



Breast Cancer Predict

Marina Estévez Almenzar
Yuying Tan
Javier Morant



Universitat
Pompeu Fabra
Barcelona

Table of Contents

01

Introduction

Background
Hypothesis

02

Methodology

Dataset
Modeling

03

Result

Data exploration
Classifier

04

Conclusion

Conclusions
Our findings
Extensions





**2nd most
common cancer
in women**

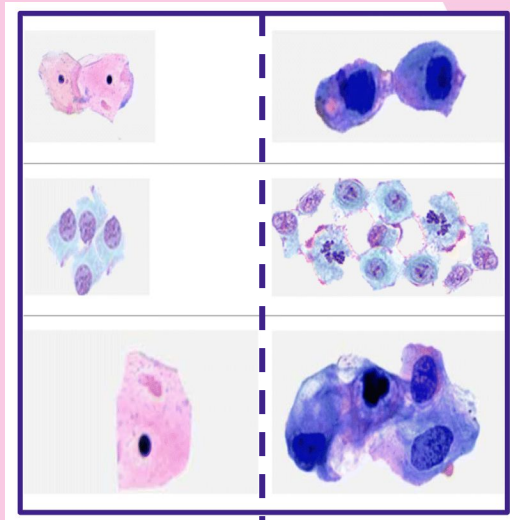
Introduction

- **Hypothesis:**

We hypothesize that Machine Learning algorithms can predict the nature of a breast tumour (benign or malignant) in order to provide health sector with assistance to breast tumour classification. Same methodology could be applied to other similar kind of cancer detection.



Fine needle aspiration



Benign


Malignant



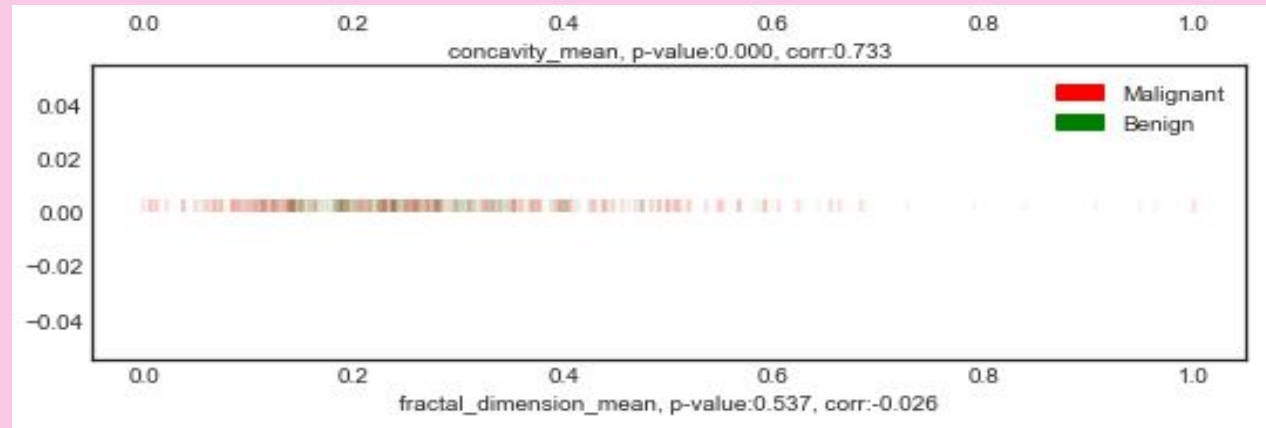
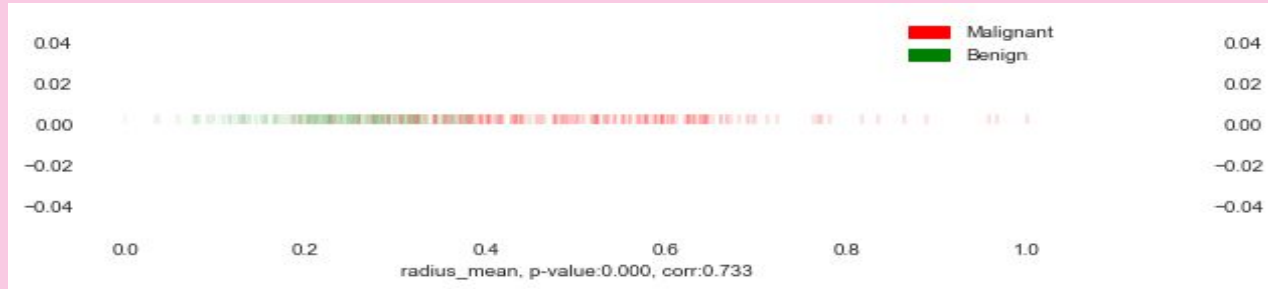
- Radius
- Texture
- Perimeter
- Area
- Smoothness
- Compactness
- Concavity
- Concave points
- Symmetry
- Fractal dimension

