# Breast Cancer Prediction

IST 718 Final Project Report

Group 1

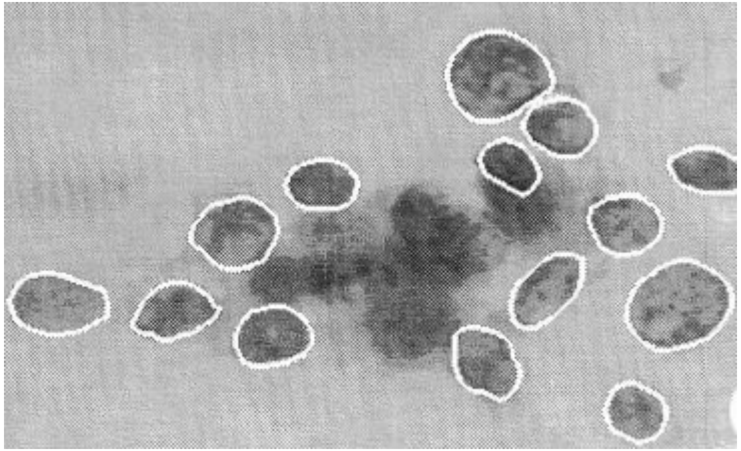
Mike DeMaria, Lu Guo, Haotian Shen, Casey Walsh

# Introduction

When a tumor is detected, it needs to be assessed if it’s benign or malignant.

Nowadays, Machine Learning has been widely used to help detect tumors.

We want to know that given the output of the image analysis, can we predict malignancy?



# Data Exploration

We obtain breast cancer data from Kaggle:
https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data/data

569 rows

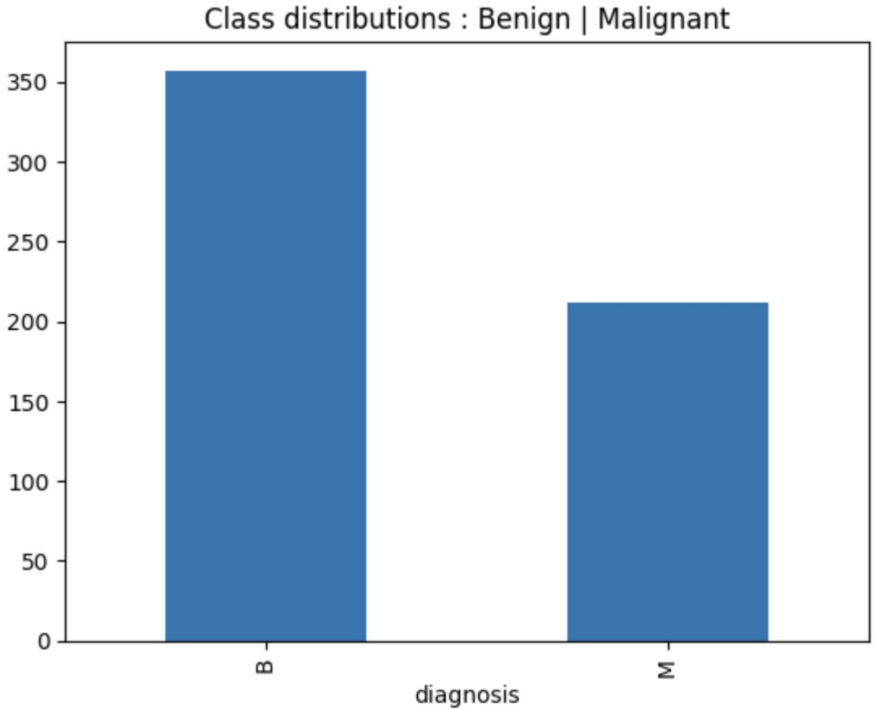
32 columns:

- patient ID
- 1 binary output variable (benign or malignant)
- 30 numerical variables of cancer cells

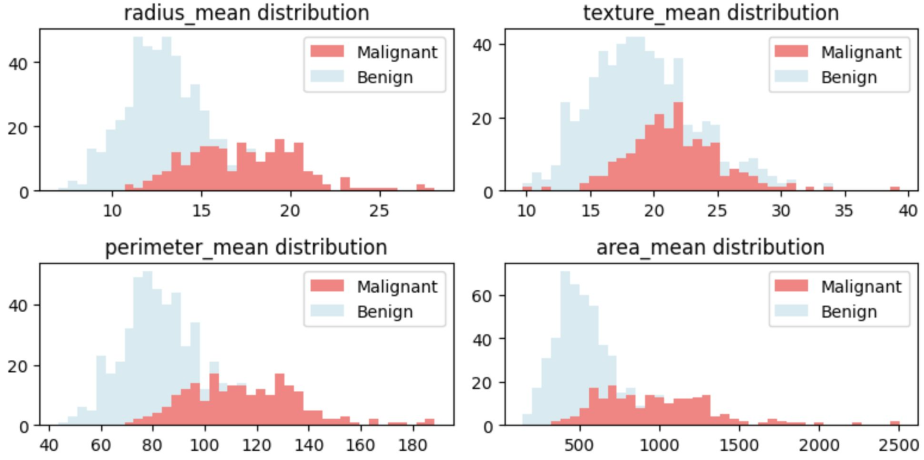
  10 categories: radius, texture, perimeter, area, smoothness, compactness, concavity, concave points, symmetry, and fractal dimension.

