



Malignant Tumor Detection Using Machine Learning through Scikit-learn

Arushi Agarwal¹, Ankur Saxena²
Amity University, Uttar Pradesh
arushiagarwal14@gmail.com, asaxena1@amity.edu

Abstract. Cancer has always been one of the greatest causes of death around the world since a long time. There have been years of research and development put into cancer but we are still unable to find an overall cure which can guarantee the eradication of cancer from the patient. Putting into attention the number of people affected by cancer, we need to develop better ways of diagnosis and treatment. In this article we go through Machine Learning and how it can be very effective in identifying and studying cancer tumors. We have used very basic steps to create a strong machine learning program which is able to identify the tumor as malignant or benign. Using Python and its open source libraries makes it possible for almost anyone to use this approach. We have used KNN classifier and Logistic Regression to make 2 models to compare the results and determine the more accurate algorithm.

Keywords: cancer, machine learning, python, anaconda, knn classifier, logistic regression, malignant, benign

1 Introduction

Machine learning has been used in cancer research from a long time now. We have been using artificially intelligent machines since even before the term “artificial intelligence” was coined. It has aimed at detection and diagnosis of cancer. For some patients, machine learning can even be used to get the individual personal records and treatment path.

Cancer, a condition in which a group of cells in body exhibit abnormal growth and may spread to other parts of the body, is one of the leading causes of mortality. It is a disease caused by mutations of cells that has occurred in the genes. These cancerous cells are called malignant cells and contract with benign cells which divide under the control of the body. When the division of cells leads to a swollen mass or extra tissue development, it is called a tumor. There are various kinds of cancers, including breast cancer, lung cancer, leukemia, prostate cancer, lymphoma etc. Most of these are more common in older ages, due to the damaging and older age of DNA, which increases the risk of mutations.

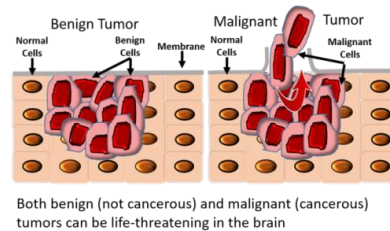


Fig.1. Benign and malignant tumors

Typical cancer treatment includes a surgery to remove the tumor, if possible, followed by chemotherapy and radiation therapies. Targeted immunotherapy and hormone therapy is also provided in cases where necessary or beneficial. However, even if the treatment works, it is very hard to say whether the patient is 100% cancer free, as recurrence is a big possibility in cancer.

According to WHO, 1 out of every 8 people die of cancer. However, even after millions of dollars of reasearch we are unable to find a cure for this disease. There are many reasons behind the same, such as-

- 1) Methods of studying cancer include the cultures of tumor cells extracted from the body. These may not provide the most appropriate results, as they lack the complexity of a real live tumor in the body.
- 2) Clonal heterogeneity- It is the rise of similar, but not same type of cancer cells in a single patient. Drugs that may affect one type may not affect the other type of cancer cell.
- 3) The suppression of immune system by the tumor.
- 4) Some malignant cells can adapt and undergo further mutation to merge in with the normal cells of the body.
- 5) The delay in diagnosis.

Cancer diagnosis is one of the most important components of the treatment. An accurate and fast diagnosis method will make it much easier for the patient and doctors to take a further course of action before the tumor grows in size and becomes more difficult to treat.

This is where, machine learning can be a solution. It enables computers to learn, and then perform functions independently without the need of programming them specifically for each task. Data makes the base and backbone of the machine learning model. To train our model, we need to provide data to it, so that it learns and is trained, and can be tested later.

The type and form of tumor can tell a lot about how harmful it is and what are the chances of it being malignant and benign. With the help of machine learning, we can train our model to identify the features which can most likely point to malignancy and otherwise. We can use a pre-existing dataset and train our model to see how accurately it can work.

There are many algorithms which are suitable for doing this task, but we will be using KNN classifier and logistic regression.

Logistic regression is a method for analysis of a dataset in which there are one or more independent variables that present any outcome. The outcome is measured with a variable (in which there are only two possible outcomes).

In this method, we follow similar steps as linear regression, and we try to divide the data by minimising the error. However, we can think of the errors as a penalty points. We try to shift the division to the point where this penalty points are minimum.

