CSIT5800 Mini-Project: Breast Cancer Diagnosis

Group 6: Lin Jialiang, Sun Wenjia, Feng Qingduo, Shen Zhengtao

---

Chapter I: Introduction & Basics

1. Background

COVID-19 (Corona Virus Disease 2019) has become a global pandemic, which has had a huge influence to our lives and word in almost every aspect. It is natural for us to pay more attention in this medical area.

Meanwhile, Breast Caner is another serous issue. It impacts 2.1 million women each year and about 627,000 women died from breast cancer in 2018 from WHO (World Health Organization).

Therefore, we want to do something related with these. Despite we may not contribute to it a lot, we can indeed help ourselves to know about it, sharing knowledge with our families and friends.

In general, as memtioned in the presentation and progress report, we have two-stage plan in this mini project. One is to make predictions on the open-source Breast Cancer dataset. The other is to extract important features from real raw Breast Cancer image to complete the whole process.

2. Basics

- Goal:
    1. Classification Task: Determine whether a new record data is related with benign breast cancer or malignant breast cancer
    2. Image Processing Task: Detect the breast cell boundaries from images and extract features like the average radius and area of cells for each image.
- Dataset:
    1. Wisconsin Diagnostic Breast Cancer:

       from UCI Machine Learning Repository (UCI-MLR)

       can also be found in Kaggle
    2. Open-source fine needle biopsies of breast masses in GIF-format. They were created by Dr. William Wolberg at the University of Wisconsin Hospital.
- General Process:
    1. Stage 1:

       Data Exploration --> Data Preprocessing --> Divide Training-set and Testing-set --> Feature Engineering --> Classifier Construction --> Result Analysis
    2. Stage 2:

       Image Preprocessing --> detect cell boundary through trained models --> calculate features --> output features for Stage1

Chapter II: Design & Findings

Stage 1: Breast Cancer Diagnosis

Recall that in stage 1, we are making classification on the dataset. Our process is very similar to common data mining procedure:

Data Exploration --> Data Preprocessing --> Divide Training-set and Testing-set --> Feature Engineering --> Classifier Construction --> Result Analysis

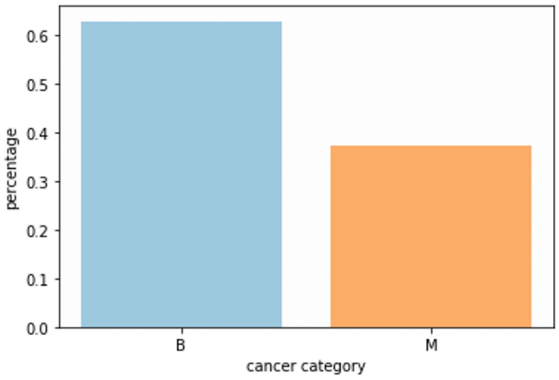
We don't want to introduce a lot of trivial methods and findings, but focus on some major issues and how we solve here:

- imbalanced class distribution
- feature importance
- feature redundancy

1.1 Major Issues

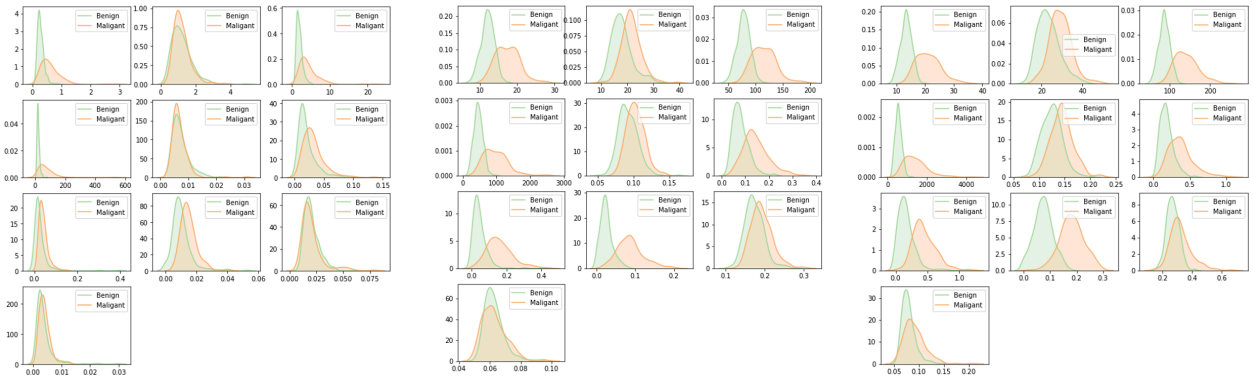
1.1.1 imbalanced class distribution:

We have an imbalanced dataset, and we need to handle this problem or it may affect the performance of the model. The disadvantage of imbalance is that it may bias the prediction to the majority class. We can utilize sampling to alleviate this problem, by under-sampling, over-sampling or combine them together.