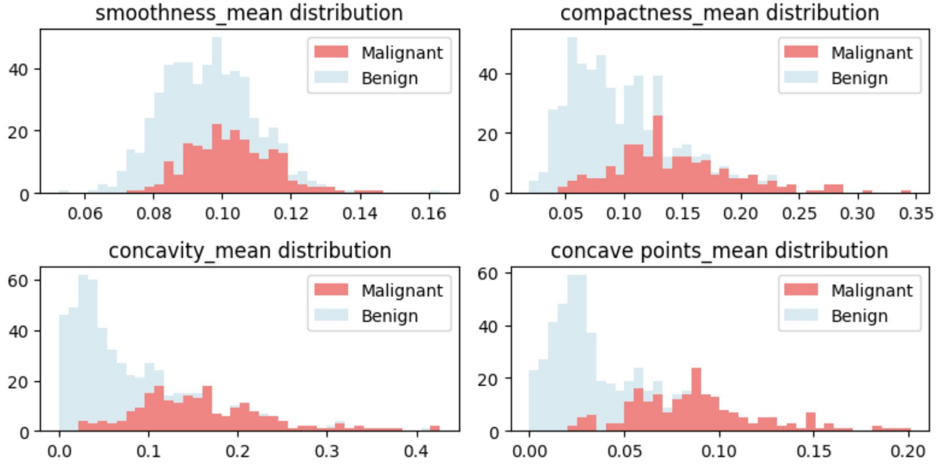
  3 values for each feature: the **mean**, the **standard error**, and the mean of the three **worst** (largest) cells.

