BREAST CANCER PREDICTION USING LOGISTIC REGRESSION

BDA-2 Mini Project

PB 17- Indira Pimpalkhare- 1032170431
PB 23- Priya Bannur- 1032170692
PB 34- Khushboo Agarwal- 1032170829

Overview

1. **Dataset:** UCI Wisconsin Breast Cancer

2. **Database driver:** PyMongo

3. **Machine Learning:** Logistic Regression

4. **Big data method:** Pyspark

5. **Visualization:** Tableau

Dataset

- Name: Breast Cancer Wisconsin (Diagnostic) Data Set
- About the dataset:
  - ID number (1)
  - Diagnosis (M = malignant, B = benign) (2)
  - (3-32) Attributes - Ten real-valued features are computed for each cell nucleus:
- Total 30 features.
- Class distribution: 357 benign, 212 malignant

Attributes:

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)

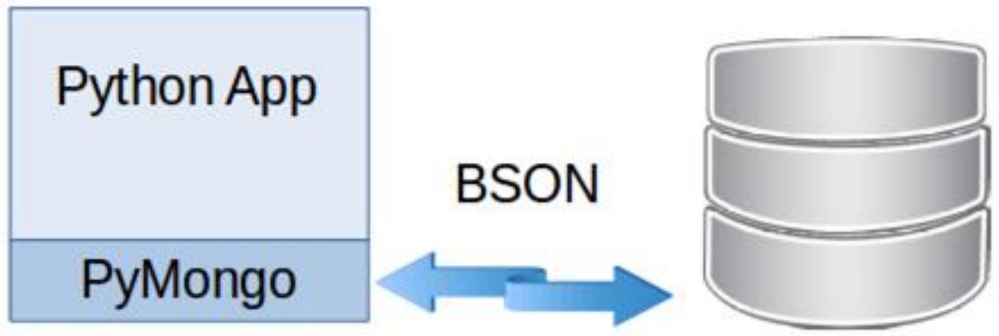
g) **concavity** (severity of concave portions of the contour)

h) **concave points** (number of concave portions of the contour)

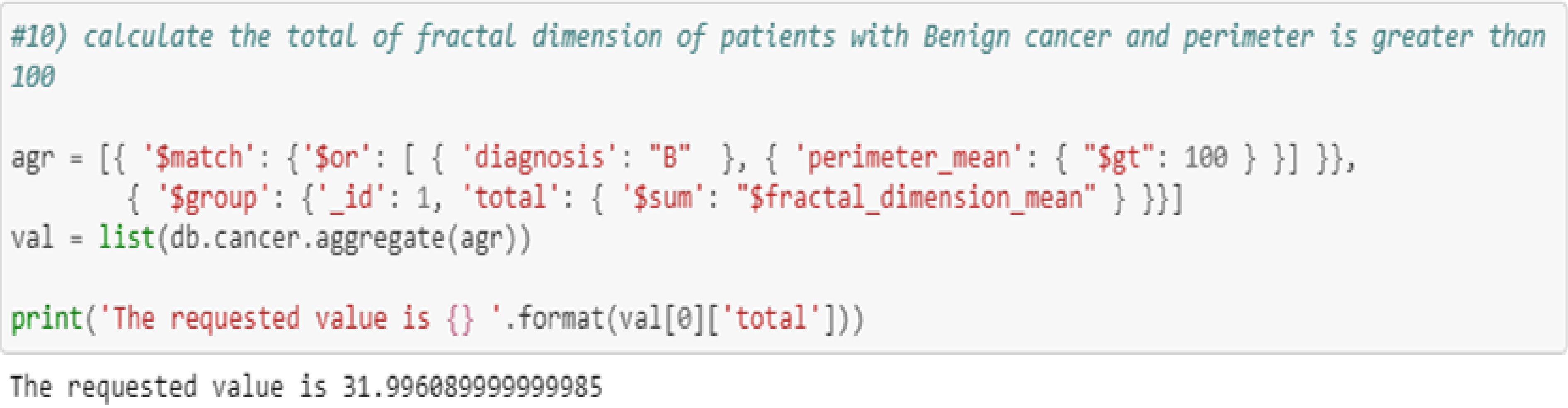
i) **symmetry**

j) **fractal dimension** ("coastline approximation" - 1)

PyMongo



A MongoDB driver for Python to access the MongoDB database



```
#10) calculate the total of fractal dimension of patients with Benign cancer and perimeter is greater than
100

agr = [{ '$match': {'$or': [ { 'diagnosis': "B"  }, { 'perimeter_mean': { "$gt": 100 } }] }},
       { '$group': {'_id': 1, 'total': { '$sum': "$fractal_dimension_mean" } }}]
val = list(db.cancer.aggregate(agr))

print('The requested value is {} '.format(val[0]['total']))
```

```
The requested value is 31.996089999999985
```

Logistic Regression

- Logistic Regression is commonly used to estimate the probability that an instance belongs to a particular class.