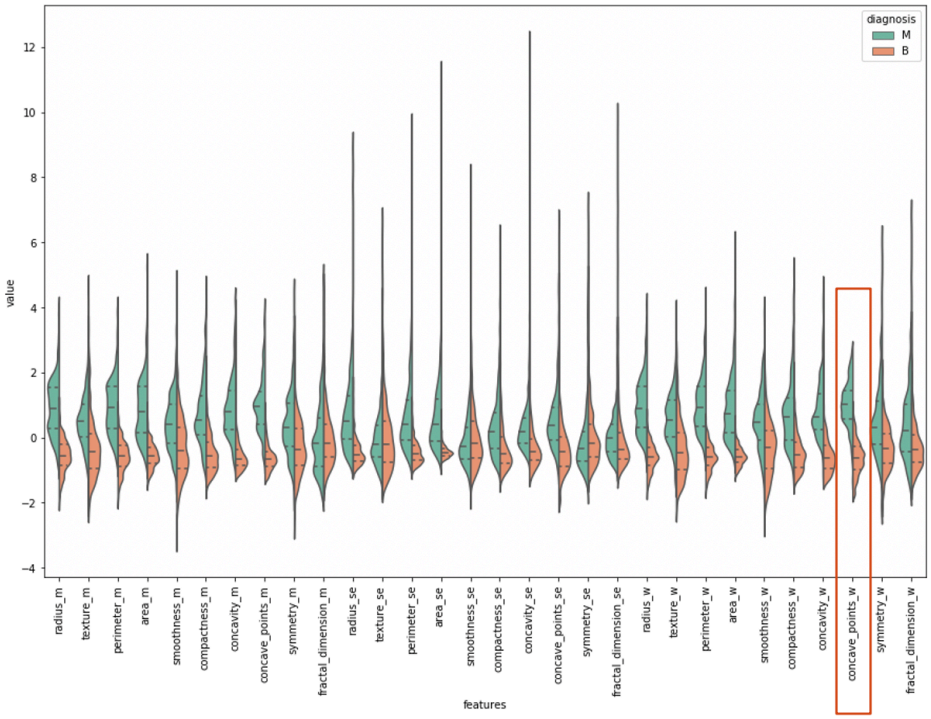


1.1.2 feature importance

From the observation of the feature values distributions of different classes, we notice that the shapes may be very different in different classes. In order words, these features may have better effectiveness to distinguish the class.

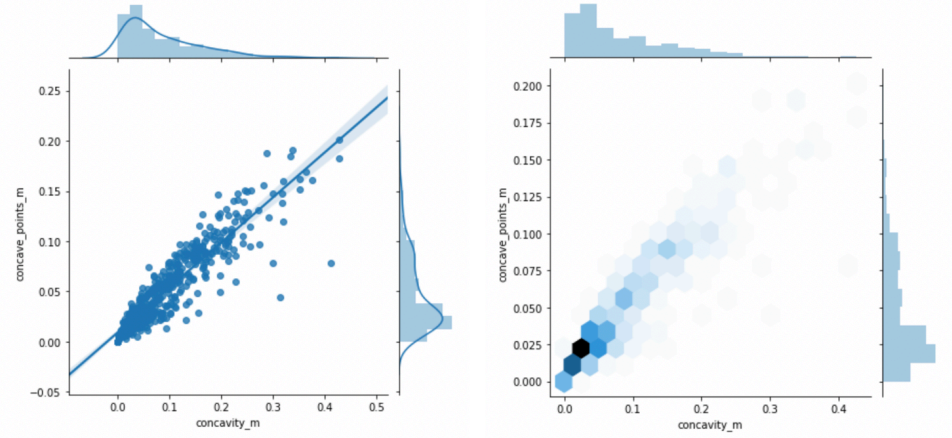


We further plot the violin diagram and find that some attributes have good distinctive performance, like "concave_points_w", while others do not. So different features may have different importance in terms of classification.