- Mean
- Standard error
- Worst/Largest

Data exploration

- 
- **Dataset:** Breast Cancer Wisconsin (Diagnostic) from Kaggle
 - **Classification & Supervised learning problem**
 - **Data formatting and exploration**
 - Normalize the dataset
 - 1) simple feature scaling
 - 2) min-max scaling
 - 3) Z-score scaling. We choose option 2)
 - **Spearman test to analyze each of the variables**

Data Exploration



- Fractal_dimension_mean
- Texture_se
- smoothness_se

Methodology

We run 5 different algorithms

- Support Vector Machine (SVM)
- Multilayer Perceptron (MLP)
- K Nearest Neighbour (KNN)
- Decision Tree (DT)
- Logistic Regression (LR)



Methodology

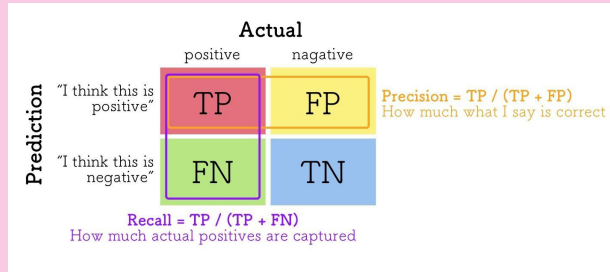
- Model evaluation

1) mean accuracy on cross validation

2) accuracy on test data

3) Jaccard index

4) F1-score



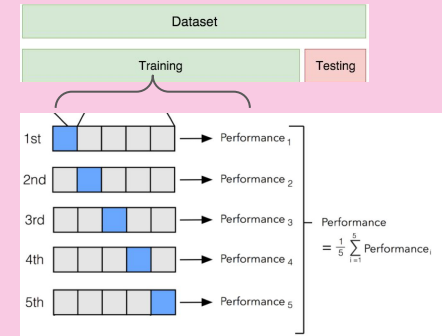
$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

High precision, low recall

TP	FP
FN	TN

Low precision, high recall

TP	FP
FN	TN



Jaccard Index

Jaccard (A, B) = $\frac{|A \cap B|}{|A \cup B|}$

A = y = actual labels
B = \hat{y} = predicted labels

$J(y, \hat{y}) = 1.0$

Higher Accuracy

Methodology

We run 5 different algorithms

- **Support Vector Machine (SVM)**
- Multilayer Perceptron (MLP)
- K Nearest Neighbour (KNN)
- Decision Tree (DT)
- Logistic Regression (LR)

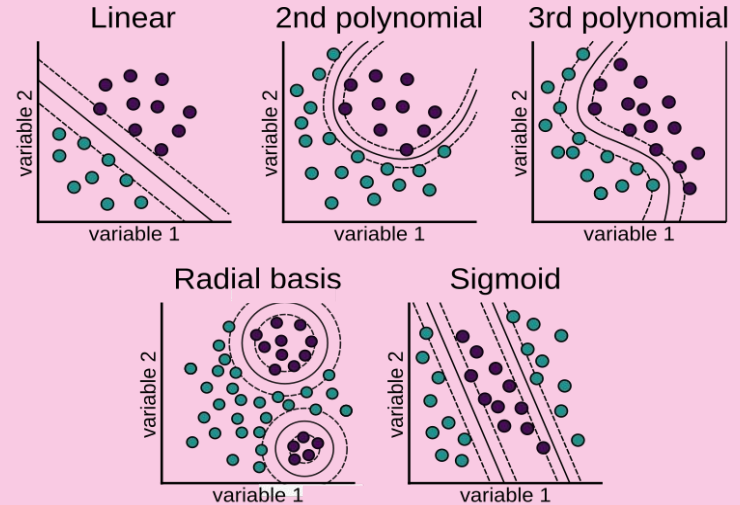
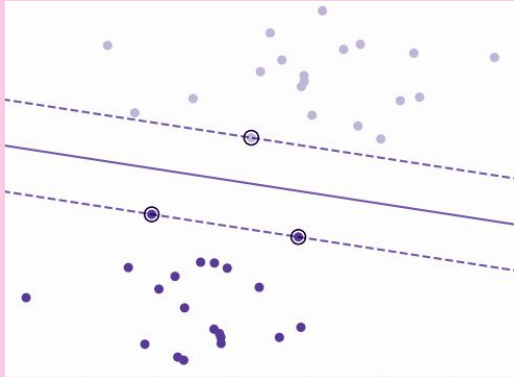


Methodology

SVM

A SVM constructs a hyperplane in a high dimensional space.