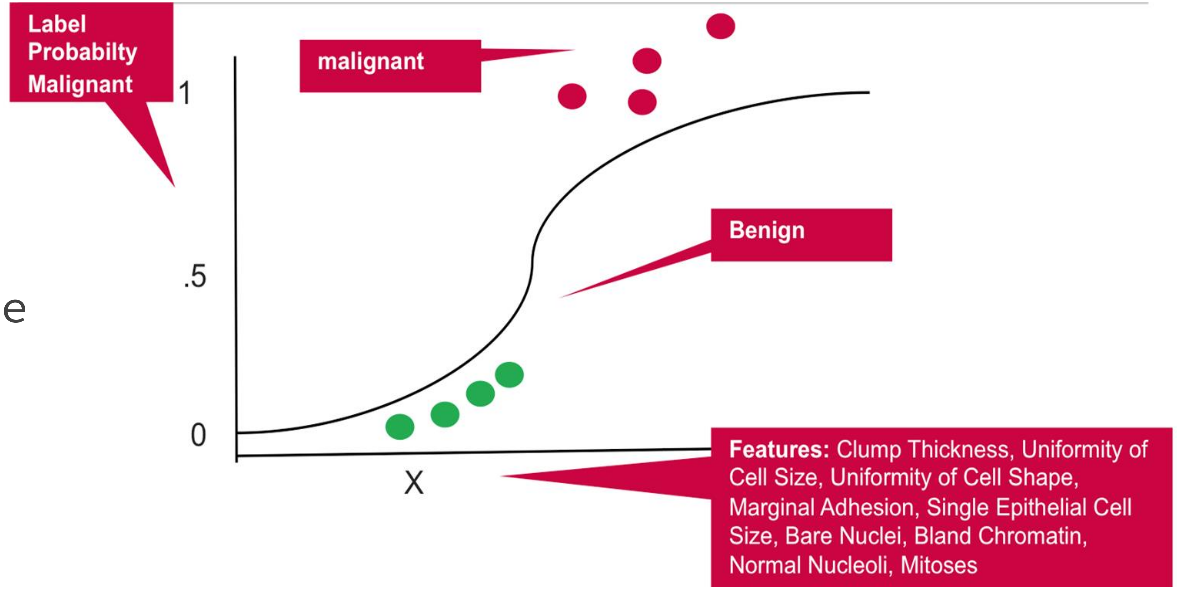
- In our case, what is the probability that this tumor is malignant or benign?

- **Binary Classifier**

Estimated Probability -

> **50%** Positive Class (1 / Malignant)

< **50%** Negative Class (0 / Benign)



Label Probabilty Malignant

malignant

Benign

Features: Clump Thickness, Uniformity of Cell Size, Uniformity of Cell Shape, Marginal Adhesion, Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses

Hadoop vs Spark

Parameters	**Hadoop**	**Spark**
Performance	Less optimal	More optimal
Latency	High latency computing	Low latency computing
Data	Process data in batch mode	Can process interactively
Usage	Batch processing with a huge volume of data	Process real-time data,from real-time events like Twitter, Facebook
Scheduler	External job scheduler is required	In memory-computation, no external scheduler required
Ease of use	Model is complex, need to handle low-level API’s	Easier to use, abstraction enables a user to process adat with high-level operators
Security and Cost	Highly secure and less costly since MapReduce model provide a cheaper strategy	Less secure and costlier since it requires in-memory solution

Why use PySpark?

1. **Requirements**

   ~~a.~~ Data coming in from multiple systems.

   b. Real-time as well as batch data.

   c. We need to perform data analytics over all these data inputs by building a system combining it.

2. **Why not Pandas?**

   a. Pandas is great for tabular data with millions of rows.

   b. Many features compared to PySpark.

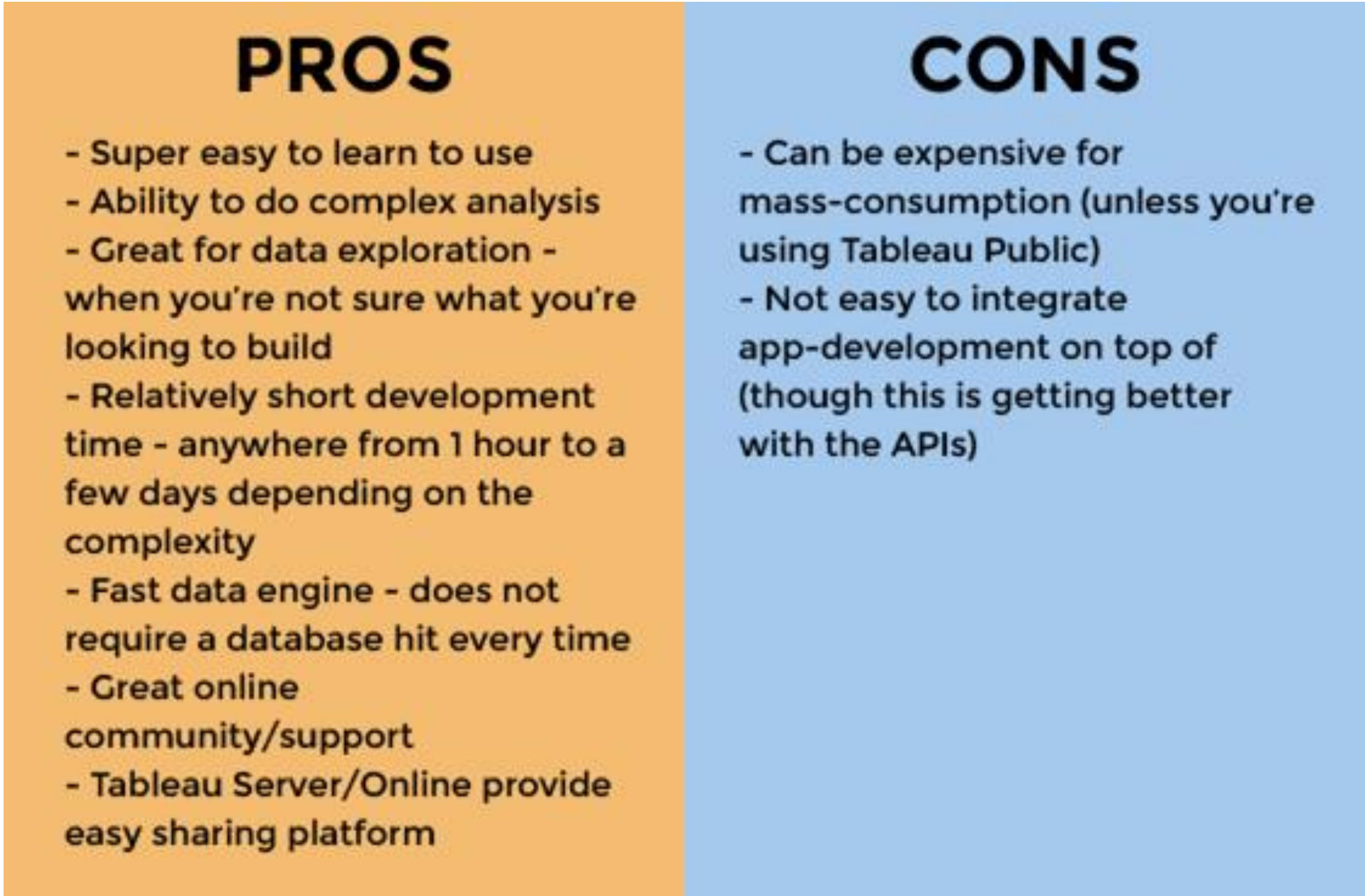
   c. Limitations - Distributed data and/or real-time data.