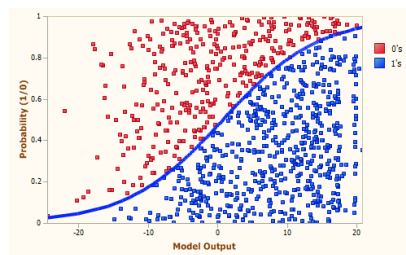


Fig.2. Logistic regression illustrated

K-nearest neighbors or KNN clustering is another class of algorithms that works on forming clusters of known points to determine which cluster our called point will belong.

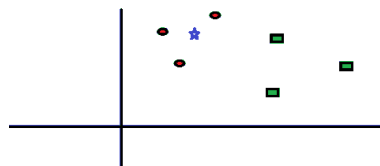


Fig. 3. Random points and our blue star

In the above image, the blue star is the point which we need to classify or group, and the red circles and green squares are the clusters and classes where it can go. Now we will draw a circle around the blue star and see which cluster it seems to neighbor with more. In KNN, we can determine the number of neighbors we want to use to form the cluster. Let's take $k=3$.

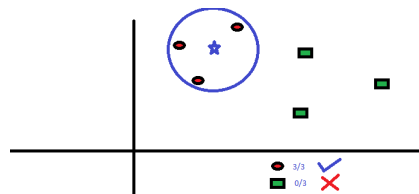


Fig. 4. Classifying our blue star with nearest neighbors

From the image it is very evident that the blue star belongs to the cluster of red circles.

This is how KNN clustering works.

2 Review of Literature

2.1 Machine learning applications in cancer prognosis and prediction- By Konstantina Kourou, Themis P.Exarchos, Konstantinos P.Exarchos, Michalis V.Karamouzis, Dimitrios I.Fotiadis

The early diagnosis of a cancer type is a necessity in cancer research and treatment, as it can stimulate the following clinical management of affected individuals. The necessity of classifying cancer patients into high or low risk groups leads research teams, to study machine learning methods.. Moreover, the ability of machine learning to detect features from complicated datasets reveals it's ability. These ML algorithms like Artificial Neural Networks , Bayesian Networks, Support Vector Machines and Decision Trees have been used in cancer research. However, these models need to be improved and validated in order to be considered fit to be used in usual clinical practice. The predictive models presented in this paper are based on supervised ML techniques.

3 Methodology

We will be making a machine learning program that will detect whether a tumor is malignant or benign, based on the physical features.

3.1 Getting the system ready

We will be using Python for program, as it comes with a lot of libraries dedicated to machine learning and data science, which will make our task much easier.

Python 3.6 is the most popular and updated version. Even though that is the official website, a much more convenient way of installing Python is through *Anaconda*, especially when we want Python for machine learning. Anaconda makes it very easy and fast to download all the libraries of python.

We can download Python and other dependencies from www.continuum.io/downloads.

After the installation, we will find Anaconda prompt in our systems which can be used to open, install, delete and do other functions related to our libraries.

Jupyter notebook is considered to be the best coding environment for Python because of the functionality, interface and the extra features it provides. It also makes it very easy to work with external files and load them into our code, and we can easily use it through Anaconda.

3.2 Working on the program

Machine learning is data driven. Some of the steps that are involved in the process are- *acquiring data, data wrangling, training our model, testing our model and improving the model.*

We will be using a Python library specifically used for machine learning and data science, *Scikit-learn*. Sklearn makes it very easy to work with the algorithms, and it comes pre installed with Anaconda.

Sklearn also provides many datasets that can be used for training purposes. To see how we can use machine learning for cancer diagnosis, we will be using Sklearn's Breast Cancer Wisconsin Database which contains features and targets of 569 samples with 30 features.

3.3 Using KNN Classifier

To start with our program, we need to import certain libraries in our Jupyter notebook.

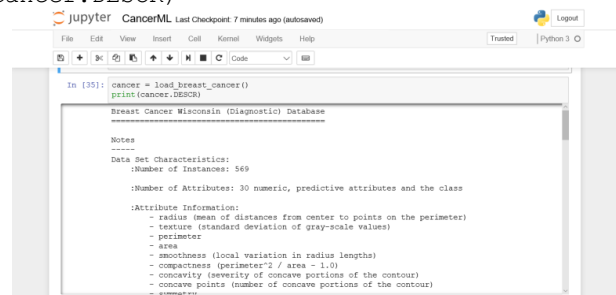
```
from sklearn.datasets import load_breast_cancer
from sklearn.neighbors import KNeighborsClassifier
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
%matplotlib inline
```

The above libraries are required for loading the dataset, importing our KNeighborsClassifier from Sklearn, and train_test_split function (it splits the arrays or matrices of data into training and testing subsets), and matplotlib (required to plot graphs and scatterplots in python).