A good separation is achieved by the hyperplane when it gets the largest distance to the nearest training data points of any class.



The shape of the hyperplane is determined by the kernel function

Methodology

We run 5 different algorithms

- Support Vector Machine (SVM)
- **Multilayer Perceptron (MLP)**
- K Nearest Neighbour (KNN)
- Decision Tree (DT)
- Logistic Regression (LR)

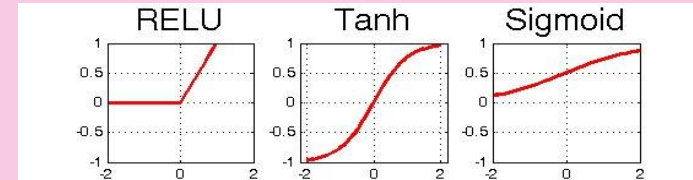
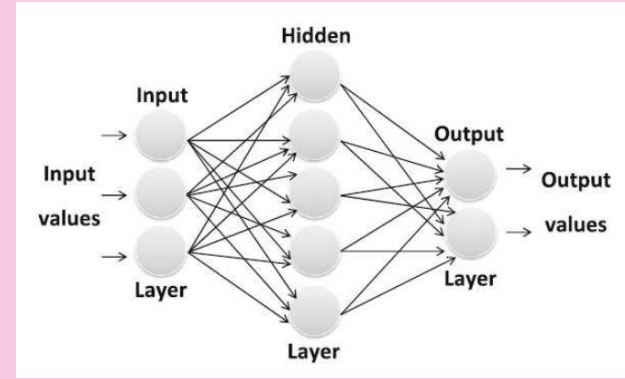


Methodology

MLP

Multilayer perceptron (MLP)

- Input layer, hidden layers, output layer
- Activation function
- Loss function
- Data is non linear separated
- Hyperparameters
- <https://www.youtube.com/watch?v=IHZwWFHwa-w>
- <https://mlfromscratch.com/optimizers-explained/>



Methodology

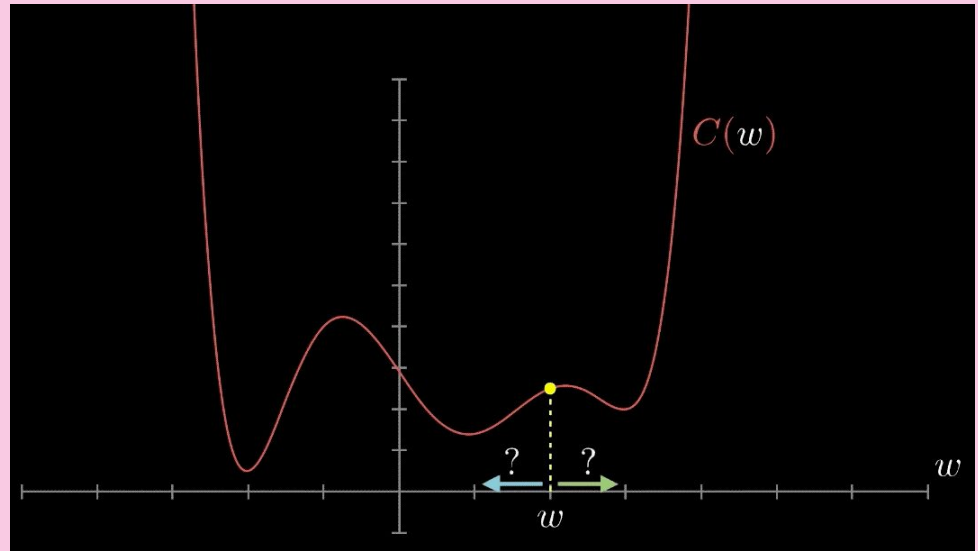
MLP

Comparison between parameters

Library: sklearn

```
MLPClassifier(hidden_layer_sizes=(100,),  
              max_iter=10000,  
              activation = 'tanh',  
              solver='adam',  
              random_state=1)  
0.9736842105263158
```

```
classifier = MLPClassifier(max_iter=10000,  
                           activation = 'relu',  
                           solver='sgd',  
                           learning_rate='adaptive',  
                           random_state=1)  
0.9298245614035088
```