3. **Note**

   a. Our dataset is not distributed or even real-time.

   b. While we have used PySpark module, there is no significant difference in efficiency and execution times, compared to a module without PySpark.

Tableau



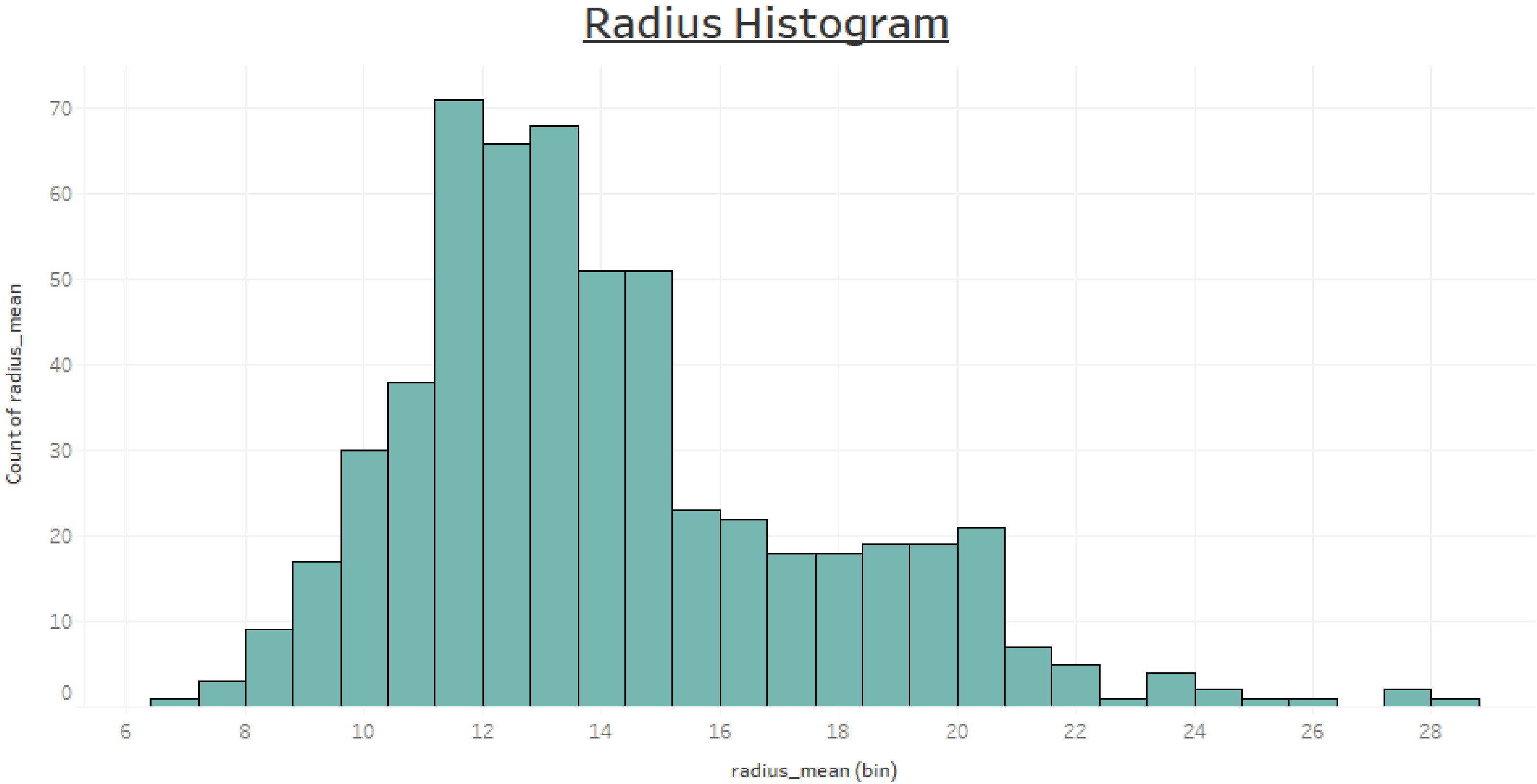
PROS

- Super easy to learn to use
- Ability to do complex analysis
- Great for data exploration - when you're not sure what you're looking to build
- Relatively short development time - anywhere from 1 hour to a few days depending on the complexity
- Fast data engine - does not require a database hit every time
- Great online community/support
- Tableau Server/Online provide easy sharing platform