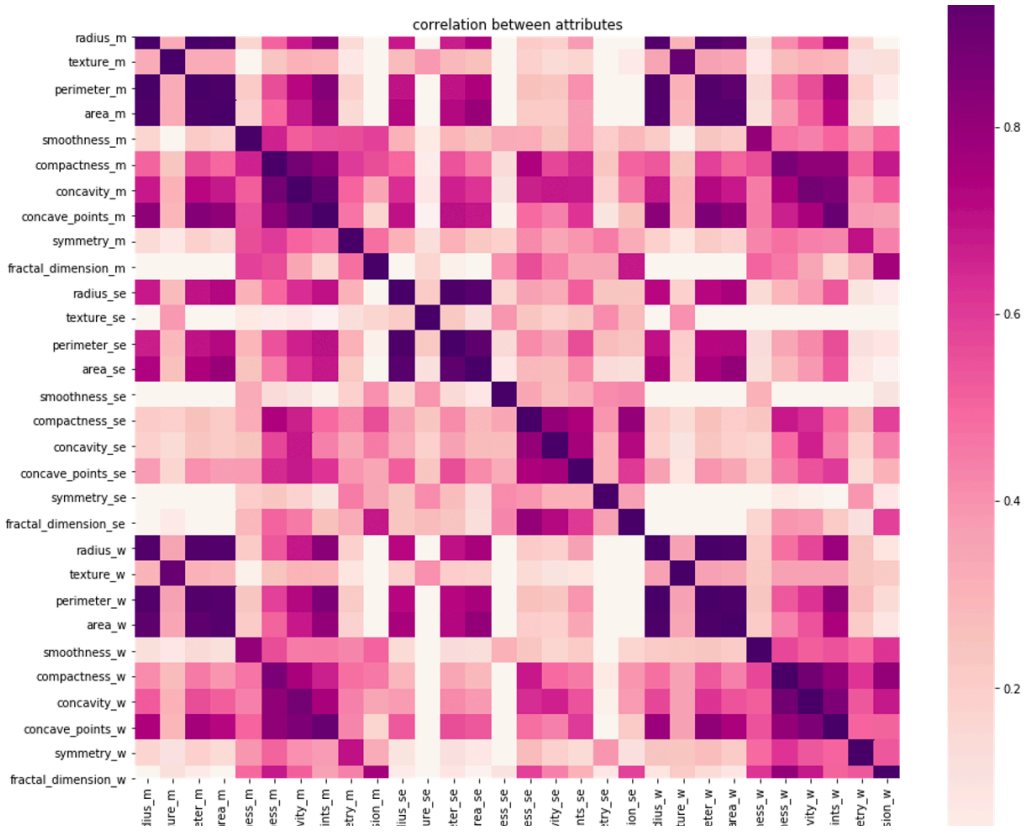


1.1.3 relation between features

From the following figure, we notice that there are some attributes with high relations like the '*concavity_m*' and '*concave_points_m*'. That means there may be feature redundancy here.



We further compute and plot the correlation matrix to see the correlation between pairwise features. We notice that there are some attributes with high relations like perimeter and area. So there is some redundancies within the features, which we can try to solve.



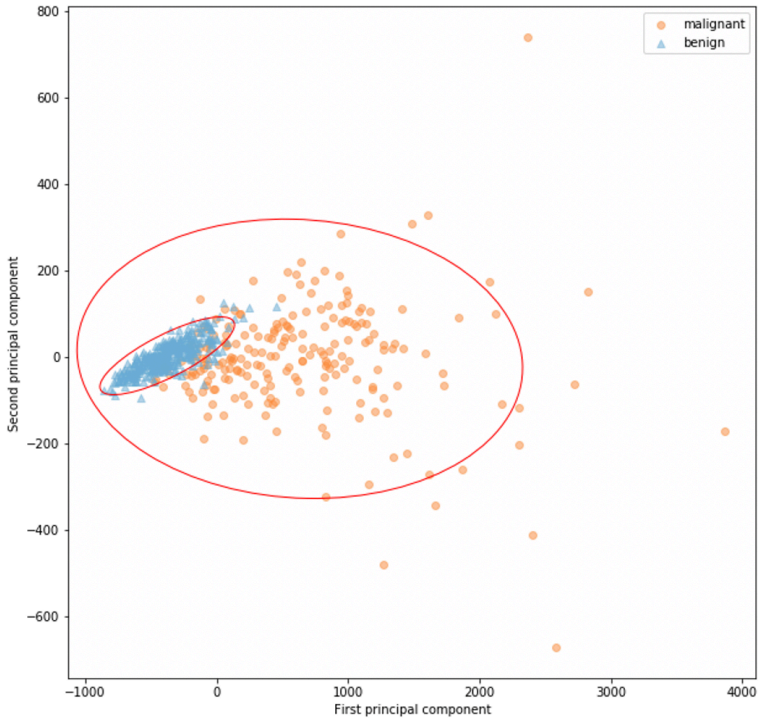
1.2 Corresponding Methods

Recall that we have these three major issues and we try to utilize the following techniques to alleviate them:

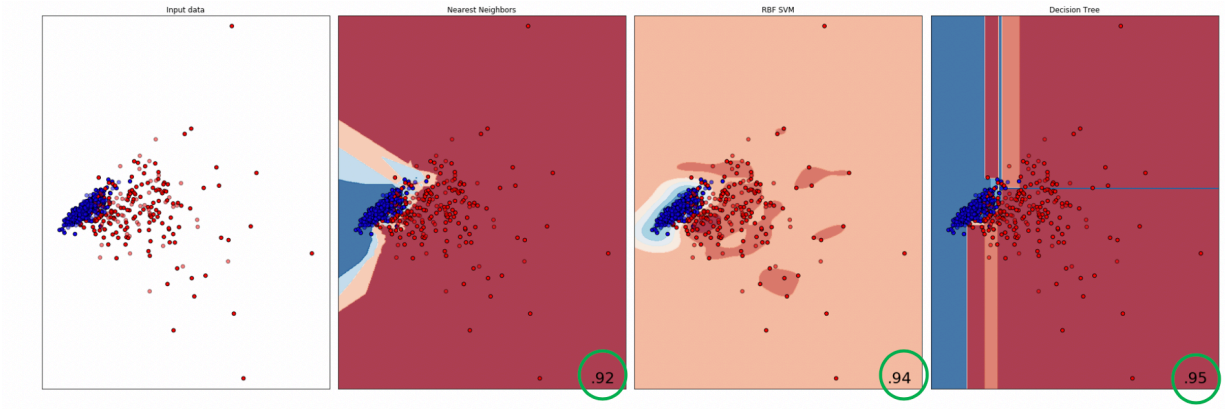
- feature redundancy <-- PCA method
- feature importance <-- information theory
- imbalanced class distribution <-- sampling method

1.2.1 PCA to reduce dimension

We reduce the dimension into 2-D with PCA method in order to have a better visualization view. The following figure is the data visualization diagram. We can see how well these the two principal components work because we can see that the benign cases have a compact distribution while the malignant ones appear seperately from the diagram with confidence ellipse (95%). The confidence ellipse of the benign cases itself can be a good decision boundary to do the classification.

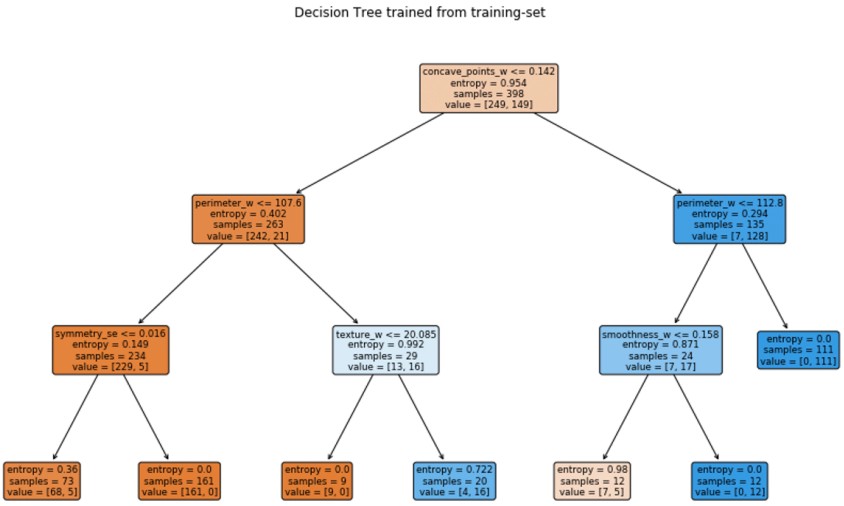


Due the good performance of these two principal components, that's no wonder that we can apply KNN or SVM or Decision Tree Classifiers to achieve good performance. And the visualization figure is shown as the following: (The two digits in the right bottom of the sub-image represent the accuracy, i.e, Decision Tree achieves the best accuracy of 0.95)

