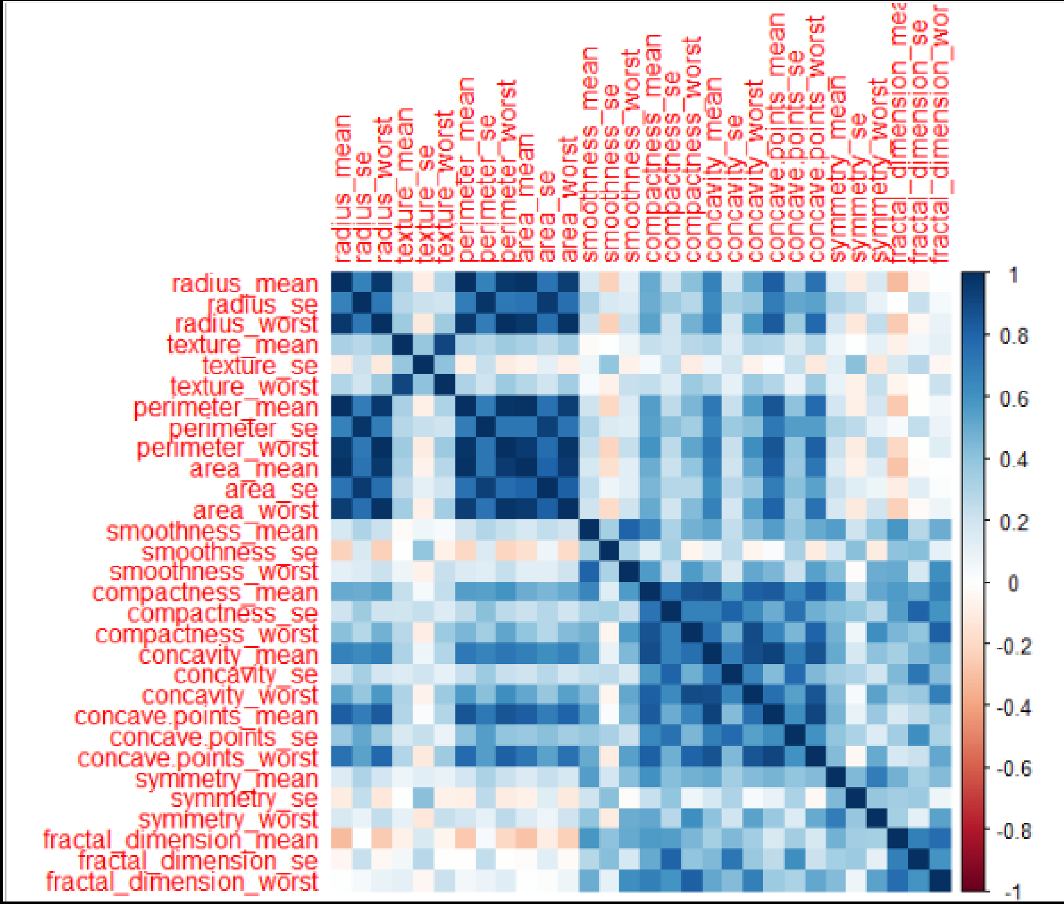
10 geometrical features of the cells:

radius, texture, perimeter, area, smoothness, compactness, concavity (severity of concave portions of the contour), concave points (number of concave portions of the contour), symmetry, fractal dimension.

For each of them: mean, standard error and “worst” value for the sample of the cells.

Hence 30 predictors from 10 groups in total.