If the above code runs, it means we have successfully loaded and imported all our libraries and dependencies.

Let us see our cancer dataset.

```
cancer = load_breast_cancer()
print(cancer.DESCR)
```



```

In [35]: cancer = load_breast_cancer()
print(cancer.DESCR)

Breast Cancer Wisconsin (Diagnostic) Database
-----
Notes
-----
Data Set Characteristics:
 :Number of Instances: 569

 :Number of Attributes: 30 numeric, predictive attributes and the class

:Attribute Information:
 - radius (mean of distances from center to points on the perimeter)
 - texture (standard deviation of gray-scale values)
 - perimeter
 - area
 - smoothness (local variation in radius lengths)
 - compactness (perimeter2 / area - 1.0)
 - concavity (severity of concave portions of the contour)
 - concave points (number of concave portions of the contour)
-----

```

Fig. 5. Description of the dataset

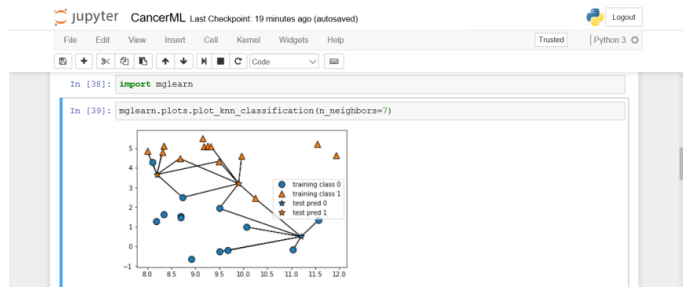


Fig. 8. Using mglearn

Here, n=7 refers to the number of neighbors we are using. Now we will fit our KNN classifier with the features to train.

```
X_train,X_test,y_train,
y_test=train_test_split(cancer.data,cancer.target,
stratify=cancer.target, random_state=42)
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
```

This will train our model, and we are ready to test it.

3.4 Using Logistic Regression

Now that we have successfully trained our KNN model, we are going to use another algorithm to compare and see which one yields better results.

Let us start by importing all the necessities for the logistic regression-

```
from sklearn.datasets import load_breast_cancer
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
```

Now let us train our model-

```
X_train,X_test,y_train,
y_test=train_test_split(cancer.data,cancer.target,
stratify=cancer.target, random_state=40)
logistic_reg = LogisticRegression()
logistic_reg.fit(X_train, y_train)
```

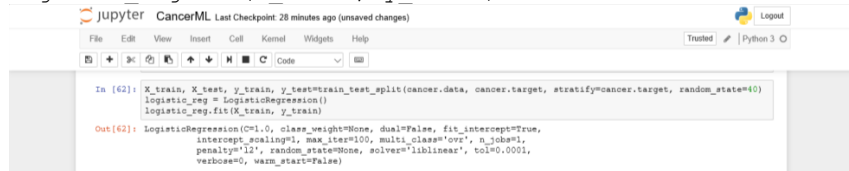


Fig. 9. Splitting the data for training and testing in logistic regression

Now we are done with our training process with both algorithms.

4 Result

After we are done with our training process, it is time to test and see how accurate results are being provided by our models.

4.1 Accuracy of KNN classifier

To check the accuracy of our KNN model, we have to use-

```
print('Accuracy of train set: {:.3f}'.format(knn.score(X_train,y_train)))
print('Accuracy of test set: {:.3f}'.format(knn.score(X_test,y_test)))
```

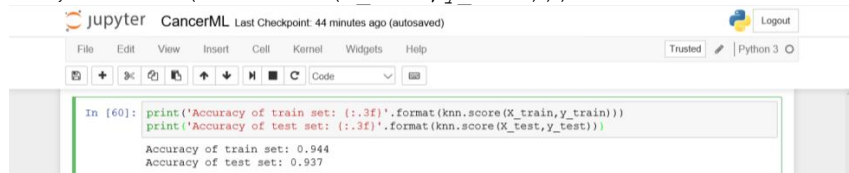


Fig. 10. Printing accuracy of KNN model

The KNN classifier shows a decent accuracy of 94.4% on the training set and 93.7% on the test set when the number of neighbors selected for clustering is 3. However, we need to make our classifier as accurate as it can be.