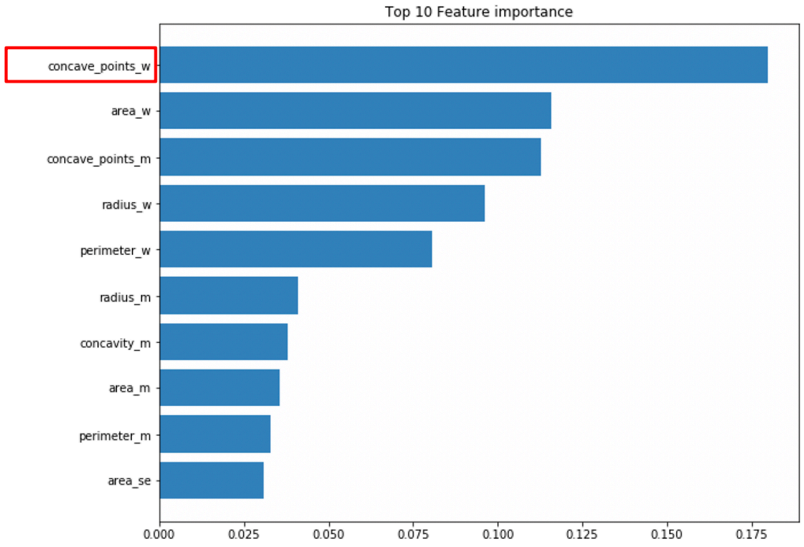


1.2.2 Tree-type Models

Since Decision Tree achieves well enough performance in 2-D dimension space, we apply it on original dimension space to see the important of these features. The metric we use is **entropy**, and here is the figure of Decision Tree:

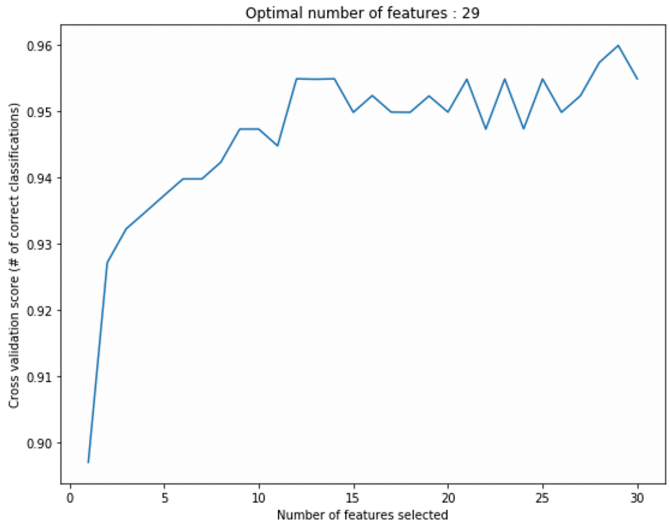


Form the tree plot diagram, we can see that most data records can be successfully classified into one category. And we also take a look at the important of features since tree-type model determine the category based on these. We use Random Forest with 100 trees and get this result:



From the top 10 important features, “concanve_points_w” are the most important feature in Random Forest with 100 trees. It is consistent with our observation in violin plot of feature distributions.

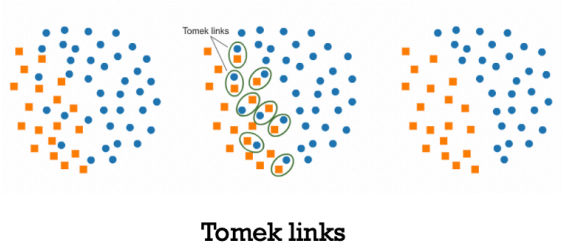
We can also utilize Recursive Feature Elimination (RFE) method to determine the most suitable number of features that we should select. This method uses the model accuracy to identify which attributes and the combination of attributes are most important. Here is our result:



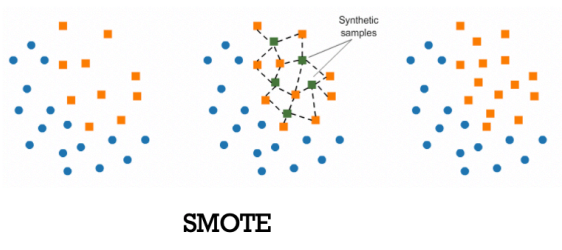
1.2.3 Overcome imbalanced data

We use sampling method to balance the data here. We try several methods: random under-sampling, random over-sampling, Tomek-links, SMOTE, SMOTE + Tomek-links.

Random under-sampling is to randomly delete the data records of the majority class. And random over-sampling is to randomly pick and duplicate the data records of the minority class.