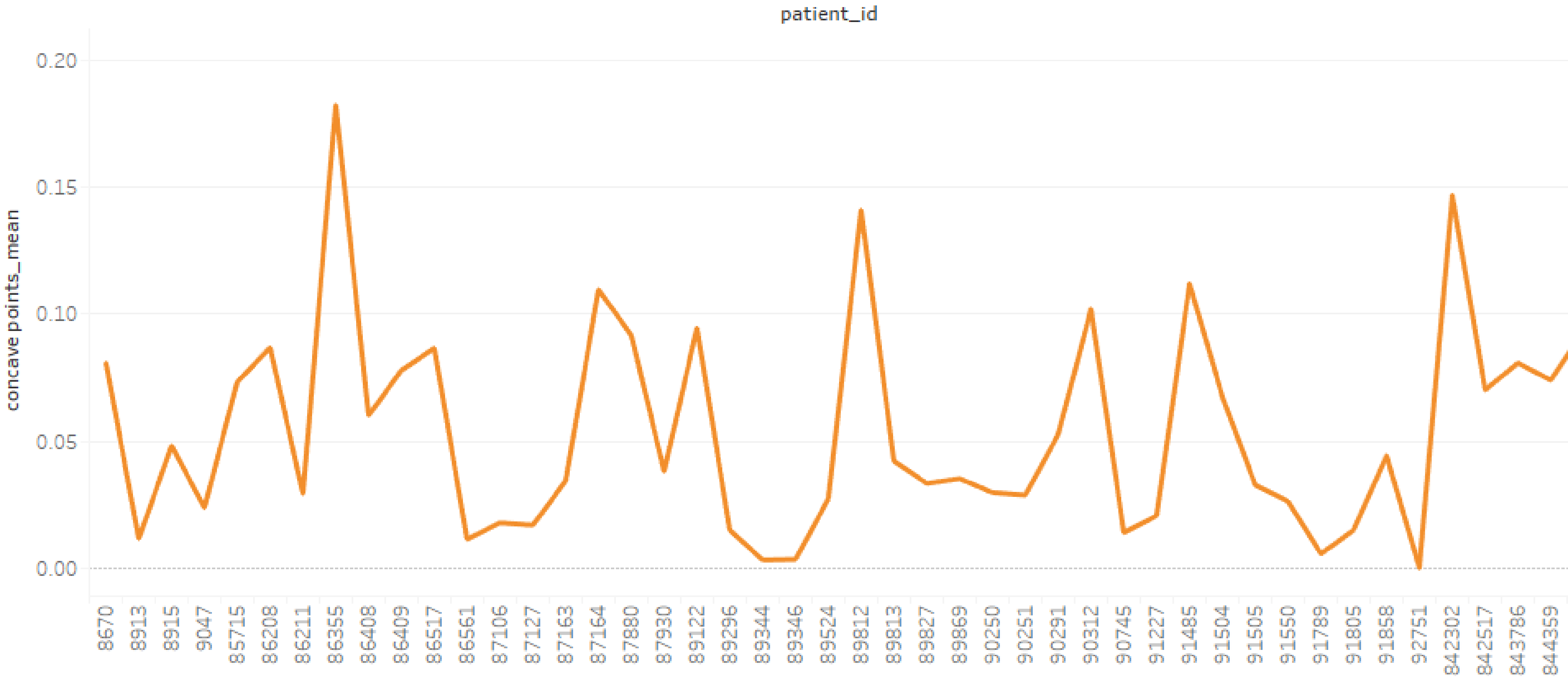
CONS

- Can be expensive for mass-consumption (unless you're using Tableau Public)
- Not easy to integrate app-development on top of (though this is getting better with the APIs)

Visualization

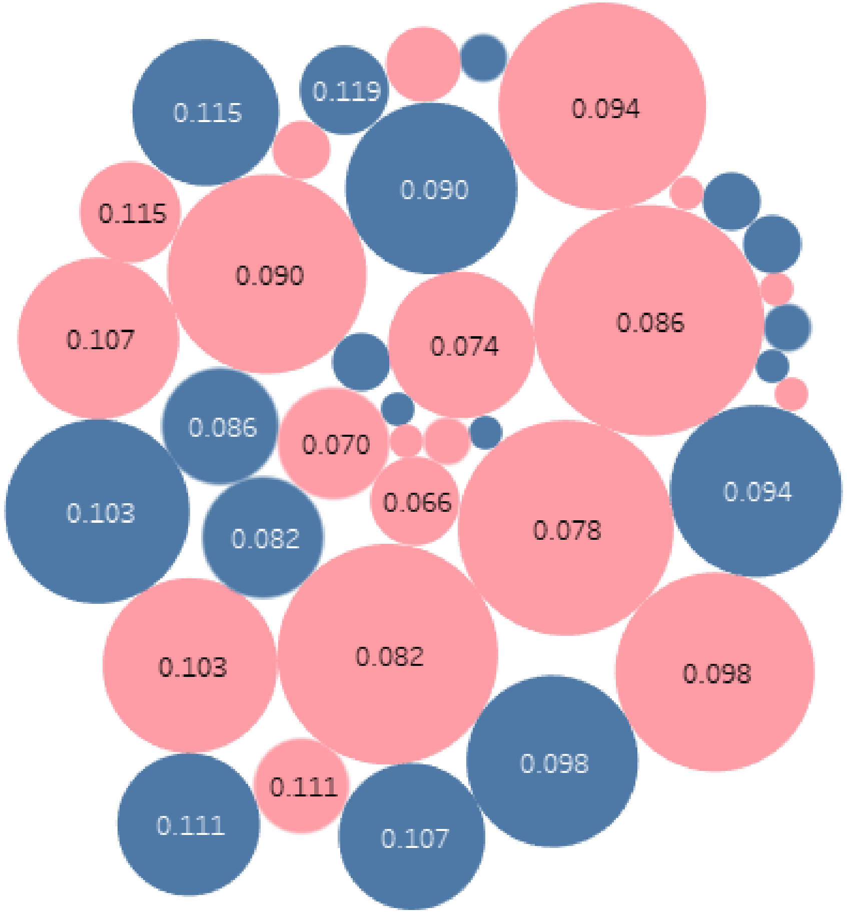


Line Graph showing concave points of each patient:

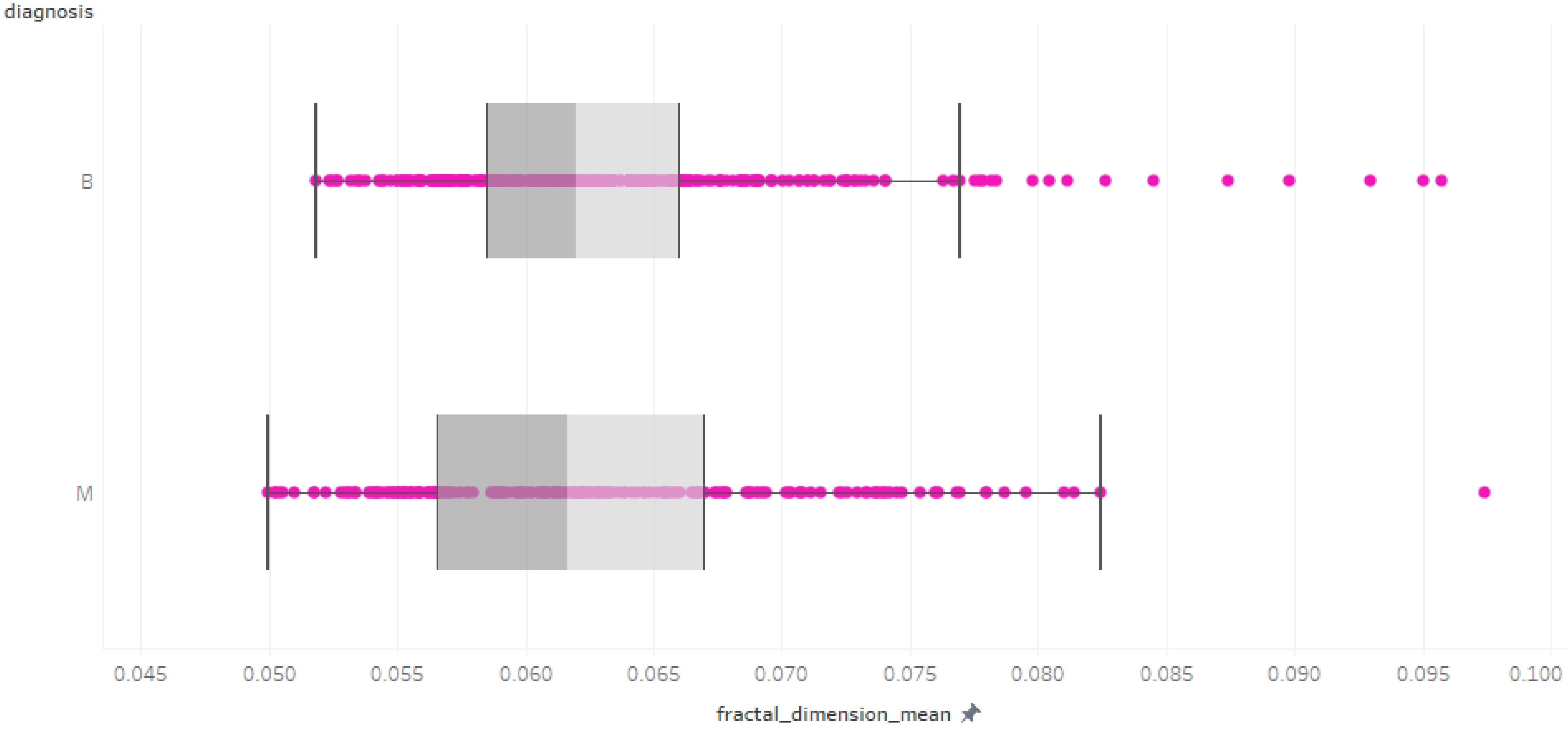


Bubble Chart

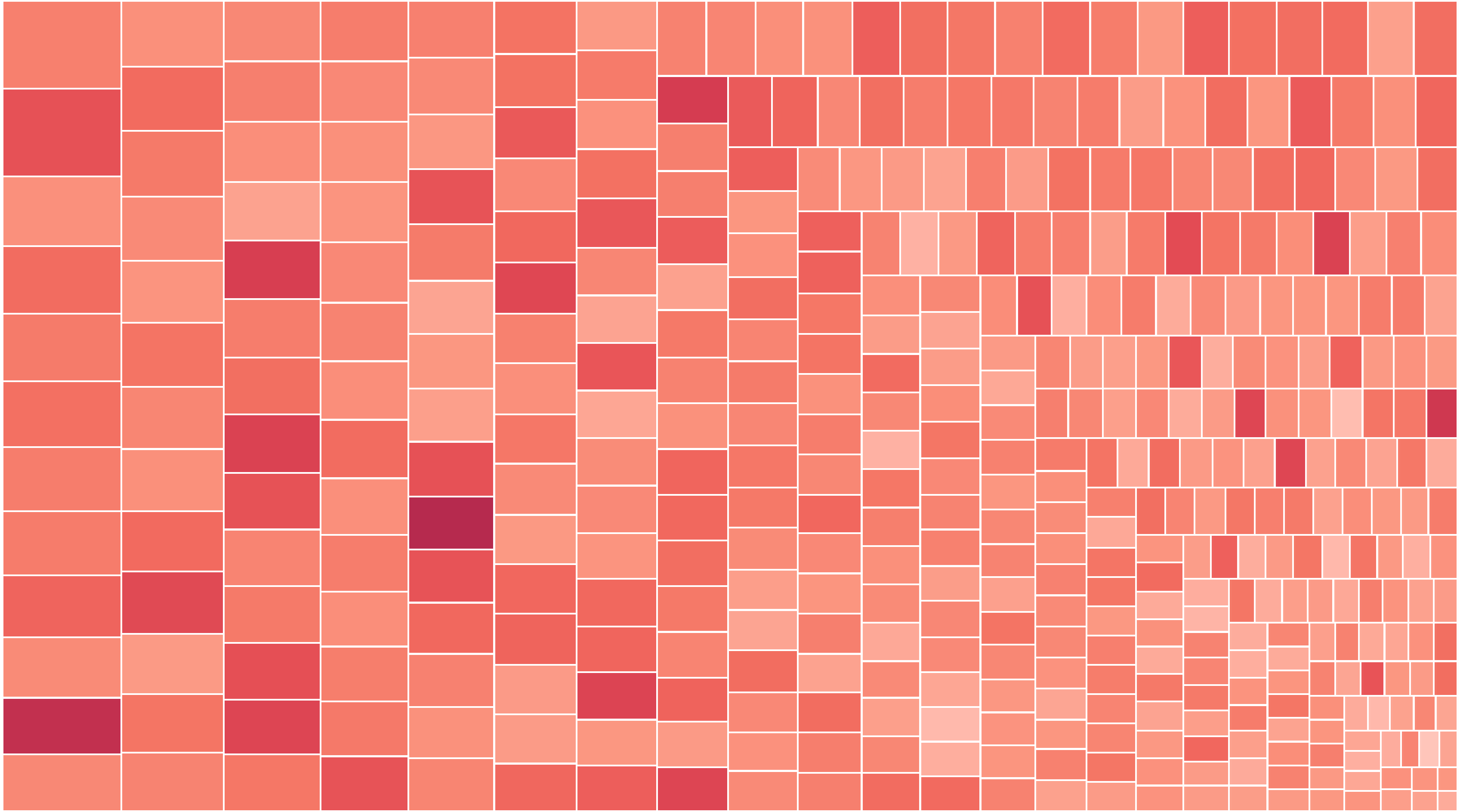
(colour:- Malignant, Benign; size:-no. of people diagnosed, label:- smoothness_index)