Methodology

We run 5 different algorithms

- Support Vector Machine (SVM)
- Multilayer Perceptron (MLP)
- **K Nearest Neighbour (KNN)**
- **Decision Tree (DT)**
- Logistic Regression (LR)



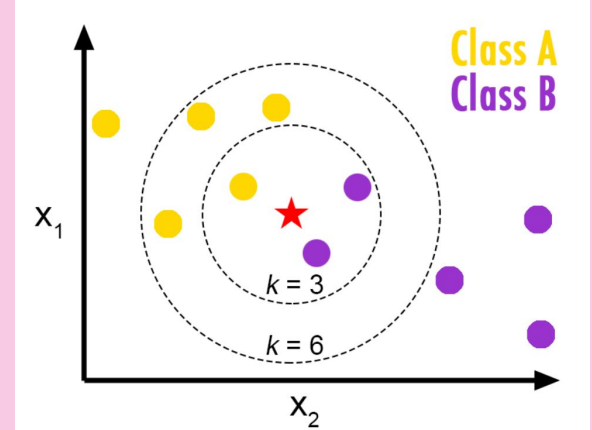
Methodology

KNN

k-Nearest Neighbour (KNN)

How to find the K?

Change the K-value from low to high values and keep track of all accuracy value.



Methodology

Finding the best K for KNN

► ML

```
Ks = 10  
jac = np.zeros(Ks)  
f1 = np.zeros(Ks)  
mcv = np.zeros(Ks)  
mt = np.zeros(Ks)
```

Try Ks value from 1 to 10

```
for n in range(Ks):  
    knn_model = KNeighborsClassifier(n_neighbors=n+1)  
    knn_model.fit(X_train,y_train)  
    yhat = knn_model.predict(X_test)  
    scores = cross_val_score(knn_model, X_train, y_train, cv=5)  
    mcv[n] = scores.mean()  
    score = knn_model.score(X_test, y_test)  
    mt[n] = score  
    jac[n] = jaccard_score(y_test, yhat)  
    f1[n] = f1_score(y_test, yhat, average='weighted')
```

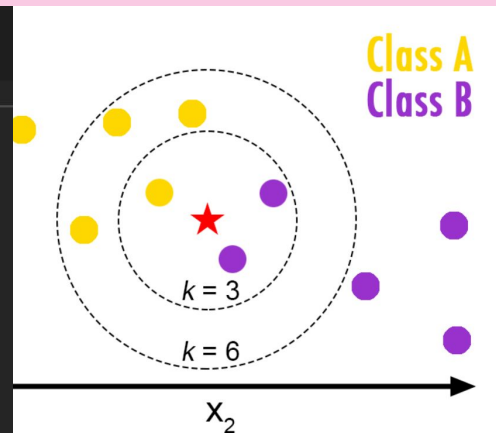
Calculate accuracy of the model on each Ks value

```
print("Mean accuracy on test data reaches the highest value for", int(np.where(mt == max(mt))[0])+1, "neighbors")  
print("Mean accuracy on cv reaches the highest value for", int(np.where(mcv == max(mcv))[0])+1, "neighbors")  
print("Jaccard index reaches the highest value for", int(np.where(jac == max(jac))[0])+1, "neighbors")  
print("F1-score reaches the highest value for", int(np.where(f1 == max(f1))[0])+1, "neighbors")
```

```
Mean accuracy on test data reaches the highest value for 3 neighbors  
Mean accuracy on cv reaches the highest value for 10 neighbors  
Jaccard index reaches the highest value for 3 neighbors  
F1-score reaches the highest value for 3 neighbors
```

Best Ks=3

Find the K for KNN

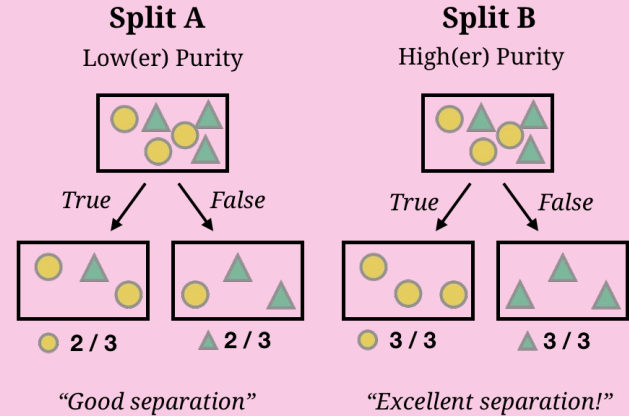


Methodology

DT

Decision Tree (DT)

Select split condition that maximize gain of purity.
Locally optimal decisions at each node cannot
guarantee return the globally optimal decision tree.