Tomek links



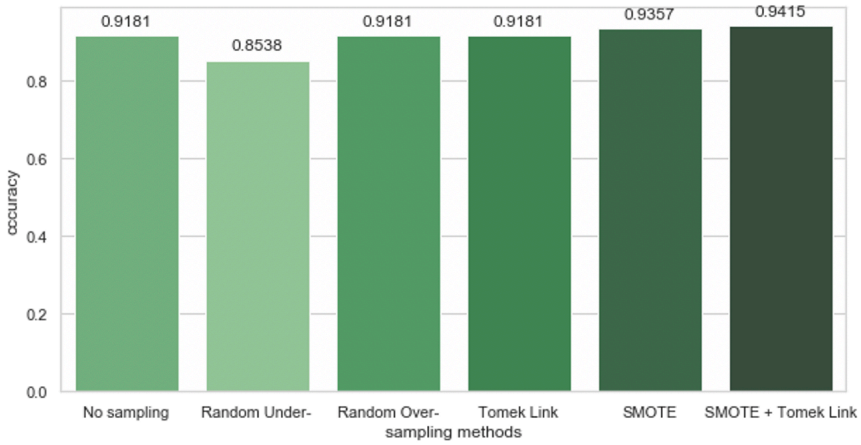
SMOTE

Tomek-links method is one of the under-sampling method. Tomek links indicate pairs of very close instances, but of opposite classes. Removing the instances of the majority class of each pair increases the space between the two classes, facilitating the classification process.

SMOTE (Synthetic Minority Oversampling Technique) consists of synthesizing elements for the minority class, based on those that already exist. It works randomly picking a point from the minority class and computing the k-nearest neighbors for this point. The synthetic points are added between the chosen point and its neighbors.

SMOTE + Tomek-links is to firstly apply SMOTE to increase data records and then apply Tomek-links to make the distance between two classes cleaner.

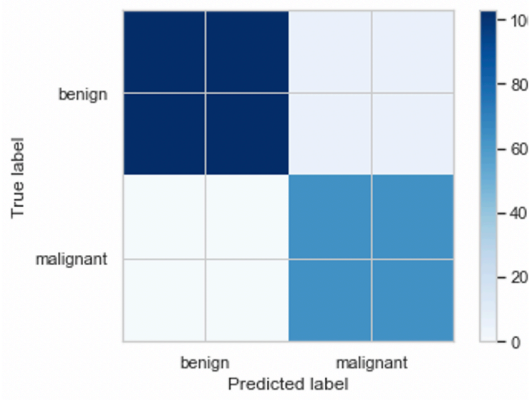
Here is the comparison between different sampling methods:



We use variable-controlling approach to compare their performance. The set up of the experiments is the same except sampling method. And we can find out that SMOTE+Tomek-Link achieves the best performance, which is not surprising due to the above discussion.

Finally, we build our model with:

- SMOTE + Tomek-Link sampling method
- Use the best features selected
- Utilize Random Forest Model



The result is : **Accuracy: 97.08%**

And from the confusion matrix, we know that the prediction is balanced, not overfitting.

Stage 2: Feature Extraction

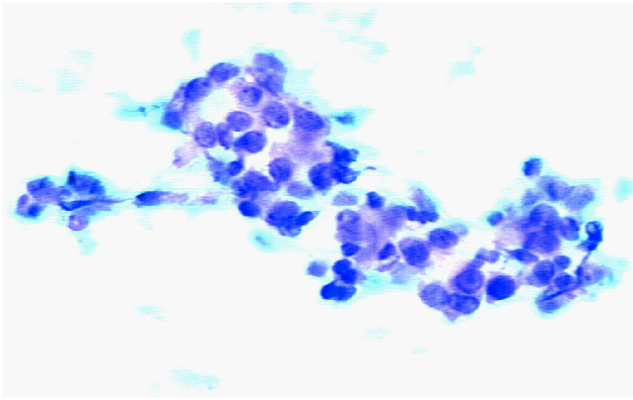
In order to improve the performance of classfier and cover more techiniques in big data, we decide to extract features from original images instead of using the existing dataset directly.

Firstly, we want to introduce the images we use. These images were taken from fine needle biopsies of breast masses. They were created by Dr. William Wolberg at the University of Wisconsin Hospital. They are in GIF format.

We divide feature extraction into two subparts: boudary recognitions and attributes calculation.