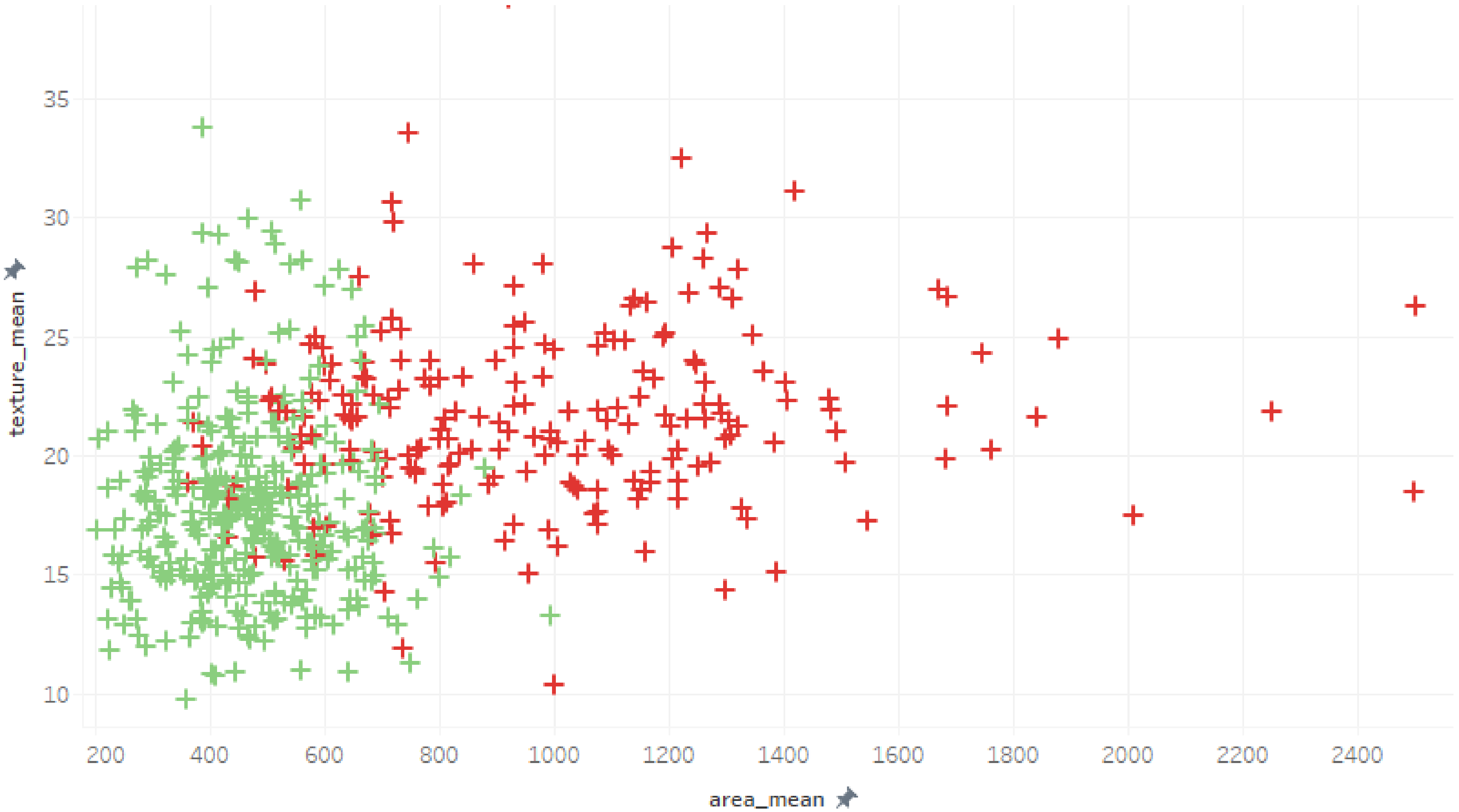
Box Plot for outlier detection of the *fractal-dimension*:



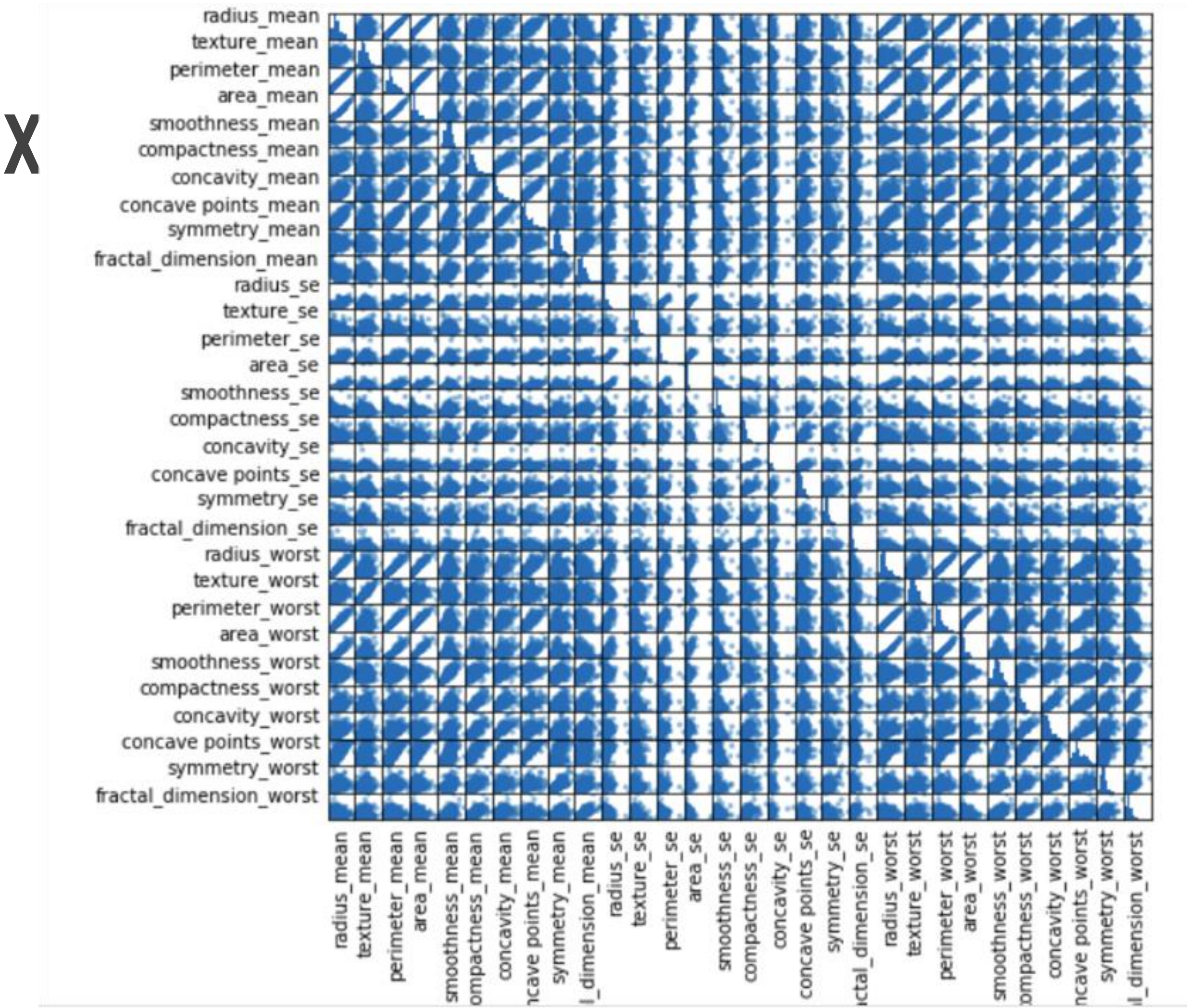
Tree Map (darker colour=more symmetry, greater size=more compactness)



Scatter Plot : Area vs Texture (Benign+, Malignant+)