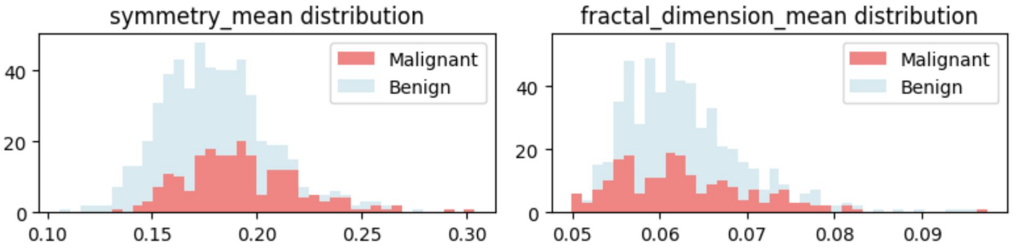
# Data Exploration

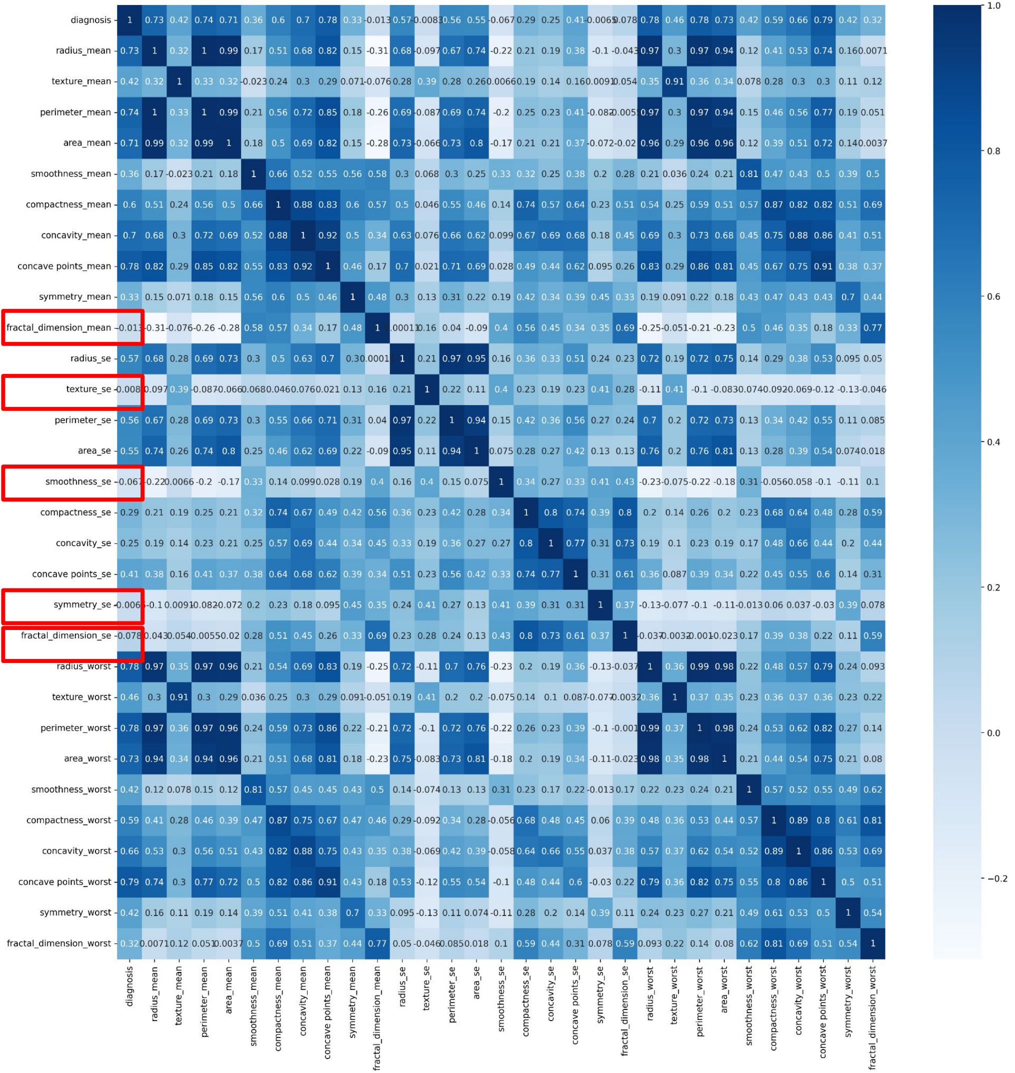


# Data Exploration









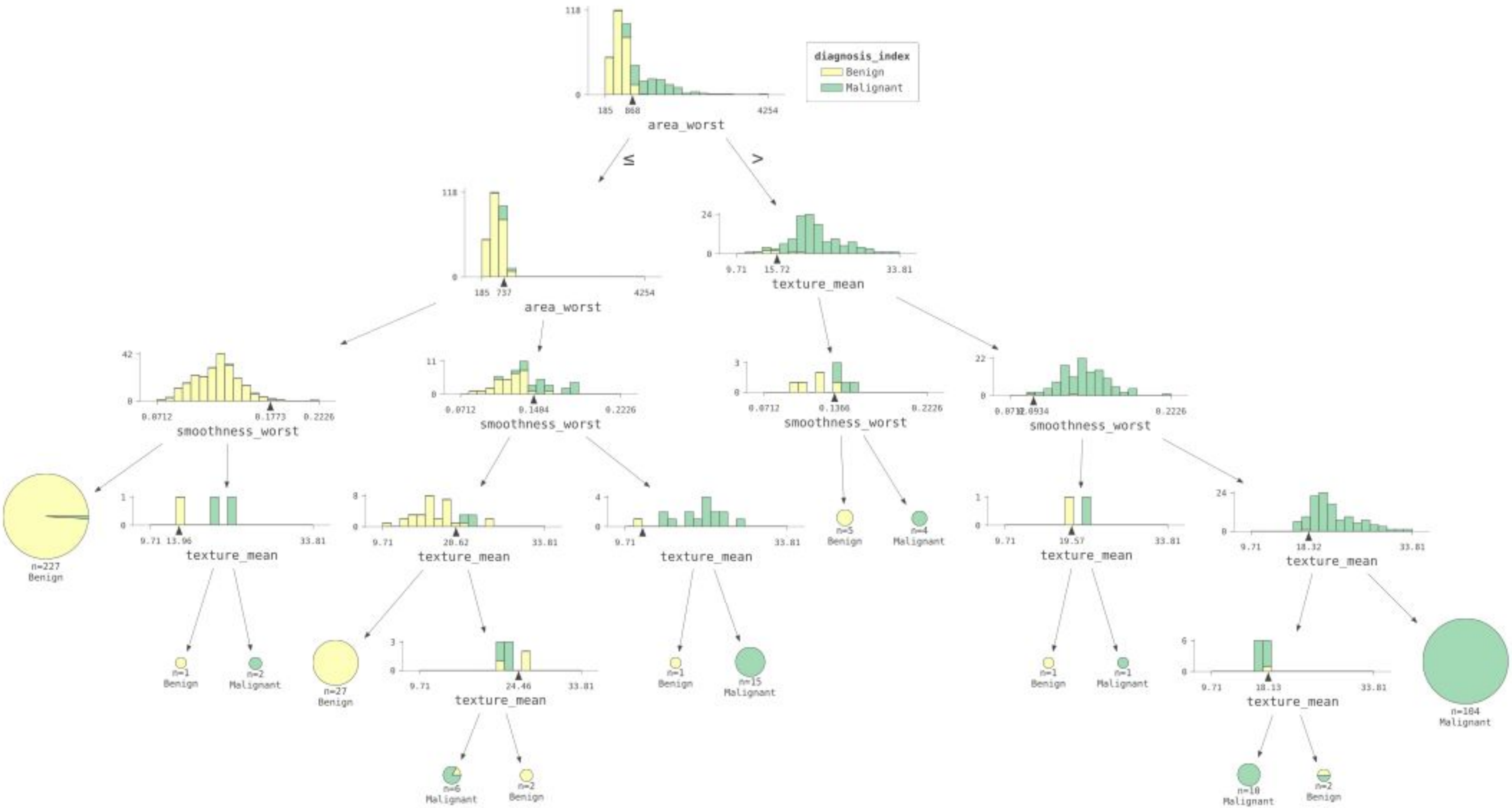
The below columns have very low correlation to diagnosis:

- fractal_dimension_mean,
- texture_se,
- smoothness_se,
- symmetry_se,
- fractal_dimension_se.