PREDICTORS

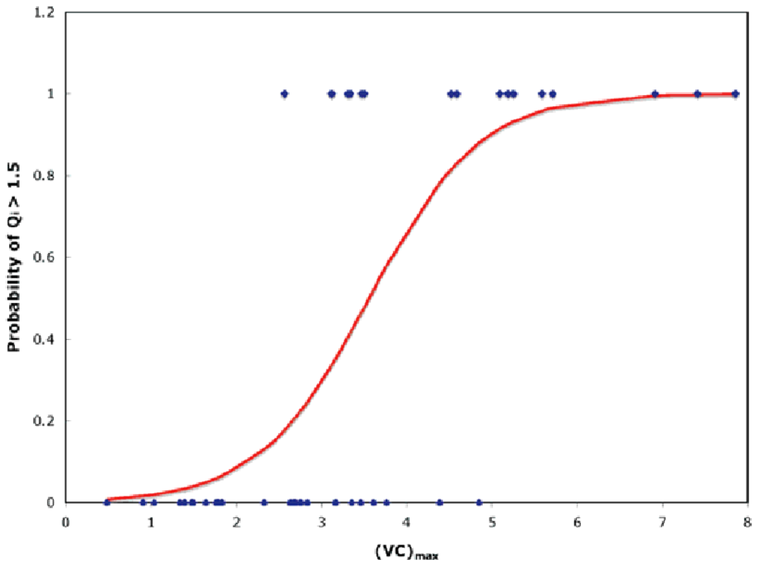


radius_mean
radius_se
radius_worst
texture_mean
texture_se
texture_worst
perimeter_mean
perimeter_se
perimeter_worst
area_mean
area_se
area_worst
smoothness_mean
smoothness_se
smoothness_worst
compactness_mean
compactness_se
compactness_worst
concavity_mean
concavity_se
concavity_worst
concave.points_mean
concave.points_se
concave.points_worst
symmetry_mean
symmetry_se
symmetry_worst
fractal_dimension_mean
fractal_dimension_se
fractal_dimension_worst
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1

LOGISTIC REGRESSION

- One of the simplest possible models
- Does not perform variable selection or shrinkage
- Still, it yields a pretty good result

Error rate on testing data: 0.053



LOGISTIC REGRESSION WITH ELASTIC NET PENALTY

Penalty of the form:

$$P_\lambda(\beta) = \lambda\left(\alpha\sum_{j=1}^{p}|\beta_j| + (1 - \alpha)\sum_{j=1}^{p}\beta_j^2\right)$$

α, λ chosen by cross validation.

- Performs variable selection.
- We expect it to perform better than the unpenalized logistic regression.

LOGISTIC REGRESSION WITH ELASTIC NET PENALTY

Results:

Selected variables: all of them except for texture mean, compactness mean, concavity standard error, concave points standard error, symmetry standard error, compactness worst.

Error rate on testing data: 0.025

Improvement in the predictive capability but still too many predictors in the model to have good interpretability.

LOGISTIC REGRESSION WITH GROUP LASSO PENALTY

Penalty of the form:

$$P_\lambda(\beta) = \lambda \sum_{g=1}^{G} \left\| \beta_{I_g} \right\|_2$$

where each of the covariates is grouped into one of G groups and β_{I_g} is a subvector of β which contains the coefficients corresponding to the covariates from the *g*-th group.

Potentially helpful when we have a natural grouping of the covariates, just like in our case (10 geometrical features of the cells).

LOGISTIC REGRESSION WITH GROUP LASSO PENALTY

Results:

Selected variables: all of them except for the ones from the group of perimeter.

Error rate on testing data: 0.055

Practically no improvement from unpenalized logistic regression both in terms of predictive capability and variable selection.

LOGISTIC REGRESSION WITH SPARSE GROUP LASSO PENALTY

Penalty of the form:

$$P_{\lambda}(\beta) = \lambda\left(\alpha \sum_{j=1}^{p} |\beta_j| + \ (1-\alpha) \sum_{g=1}^{G} \left\|\beta_{I_g}\right\|_2\right)$$

combination of the sparse group and lasso penalties.

LOGISTIC REGRESSION WITH SPARSE GROUP LASSO PENALTY

Selected covariates:

- radius (mean, standard error, worst)
- texture (mean, worst)
- smoothness (worst)
- concavity (mean, worst)
- concave points (mean, standard error, worst)
- symmetry (worst)