We can go ahead and check whether changing the number of neighbors will increase our accuracy. To check this, we will need matplotlib and we will use it to plot a line graph between number of neighbors and accuracy.

```
X_train,X_test,y_train,y_test=
train_test_split(cancer.data,cancer.target,
stratify=cancer.target, random_state=66)

train_accuracy=[]
test_accuracy=[]
neighbors_set = range(1,15)

for n_neighbors in neighbors_set:
    clf = KNeighborsClassifier(n_neighbors=n_neighbors)
    clf.fit(X_train,y_train)
    train_accuracy.append(clf.score(X_train, y_train))
    test_accuracy.append(clf.score(X_test, y_test))
    plt.plot(neighbors_set,train_accuracy, label='Training
set accuracy')
    plt.plot(neighbors_set,test_accuracy, label='Test set
accuracy')
    plt.ylabel('Accuracy')
    plt.xlabel('No. of Neighbors')
```



```
plt.legend()

jupyter CancerML Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 O
In [52]: X_train, X_test, y_train, y_test = train_test_split(cancer.data, cancer.target, stratify=cancer.target, random_state=6)
train_accuracy=[]
test_accuracy=[]
neighbor_set = range(1,15)
for n_neighbors in neighbor_set:
    clf = KNeighborsClassifier(n_neighbors=n_neighbors)
    clf.fit(X_train, y_train)
    train_accuracy.append(clf.score(X_train, y_train))
    test_accuracy.append(clf.score(X_test, y_test))
plt.plot(neighbor_set, train_accuracy, label='training set accuracy')
plt.plot(neighbor_set, test_accuracy, label='test set accuracy')
plt.xlabel('No. of Neighbors')
plt.legend()
```

Fig. 11. Code to get a graph for the number of neighbors giving the maximum accuracy

After running this, we will get a graph like this-

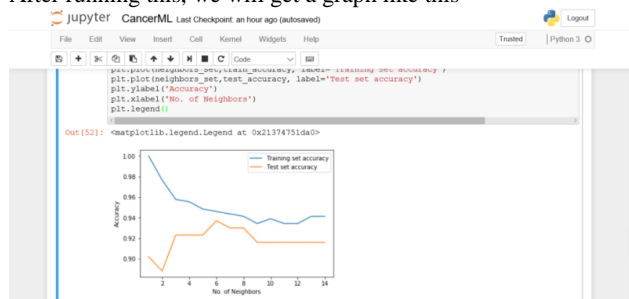


Fig. 12. Graph suggesting 6 as the most accurate number of neighbors

This graph suggests, that the test set accuracy will be most accurate when the number of neighbours is 6.

So now, let us change the number to 6 from 3 to see whether there is any increase in the accuracy.

```
jupyter CancerML Last Checkpoint: an hour ago (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 O
In [72]: print('Accuracy of train set: {:.6f}'.format(knn.score(X_train, y_train)))
print('Accuracy of test set: {:.6f}'.format(knn.score(X_test, y_test)))

Accuracy of train set: 0.936620
Accuracy of test set: 0.958042
```

Fig. 13. Accuracy with 6 neighbors

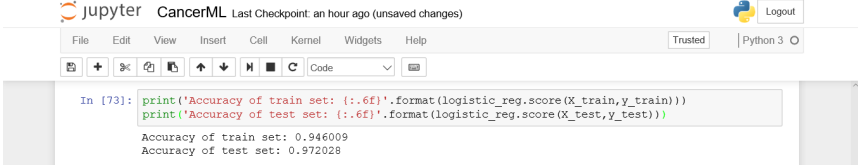
As we can see, the accuracy is now 93.66% on training set and 95.8% on test set. This clearly shows that there is an increase of 2.1% on our test set.