When doing feature engineering, maybe we can try to remove these columns.

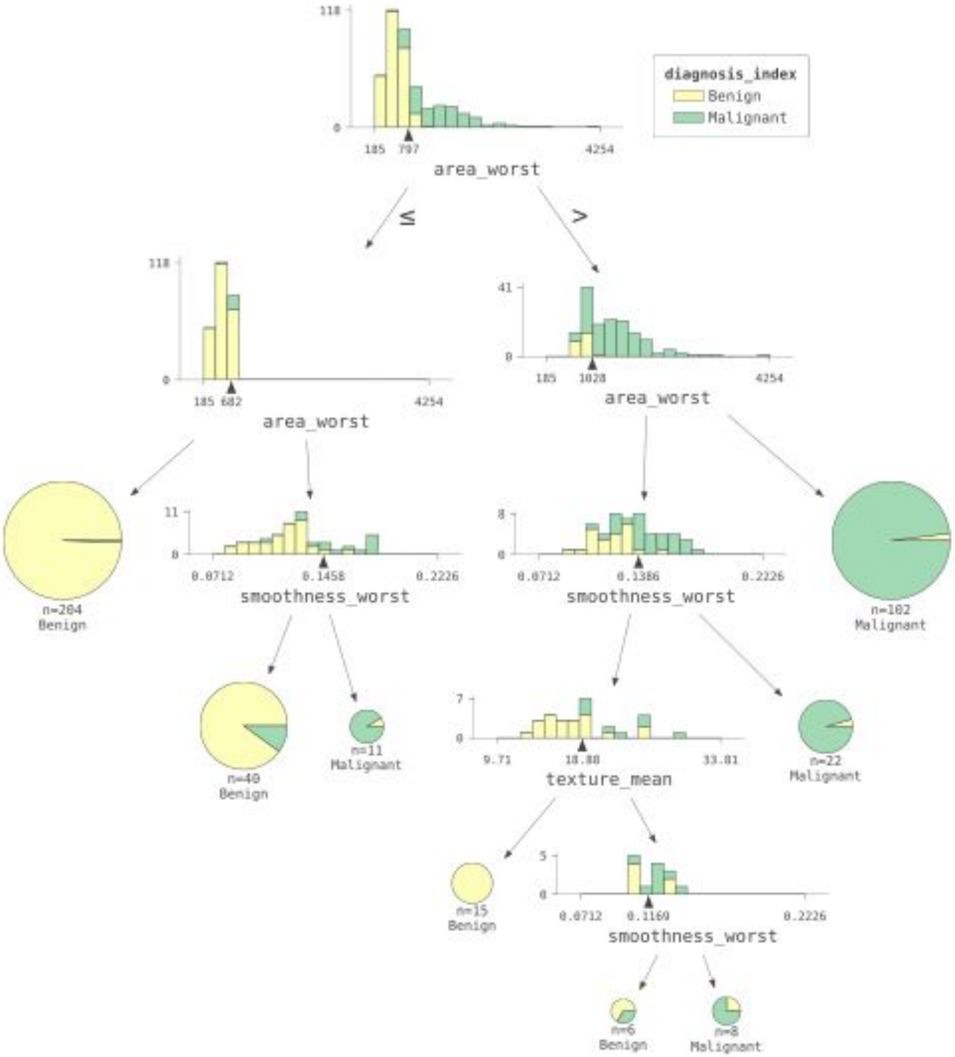
# Original Model

- Decision Tree Model
- Used 3 of the thirty available features
    - Mean Texture
    - Worst Area
    - Worst Smoothness
- Small subsets chosen for compute speed
- Original Model achieved 97% accuracy
- Attempt to replicate this model achieved 92% accuracy
    - Depth 6
    - 28 nodes


diagnosis_index
🟨 Benign
🟩 Malignant
118
0
185
868
4254
area_worst
≤
>
118
0
185
737
4254
area_worst
24
0
9.71
15.72
33.81
texture_mean
42
0
0.0712
0.1773
0.2226
smoothness_worst
11
0
0.0712
0.1404
0.2226
smoothness_worst
3
0
0.0712
0.1366
0.2226
smoothness_worst
22
0
0.0712
0.0934
0.2226
smoothness_worst
n=227
Benign
1
0
9.71
13.96
33.81
texture_mean
8
0
9.71
20.62
33.81
texture_mean
4
0
9.71
33.81
texture_mean
n=5
Benign
n=4
Malignant
1
0
9.71
19.57
33.81
texture_mean
24
0
9.71
18.32
33.81
texture_mean
n=1
Benign
n=2
Malignant
n=27
Benign
3
0
9.71
24.46
33.81
texture_mean
n=1
Benign
n=15
Malignant
n=1
Benign
n=1
Malignant
6
0
9.71
18.13
33.81
texture_mean
n=104
Malignant
n=6
Malignant
n=2
Benign
n=10
Malignant
n=2
Benign