2.1 Boundary Recognition

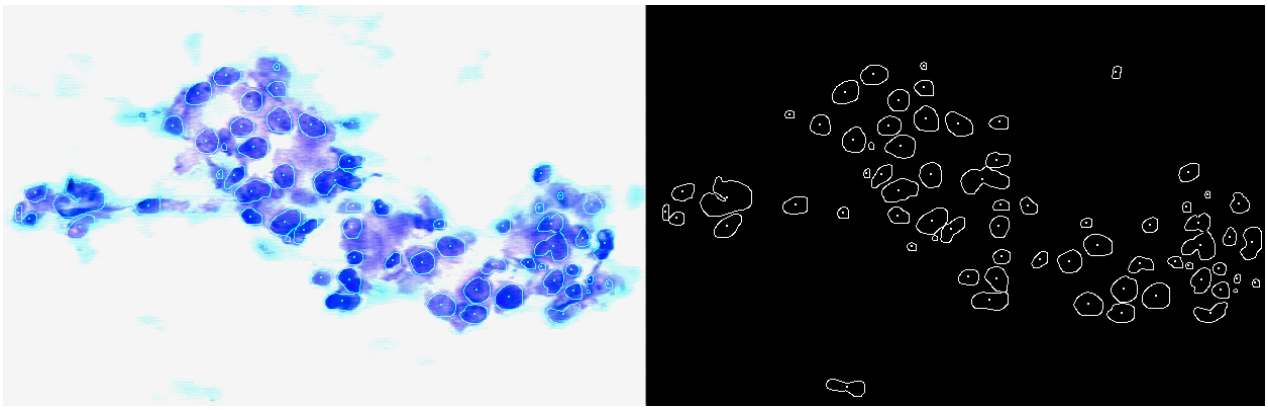
We needs to detect the location and draw the boundary of the nuclei from tissue microscopic images. The following picture shows the one of the orginial image.



Model is based on U-net with contour enhancement in loss function. Overlap patch based strategy is used to

- adapt to the variant input image size
- use random clip and rotation for the data augmentation.
- each region in output mask is determined by combining inference results from multiple patches.

The results of boundary detecting shows in the following pictures.

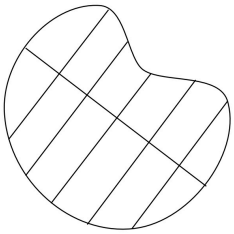


2.2 Attributes Calculation

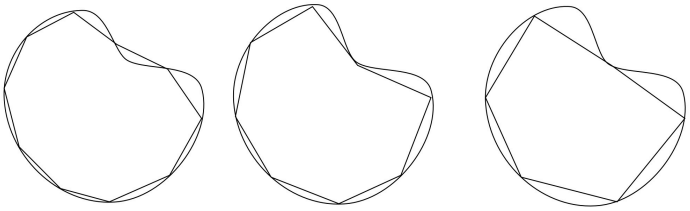
Based on the results from boundary recognition, we further calculate the relevant attributes.

- Radius: Mean of distances from center to points on the perimeter
- Perimeter: Total distance between nucleus boundary points
- Area: The number of pixels on the interior of the snake and adding one-half of the pixels in the perimeter.
- Compactness: Perimeter^2 / Area - 1.0
- Smoothness: Differences between the length of radial line and the mean length of the lines surrounding it.
- Concavity: Draw chords between non-adjacent boundary pints and measure the extent to which the actual boundary of the nucleus lines on the inside of each chord.

- Concave Points: Similar to Concavity, but measure the number of concave portions of the contour.
- Symmetry: Draw teh major axis or the longest chord through the center. The length difference between lines perpendicular to the major axis to the cell boundary in both directions.



- Fractal Dimension: Coastline approximation -1. The perimeter of the nucleus is measured using increasingly larger 'rulers'. As the ruler size increases, decreasing the precision of the measurement, the observed perimeter decreases. Plotting these to values on a log scale and measuring the downward slope gives (the negative of) an approximation to the fractal dimension. As with all the shape features, a higher value corresponds to a less regular contour and thus to a higher probability of malignancy.



- Texture: standard deviation of gray-scale values

In order to increase the accuracy of attributes, we use the mean, standard error, and "worst" or largest (mean of the three largest values) of these features were computed for each image, which results in 30 features.

These 30 attributes are utilized in stage 1. The feature results can be seen as the following:

radius_m	perimeter_m	area_m	compactnes	smoothness	concavity_m	concavity_p	symmetry_m	fractal_dime	texture_m	radius_se	perimeter_s	area_se	compactnes	smoothness	concavity_se
8.974	55.844	327.8807	11.05	1.2689	501.8653	55.8165	444.637	-1.3969	1176.744	4.3151	28.0027	230.5436	3.5811	0.9066	369.2861
11.0614	75.675	309.925	21.7707	1.0443	371.5679	75.625	1413.2822	-1.3714	799.2044	22.4959	181.5869	503.4866	67.4817	0.7719	287.7856
8.0019	51.7978	306.236	11.5652	1.4812	449.9073	51.7528	660.8562	-1.0656	736.6904	5.5039	38.7635	357.0988	5.1206	1.3359	526.3051
8.0532	54.9016	342.7705	11.9539	1.3407	509.5129	54.8361	909.2405	-1.3024	732.0401	6.1873	47.424	416.963	6.935	1.2344	627.2332
7.284	46.8658	217.2953	11.6184	1.3218	315.4813	46.8255	358.9703	-1.0894	1145.8618	3.7948	27.8331	182.4223	4.945	0.9799	270.4757
6.9697	43.5294	189.1912	11.5117	1.1569	288.0376	43.5	296.3281	-1.0998	542.7589	3.8	28.0223	149.1208	9.194	0.8804	239.9495
5.0721	32.2917	108.9861	11.6123	0.9787	148.205	32.2361	191.056	-0.7946	518.2479	3.4938	26.7012	122.0446	11.0915	0.7956	170.1111
7.5063	47.0693	233.0792	11.2356	1.1494	339.2513	47.0495	321.8769	-1.2318	552.7384	3.8401	26.9213	188.9209	4.7443	0.7943	281.2156
7.314	48.7308	228.7308	12.1591	1.5923	295.7356	48.5769	477.8752	-1.2423	519.014	5.0456	39.4184	260.516	6.1733	1.2946	305.0074
6.5364	39.6706	184.9412	10.3058	1.0037	290.0288	39.6471	183.1146	-1.0451	1167.6869	3.472	21.6916	166.12	2.9207	0.7288	277.6946
7.3355	56.4661	288.9237	13.7139	1.3946	349.825	56.4237	829.8847	-0.8685	566.9448	6.8355	81.0945	662.1331	11.4751	1.4919	601.1508
8.9831	59.9211	375.3158	11.8073	1.5247	556.107	59.7632	775.6942	-1.1792	479.0376	6.0073	43.3732	358.2603	6.8374	1.2502	554.7631
7.0977	46.1194	243.7463	11.3056	1.0704	363.9208	46.1194	309.9167	-1.0316	257.2144	4.7999	33.5802	262.163	5.0365	0.6782	402.3157
9.6459	63.2879	395.6818	13.2591	1.6195	609.4097	63.2576	4134.1724	-1.3444	522.9974	5.7189	40.5271	348.2903	8.8519	1.2643	560.7606
7.6056	47.4776	237.3731	10.7578	1.2318	350.2874	47.4478	276.9322	-1.2101	448.915	3.6437	27.9212	192.8525	3.7859	0.8963	260.6575
7.0614	49.0169	152.1017	17.7042	0.9593	145.9277	49	328.0754	-0.7915	169.3398	16.4935	132.3929	366.1688	48.7419	0.778	155.4485
6.2293	41.4067	183.34	11.1749	1.1127	255.7779	41.34	409.6081	-0.9227	597.0661	3.9228	38.135	264.784	6.1699	0.8559	317.6418
8.5524	57.8305	315	13.1128	1.5239	453.4237	57.8305	987.7374	-1.2243	1007.2624	4.7996	37.6017	302.3245	6.6135	1.183	458.1056
8.6203	51.9206	312.381	10.2875	0.9913	485.0159	51.8889	264.821	-1.41	1031.9398	4.3051	26.7661	235.7599	3.1836	0.6681	375.2641
7.4707	48.9697	266.7121	11.6389	1.3175	397.1971	48.8939	384.6639	-1.1984	998.0661	5.0014	36.386	292.5003	6.1302	1.0181	445.3008

Chapter III: Acknowledgement & Refference

Sincere thanks to Dr. Cecia Chan, Dr. Desmond Tsoi and TA Wallace Mak. Without their helps, we cannot achieve this.

Also, we make use of the following open-source dataset, toolkits and papers. With them, we can polish our work. Big Thanks.

Wisconsin Diagnostic Breast Cancer dataset: [https://www.kaggle.com/uciml/breast-cancer-wisconsin-data]

Breast Cancer Image: [ftp://cs.wisc.edu/math-prog/cpo-dataset/machine-learn/WDBC/]

pandas: [https://pandas.pydata.org/docs/]

scikit-learn: [https://scikit-learn.org/stable/]

matplotlib: [https://matplotlib.org]

seaborn: [https://seaborn.pydata.org]

imblearn: [https://imbalanced-learn.readthedocs.io/en/stable/api.html]

Tensorflow [https://tensorflow.google.cn]

OpenCV [https://opencv.org]

Olaf Ronneberger, Philipp Fischer, Thomas Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, arXiv:1505.04597.

K.Chen, N. Zhang, L.S.Powers, J.M.Roveda, Cell Nuclei Detection and Segmentation for Computational Pathology Using Deep Learning, SpringSim 2019 Modeling and Simulation in Medicine, Society for Modeling and Simuation (SCS) International (accepted).

W. Nick Street, W. H. Wolberg, and O. L. Mangasarian "Nuclear feature extraction for breast tumor diagnosis", Proc. SPIE 1905, Biomedical Image Processing and Biomedical Visualization, (29 July 1993);