4.2 Accuracy of Logistic regression model

Now let us see how much accuracy our logistic regression algorithm will provide us with.

```
print('Accuracy of train set: {:.6f}'.format(logistic_reg.score(X_train, y_train)))
```

```
print('Accuracy of test set:
{:.6f}'.format(logistic_reg.score(X_test,y_test)))
```



The screenshot shows a Jupyter Notebook window titled 'CancerML Last Checkpoint: an hour ago (unsaved changes)'. The code cell contains the following Python code:

```
In [73]: print('Accuracy of train set: {:.6f}'.format(logistic_reg.score(X_train,y_train)))
print('Accuracy of test set: {:.6f}'.format(logistic_reg.score(X_test,y_test)))
```

The output of the code cell is:

```
Accuracy of train set: 0.946009
Accuracy of test set: 0.972028
```

Fig. 14. Accuracy with logistic regression

This shows that our model has a very impressive accuracy of 94.6% on training set, and more importantly, 97.2% on test set.

This shows that our model is much more accurate when we use logistic regression, rather than KNN classifier, which also yields decent results, but not as good as the former.

5 Discussion & Conclusion

From the program we can see that logistic regression provides us with more accuracy. There are other algorithms that can be used for the same, including Naïve Bayes, Support Vector Machines, Decision Trees etc. Even though, K-nearest-neighbors provide a very practical approach, they are dependent on the nearest neighbors for the prediction of target point. Logistic regression, on the other hand, works on the eradication of errors to draw the correct line between the points, on the basis of gradient descent.

In order to make more advancements in this field of study, we can try and focus on applying unsupervised learning as well, and also use image classification and regression techniques to make our models more expert in detecting the malignancy.

ML has still a long way to go if we are talking about completely replacing diagnostic tests like PET SCAN and biopsies. However, if successful, the detection will be very accurate and painless.

Acknowledgments. I wish to express my sincere gratitude to Dr Ankur Saxena for his utmost guidance and encouragement. Without his support and supervision, the completion of this paper would not have been possible.

References

1. D. Delen, G. Walker, and A. Kadam, "Predicting breast cancer survivability: a comparison of three data mining methods," *Artif. Intell. Med.*, vol. 34, pp. 113–127, 2005.
2. Siegel RL, Miller KD, Jemal A. *Cancer Statistics*, 2016. 2016;00(00):1-24. doi:10.3322/caac.21332.
3. "Globocan 2012 - Home." [Online]. Available: <http://globocan.iarc.fr/Default.aspx>. [Accessed: 28-Dec-2015].

4. Asri H, Mousannif H, Al Moatassime H, Noel T. Big data in healthcare: Challenges and opportunities. 2015 Int Conf Cloud Technol Appl. 2015:1-7. doi:10.1109/CloudTech.2015.7337020.
5. Quinlan JR. C4.5: Programs for Machine Learning.; 2014:302.
<https://books.google.com/books?hl=fr&lr=&id=b3ujBQAAQBAJ&pgis=1>. Accessed January 5, 2016.
6. Dataflog - Top 10 Data Mining Algorithms, Demystified.
<https://dataflog.com/read/top-10-data-mining-algorithmsdemystified/1144>. Accessed December 29, 2015.
7. V. Chaurasia and S. Pal, "Data Mining Techniques: To Predict and Resolve Breast Cancer Survivability," vol. 3, no. 1, pp. 10– 22, 2014.
8. Djebbari, A., Liu, Z., Phan, S., AND Famili, F. International journal of computational biology and drug design (ijcbdd). 21st Annual Conference on Neural Information Processing Systems (2008).
9. "UCI Machine Learning Repository: Breast Cancer Wisconsin (Original) Data Set." [Online]. Available:
<https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29>. [Accessed: 29-Dec-2015].
10. U.S. Cancer Statistics Working Group. United States Cancer Statistics: 1999–2008 Incidence and Mortality Web-based Report. Atlanta (GA): Department of Health and Human Services, Centers for Disease Control and Prevention, and National Cancer Institute; 2012.
11. A. Pradesh, "Analysis of Feature Selection with Classification : Breast Cancer Datasets," Indian J. Comput. Sci. Eng., vol. 2, no. 5, pp. 756–763, 2011.
12. Rui Xu, Xindi Cai, Donald C. , Wunsch II. Gene Expression Data for DLBCL Cancer Survival Prediction with A Combination of Machine Learning Technologies; In Proceedings of the IEEE International Conference on Medicine and Biology, 2005, p. 894-897.