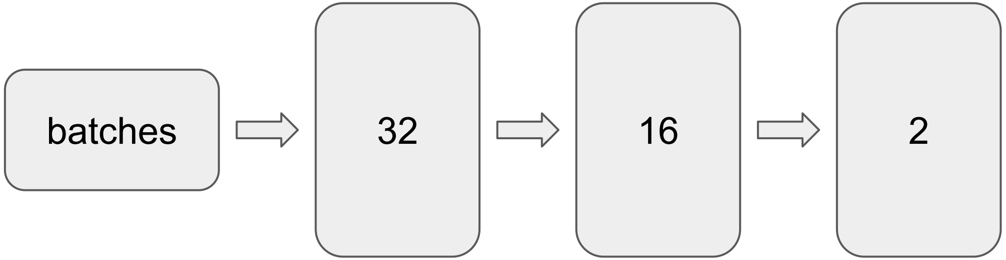
# Recreating the Original Model

- Decision Tree Model trained with the same three features
  - Trained in under 10 seconds
- Initial accuracy of 92%
- Retrained model with three features and hyperparameter grid search
  - Trained 120 models in approximately 10 minutes
  - Best model parameters:
    - Depth 5
    - 15 nodes
- Final accuracy of 96%

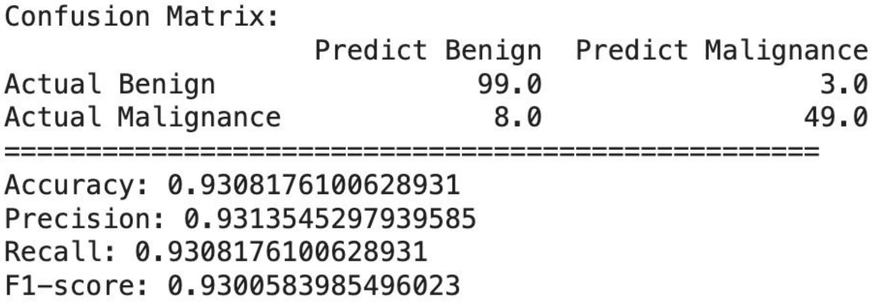


118
0
185
797
4254
area_worst
≤
>
diagnosis_index
Benign
Malignant
118
0
185
682
4254
area_worst
41
0
185
1028
4254
area_worst
n=204
Benign
11
0
0.0712
0.1458
0.2226
smoothness_worst
8
0
0.0712
0.1386
0.2226
smoothness_worst
n=102
Malignant
n=40
Benign
n=11
Malignant
7
0
9.71
18.88
33.81
texture_mean
n=22
Malignant
n=15
Benign
5
0
0.0712
0.1169
0.2226
smoothness_worst
n=6
Benign
n=8
Malignant

# Neural Network

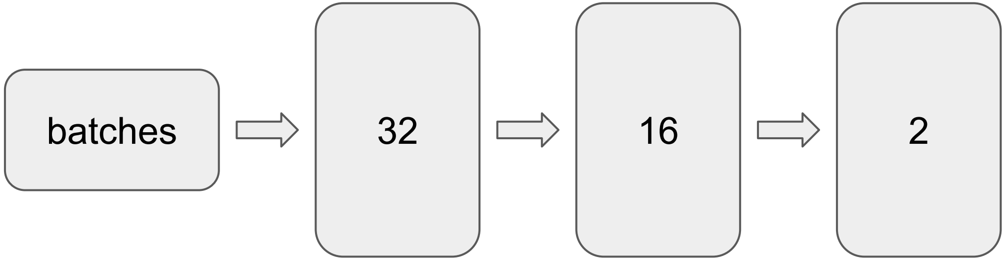


Select all mean features



```
Confusion Matrix:
                   Predict Benign  Predict Malignance
Actual Benign                99.0                 3.0
Actual Malignance             8.0                49.0
==================================================
Accuracy: 0.9308176100628931
Precision: 0.9313545297939585
Recall: 0.9308176100628931
F1-score: 0.9300583985496023
```