Methodology

We run 5 different algorithms

- Support Vector Machine (SVM)
- Multilayer Perceptron (MLP)
- K Nearest Neighbour (KNN)
- Decision Tree (DT)
- **Logistic Regression (LR)**



Methodology

LR

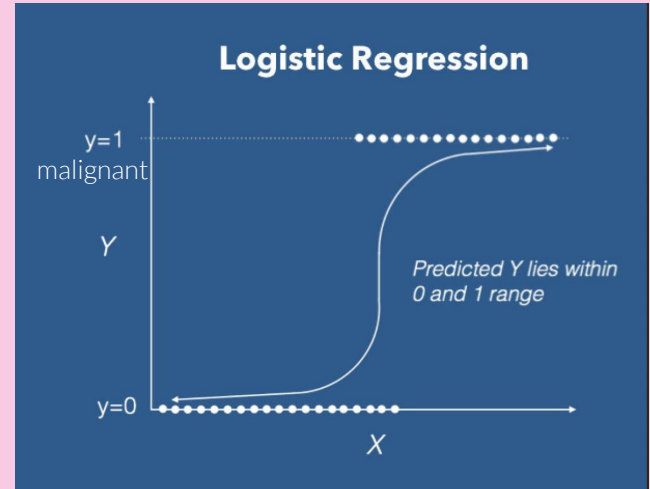
Logistic regression (LR)

Predict binary classes.

S-shaped curve and map all values into between 0 and 1.

Computes the probability.

Log loss evaluate how good are the predicted probabilities.



“what is the probability of the point being malignant”?

Conclusions

Algorithm	Mean accuracy on cross-validation	Accuracy on the test data	Jaccard score	F1-score
SVM linear kernel	0.97 (+/- 0.04)	0.96	0.90	0.96
SVM polynomial kernel (2)	0.98 (+/- 0.04)	0.97	0.93	0.97
SVM polynomial kernel (3)	0.98 (+/- 0.04)	0.97	0.93	0.97
SVM polynomial kernel (4)	0.97 (+/- 0.05)	0.96	0.88	0.96
SVM polynomial kernel (5)	0.95 (+/- 0.03)	0.96	0.89	0.96
SVM RFB kernel	0.97 (+/- 0.04)	0.97	0.93	0.97
SVM sigmoid kernel	0.32 (+/- 0.11)	0.29	0.01	0.29
MLP with adam solver	0.97 (+/- 0.05)	0.97	0.93	0.97
MLP with sgd solver	0.96 (+/- 0.04)	0.93	0.82	0.93
K Nearest Neighbors	0.96 (+/- 0.02)	0.96	0.91	0.96
Decision Tree	0.91 (+/- 0.07)	0.95	0.86	0.95
Logistic Regression	0.96 (+/- 0.03)	0.95	0.86	0.95

Features selection

- **SelectKBest.** Select features according to the K highest scores. It is based on the *f_classif* function, that computes the ANOVA-F value for our samples.
- **Recursive feature elimination.** Given an external estimator that assigns weights to features, recursive feature elimination is to select features by recursively considering smaller and smaller sets of features. It is performed in a cross-validation loop to find the optimal number of features.
 - External estimator $\in \{ \text{SVM with linear kernel, Decision Tree} \}$

Less relevant features:

- area_se
- texture_se
- smoothness_se

Extensions

- **Obtain more data.** By using more data we could get to more accurate and useful classifier.
- **Keep execution times in mind.** In case we were able to access more data, we could consider to incorporate execution times to our analysis.
- **Fit hyperparameters.** As done with the search of the optimal number of neighbors for KNN, we could fit other hyperparameters such as the regularization parameter C for SVM, the number of hidden layers for NN, etc.
- **Use selected features.** We could modify the initial data and keep just the selected features, then repeat these executions and see whether the results obtained are more accurate.
- *Added after the presentation (thanks Giovanni and Riccardo!)* – We can consider more **real life factors rather than only accuracy**, especially in clinical, medical context when conclude which are the methodological details we choose to solve this kind of problem.

Thank you for your attention!