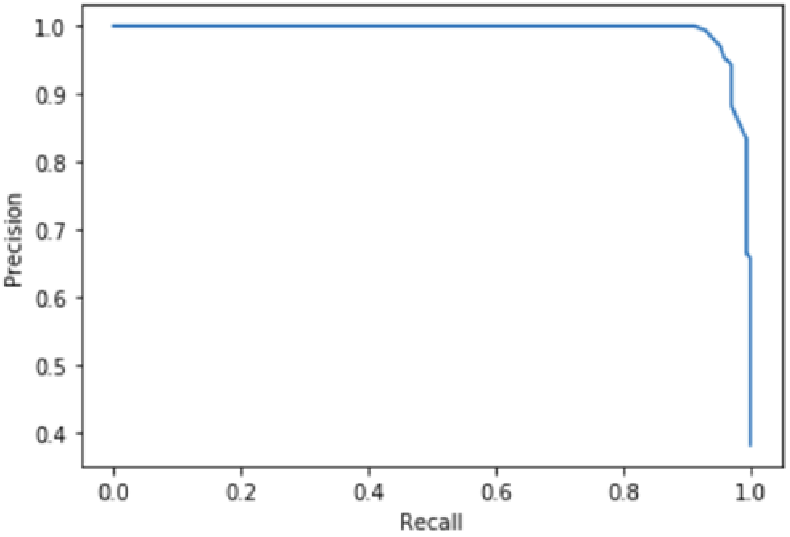


Scatter Matrix

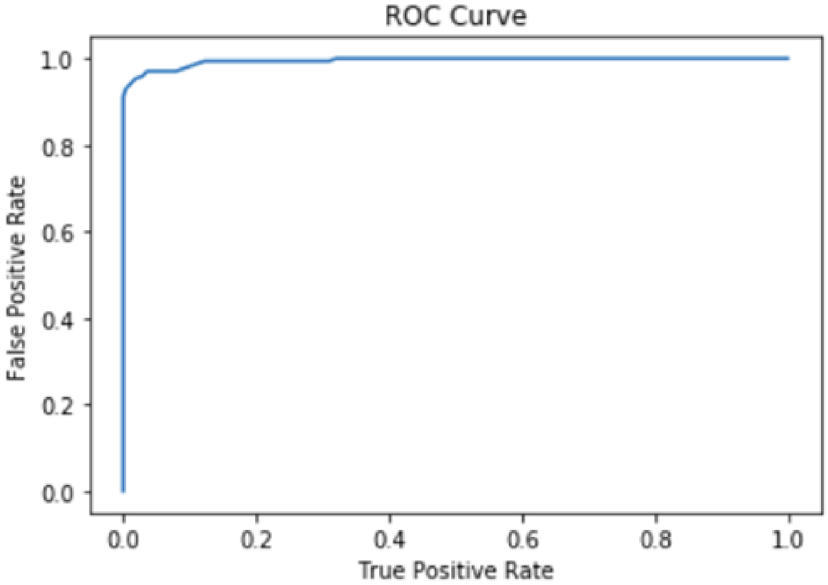


Performance

Precision vs Recall

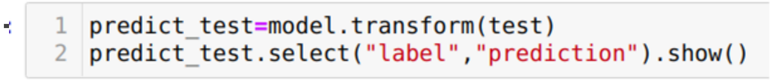


ROC curve for training data



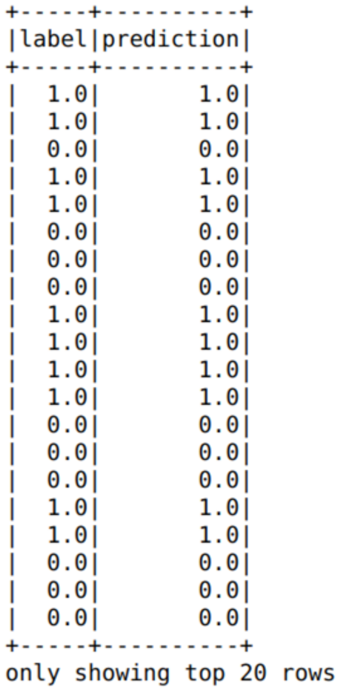
```
Training set areaUnderROC: 0.9950039968025579
```

Results



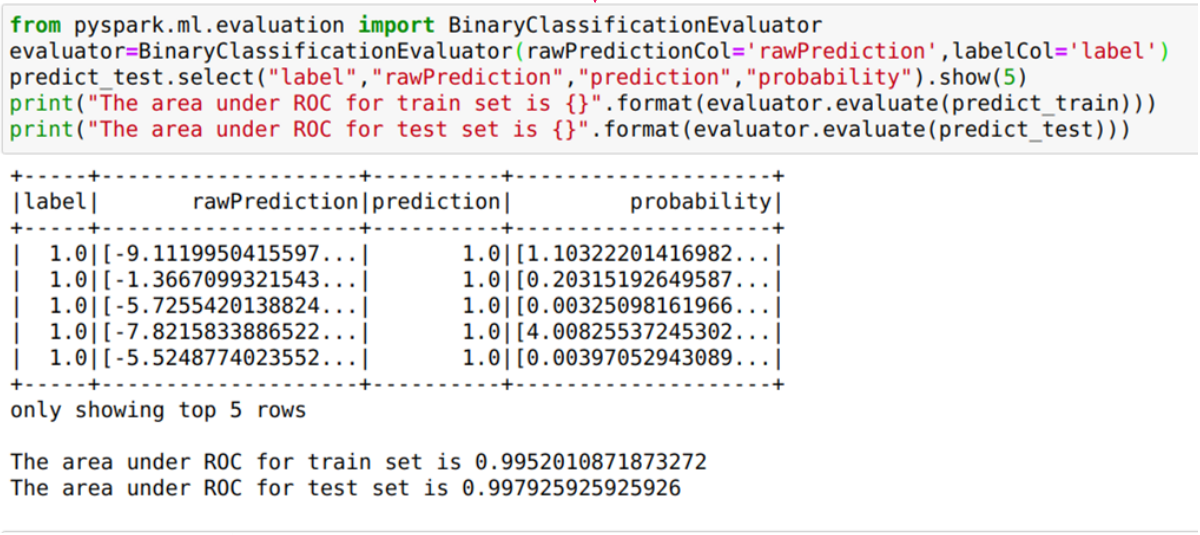
```
predict_test=model.transform(test)
predict_test.select("label","prediction").show()
```

Label & Predictions



```
+-----+----------+
|label|prediction|
+-----+----------+
|  1.0|       1.0|
|  1.0|       1.0|
|  0.0|       0.0|
|  1.0|       1.0|
|  1.0|       1.0|
|  0.0|       0.0|
|  0.0|       0.0|
|  0.0|       0.0|
|  1.0|       1.0|
|  1.0|       1.0|
|  1.0|       1.0|
|  1.0|       1.0|
|  0.0|       0.0|
|  0.0|       0.0|
|  0.0|       0.0|
|  1.0|       1.0|
|  1.0|       1.0|
|  0.0|       0.0|
|  0.0|       0.0|
|  0.0|       0.0|
+-----+----------+
only showing top 20 rows
```

Area under ROC for Test set



```
from pyspark.ml.evaluation import BinaryClassificationEvaluator
evaluator=BinaryClassificationEvaluator(rawPredictionCol='rawPrediction',labelCol='label')
predict_test.select("label","rawPrediction","prediction","probability").show(5)
print("The area under ROC for train set is {}".format(evaluator.evaluate(predict_train)))
print("The area under ROC for test set is {}".format(evaluator.evaluate(predict_test)))
```

```
+-----+--------------------+----------+--------------------+
|label|       rawPrediction|prediction|         probability|
+-----+--------------------+----------+--------------------+
|  1.0|[-9.1119950415597...|       1.0|[1.10322201416982...|
|  1.0|[-1.3667099321543...|       1.0|[0.20315192649587...|
|  1.0|[-5.7255420138824...|       1.0|[0.00325098161966...|
|  1.0|[-7.8215833886522...|       1.0|[4.00825537245302...|
|  1.0|[-5.5248774023552...|       1.0|[0.00397052943089...|
+-----+--------------------+----------+--------------------+
only showing top 5 rows

The area under ROC for train set is 0.9952010871873272
The area under ROC for test set is 0.997925925925926
```

Let’s defeat

breast cancer

together