# Neural Network



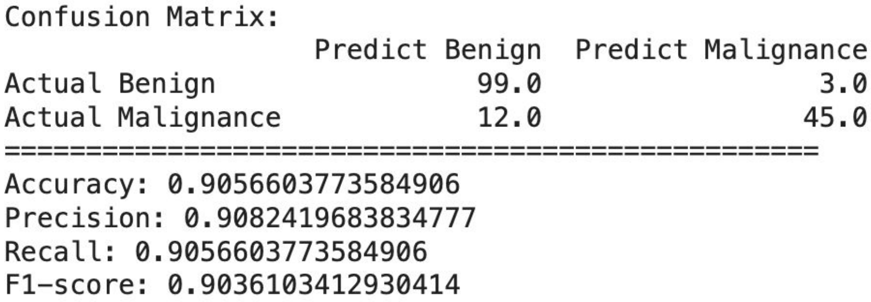
batches

32

16

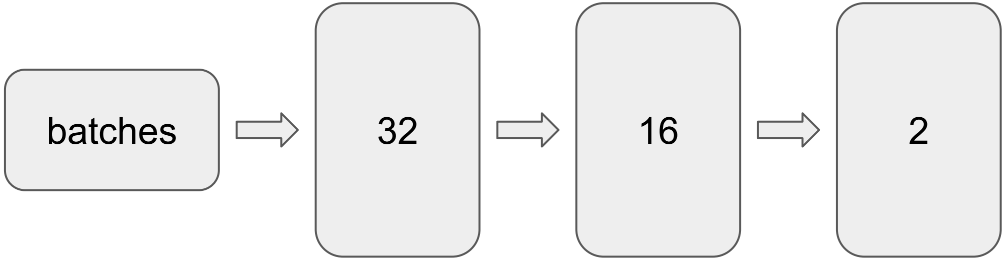
2

Select all standard error features



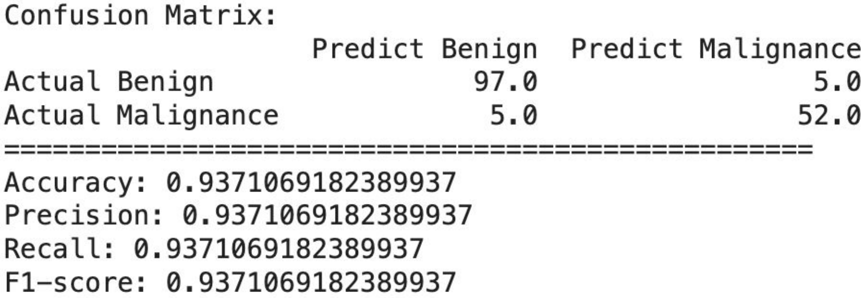
```
Confusion Matrix:
                   Predict Benign  Predict Malignance
Actual Benign                99.0                 3.0
Actual Malignance            12.0                45.0
==================================================
Accuracy: 0.9056603773584906
Precision: 0.9082419683834777
Recall: 0.9056603773584906
F1-score: 0.9036103412930414
```

# Neural Network



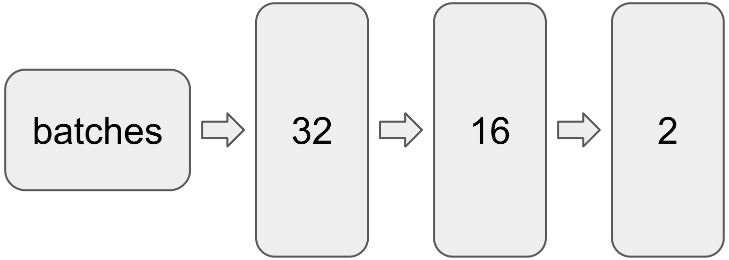
batches → 32 → 16 → 2

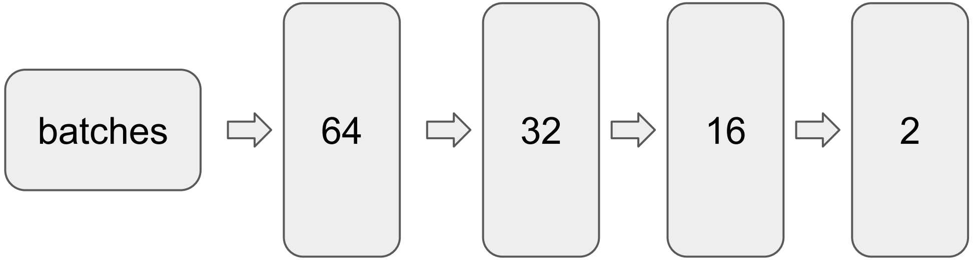
Select all worst features

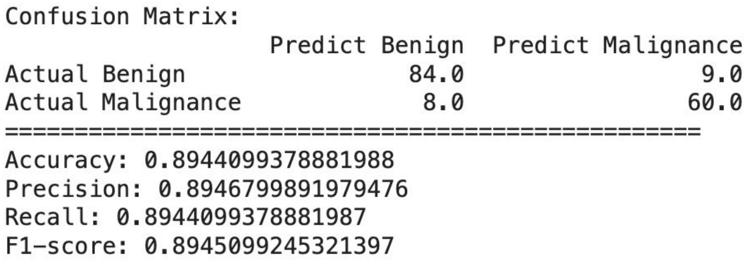


```
Confusion Matrix:
                     Predict Benign  Predict Malignance
Actual Benign                  97.0                 5.0
Actual Malignance               5.0                52.0
==================================================
Accuracy: 0.9371069182389937
Precision: 0.9371069182389937
Recall: 0.9371069182389937
F1-score: 0.9371069182389937
```