Error rate on testing data: 0.028

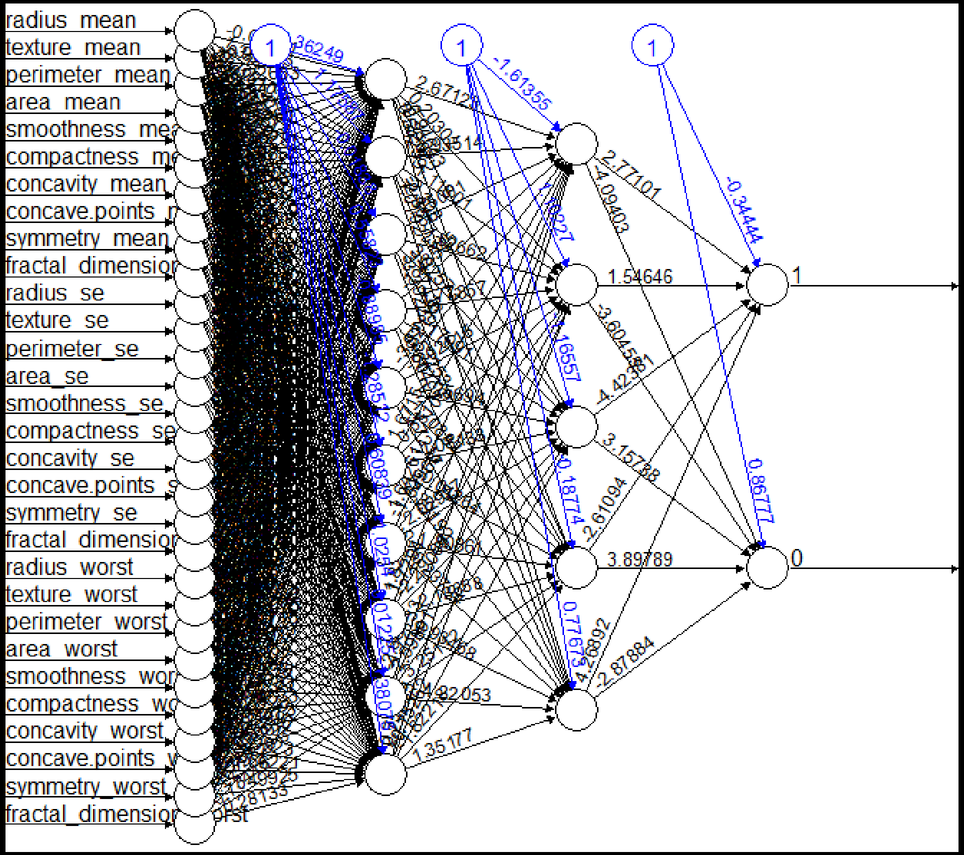
IMPROVING PREDICTION ACCURACY

ARTIFICIAL NEURAL NETWORK

Error function: **cross-entropy**

Optimization algorithm:
resilient backpropagation
with weight backtracking

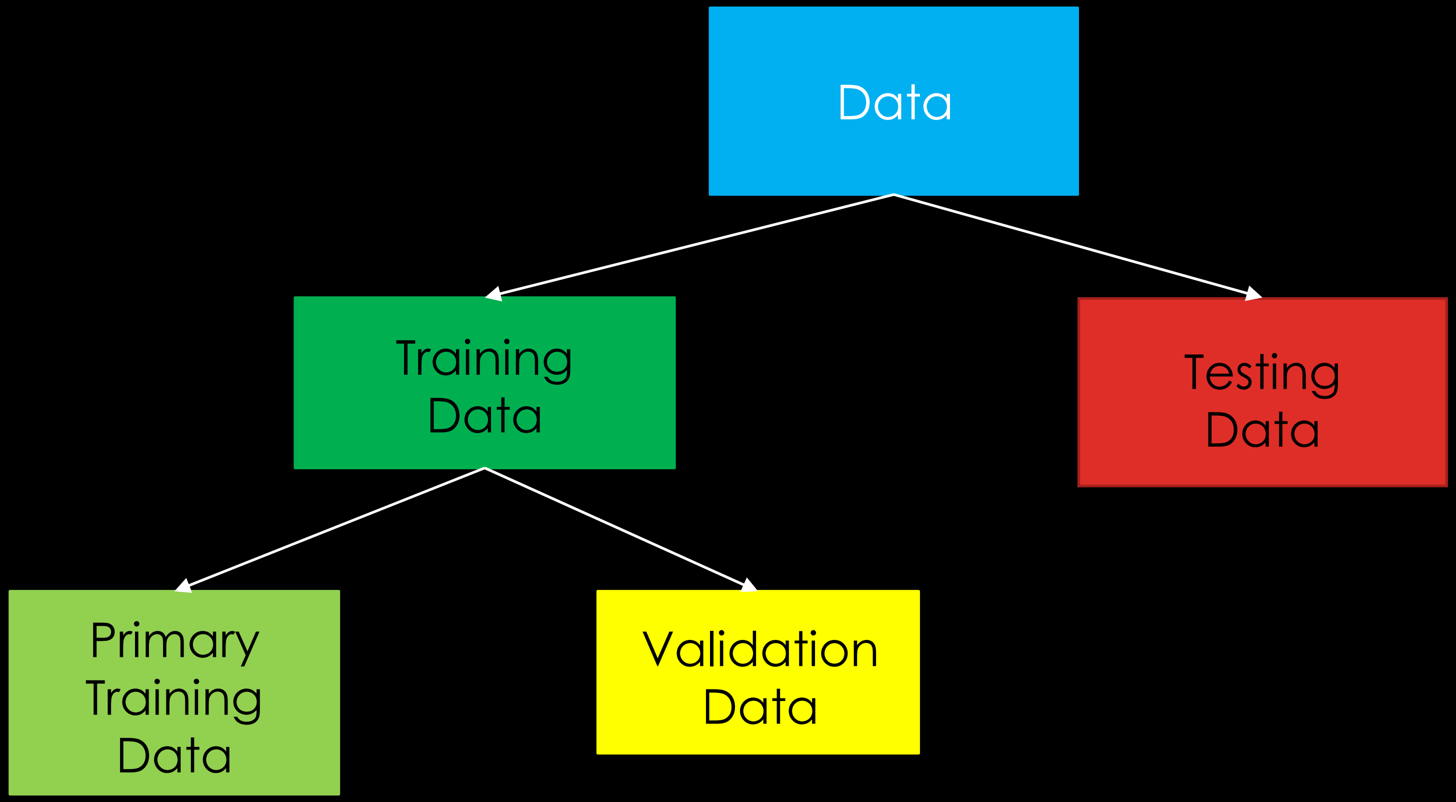
Error rate on testing data:
0.035



ENSEMBLE METHOD

- We will now attempt to combine the Neural Network with the Elastic Net Logistic Regression.
- We will lose the interpretability of the Logistic Regression but hopefully we can obtain better accuracy of prediction.
- We will use a particular version of Stacking Classifier.

STACKING CLASSIFIER



STACKING CLASSIFIER

Training:

- Train both classifiers on training data.
- Apply them to obtain the probabilities for the validation data.
- Train a meta-classifier on the probabilities for the validation data.

Evaluating:

- Apply the original classifiers to testing data to obtain probabilities.
- Make predictions by using the meta-classifier on these probabilities.

META-CLASSIFIERS AND THEIR PERFORMANCE

- Logistic Regression

  Error rate on testing data: 0.137

- LDA

  Error rate on testing data: 0.137

- QDA

  Error rate on testing data: 0.109

SUMMARY

Algorithm	Logistic Regression	Elastic Net Logistic Regression	Group Lasso Logistic Regression	Sparse Group Lasso Logistic Regression	Artificial Neural Network	Ensemble Method
Error rate	0.053	0.025	0.055	0.028	0.035	0.109

FURTHER INVESTIGATION

- Improving the ensemble method utilized – instead of fixing the validation portion of the data use cross-validation and average over the resulting models.
- Incorporating more algorithms into the analysis, for example random forest.
- Attempting to control the percentage of false negative diagnoses of a malignant tumor at a fixed level, lower than the one achieved now.
- Attempting to perform inference on some of the models. Unfortunately, using bootstrap did not give any conclusive results because of the nature of the penalties we used.

CONCLUSIONS

- The covariates seem to be very predictive of the diagnosis with the best classifier (the Elastic Net Logistic Regression) having its error rate at just 2.5%.

- Potentially, this number could be improved by applying better tuned ensemble methods.

REFERENCES

[1] Breast Cancer Wisconsin Data Set

https://www.kaggle.com/uciml/breast-cancer-wisconsin-data

[2] Package ‘msgl’, by Martin Vincent [aut], Niels Richard Hansen [ctb, cre]

https://cran.r-project.org/web/packages/msgl/msgl.pdf

[3] Stacking Classifier, by Bhavesh Bhatt

https://www.youtube.com/watch?v=sBrQnqwMpvA&list=LL&index=2

THANK YOU!