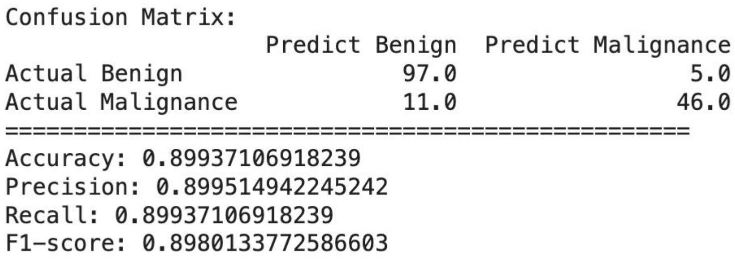
# Neural Network







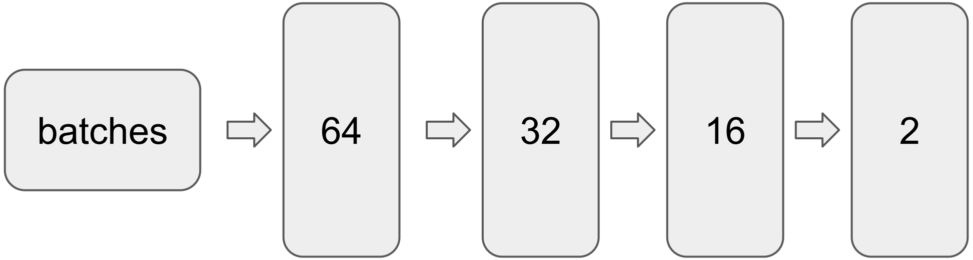
```
Confusion Matrix:
                   Predict Benign  Predict Malignance
Actual Benign                84.0                 9.0
Actual Malignance             8.0                60.0
==================================================
Accuracy: 0.8944099378881988
Precision: 0.8946799891979476
Recall: 0.8944099378881987
F1-score: 0.8945099245321397
```



```
Confusion Matrix:
                   Predict Benign  Predict Malignance
Actual Benign                97.0                 5.0
Actual Malignance            11.0                46.0
=================================================
Accuracy: 0.89937106918239
Precision: 0.899514942245242
Recall: 0.89937106918239
F1-score: 0.8980133772586603
```

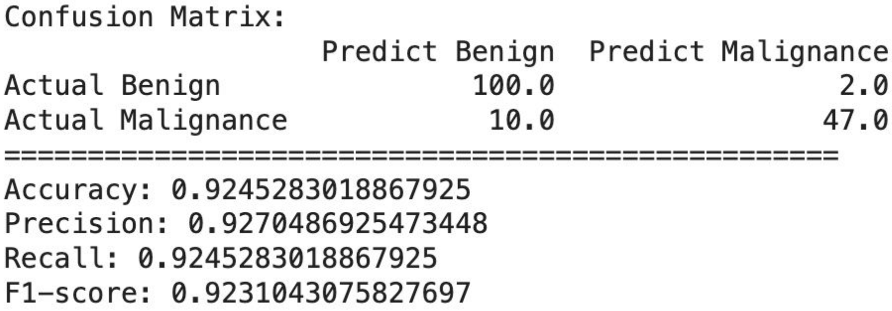
Select all features

# Neural Network



batches → 64 → 32 → 16 → 2

Drop
'fractal_dimension_mean',
'texture_se', 'smoothness_se',
'symmetry_se',
'fractal_dimension_se'



```
Confusion Matrix:
                   Predict Benign  Predict Malignance
Actual Benign               100.0                 2.0
Actual Malignance            10.0                47.0
=====================================================
Accuracy: 0.9245283018867925
Precision: 0.9270486925473448
Recall: 0.9245283018867925
F1-score: 0.9231043075827697
```

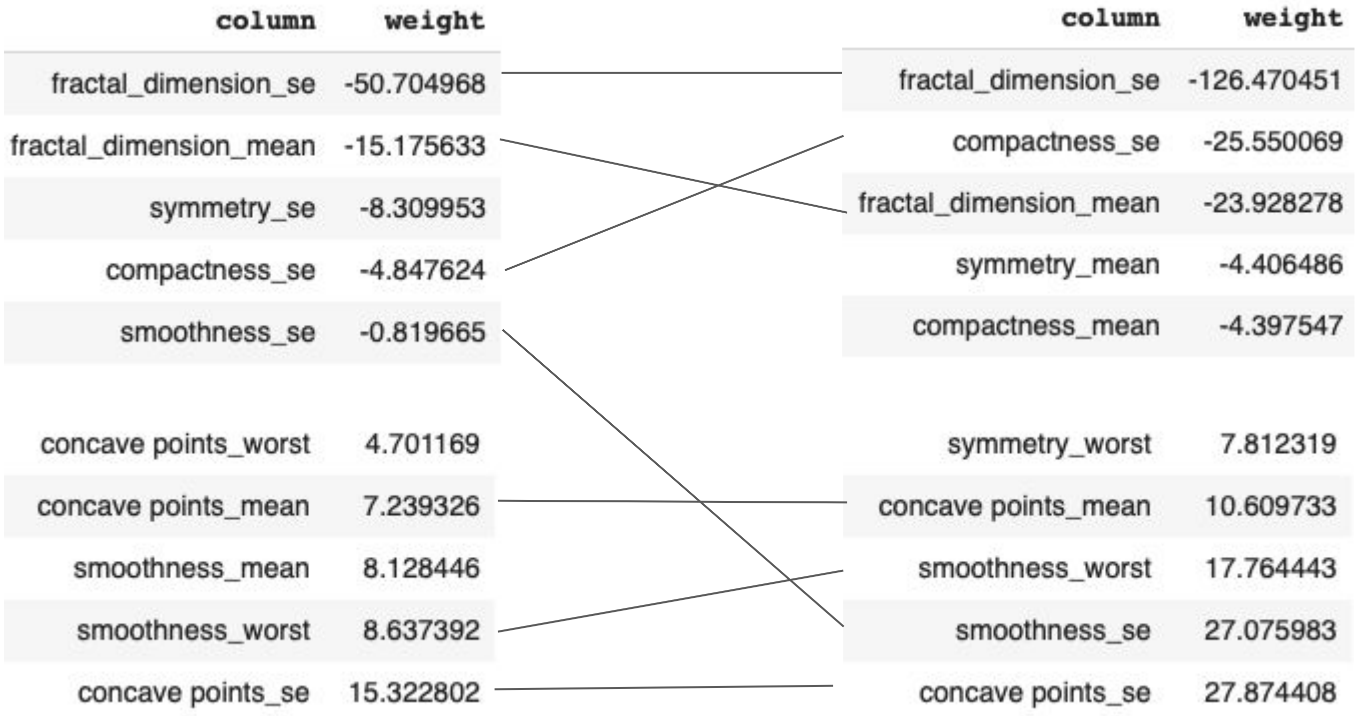
# Neural Network

| | accuracy | precision | recall | f1-score |
|---|---|---|---|---|
| All mean features | 0.9308 | 0.9314 | 0.9308 | 0.9301 |
| All standard error features | 0.9057 | 0.9082 | 0.9057 | 0.9036 |
| **All worst features** | **0.9371** | **0.9371** | **0.9371** | **0.9371** |
| All features | 0.8944 | 0.8947 | 0.8944 | 0.8945 |
| All features(large) | 0.8994 | 0.8995 | 0.8994 | 0.8980 |
| All features+Drop(large) | 0.9245 | 0.9270 | 0.9245 | 0.9231 |

# Additional Models

| Algorithm | Data Used | AUC | Runtime (seconds) |
|---|---|---|---|
| Logistic Regression | “Worst” Cells | 95.5% | 2.1 |
| Random Forest | All | 97.9% | 6.2 |
| Logistic Regression (Increased iterations) | Mean/Std Dev | 98.3% | 4.8 |
| GBT | All | 98.9% | 11.8 |
| Logistic Regression | Mean/Std Dev | 99.1% | 2.14 |
| LinearSVC | All | 99.6% | 28.3 |
| Logistic Regression | All | 99.7% | 4.2 |

# Best Models Bottom/Top Coefficients



## Logistic Regression

| column | weight |
|---|---|
| fractal_dimension_se | -50.704968 |
| fractal_dimension_mean | -15.175633 |
| symmetry_se | -8.309953 |
| compactness_se | -4.847624 |
| smoothness_se | -0.819665 |
| concave points_worst | 4.701169 |
| concave points_mean | 7.239326 |
| smoothness_mean | 8.128446 |
| smoothness_worst | 8.637392 |
| concave points_se | 15.322802 |

## LinearSVC

| column | weight |
|---|---|
| fractal_dimension_se | -126.470451 |
| compactness_se | -25.550069 |
| fractal_dimension_mean | -23.928278 |
| symmetry_mean | -4.406486 |
| compactness_mean | -4.397547 |
| symmetry_worst | 7.812319 |
| concave points_mean | 10.609733 |
| smoothness_worst | 17.764443 |
| smoothness_se | 27.075983 |
| concave points_se | 27.874408 |

# Best Models Quality

| | Linear Regression | LinearSVC |
|---|---|---|
| Runtime | 4.2 seconds | 28.3 seconds |
| AUC | 99.7% | 99.6% |
| Accuracy | 98% | 98% |
| Precision | 96% | 98% |
| Recall | 100% | 99% |
| F-measure | 98% | 99% |
| False Positives | 4 | 2 |
| False Negatives | 0 | 1 |

# Thanks!