How to save files: NB1\_NB2\_NB3\_NB\_4 ( EACH SECTION SHOULD HAVE A ROADMAP./SUMMARY FOLDERS, JUST DATA AND FIGURES

README File

Necessity, who is the mother of invention*. – Plato*

Welcome to my GitHub repository on Using Predictive Analytics model to diagnose breast cancer.

## **Objective:**

The repository is a learning exercise to:

* Apply the fundamental concepts of machine learning from an available dataset
* Evaluate and interpret my results and justify my interpretation based on observed data set
* Create notebooks that serve as computational records and document my thought process.

The analysis is divided into four sections, saved in juypter notebooks in this repository

1. Identifying the problem and Data Sources
2. Exploratory Data Analysis
3. Pre-Processing the Data
4. Build model to predict whether breast cell tissue is malignant or Benign

**Notebook 1:** Identifying the problem and **Use external data sources for data enrichment.**

**Notebook goal**: **Identify the types of information contained in our data set**

In this notebook I used Python modules to import external data sets for the purpose of getting to know/familiarize myself with the data to get a good grasp of the data and think about how to handle the data in different ways.

**Notebook 2: Exploratory Data Analysis**

**Notebook goal:**  **Explore the variables to assess how they relate to the response variable**

In this notebook, I am getting familiar with the data using data exploration and visualization techniques using python libraries (Pandas, matplotlib, seaborn. Familiarity with the data is important which will provide useful knowledge for data pre-processing)

**Notebook 3: Pre-Processing the data**

**Notebook goal:**   **Find the most predictive features of the data and filter it so it will enhance the predictive power of the analytics model**

In this notebook I use feature selection to reduce high-dimension data, feature extraction and transformation for dimensionality reduction. This is essential in preparing the data before predictive models are developed.

**Notebook 4: Predictive model using Support Vector Machine (SVC)**

**Notebook goal: Construct predictive models** to predict the diagnosis of a breast tumor

In this notebook, I construct a predictive model using SVM machine learning algorithm to predict the diagnosis of a breast tumor. The diagnosis of a breast tumor is a binary variable (benign or malignant). I also evaluate the model using confusion matrix the receiver operating curves (ROC), which are essential in assessing and interpreting the fitted model.

**Notebook 1:** Identifying the problem and **Use external data sources for data enrichment.**

**Notebook goal**: **Identify the types of information contained in our data set**

**Identify the problem**

Breast cancer is the most common malignancy among women, accounting for nearly 1 in 3 cancers diagnosed among women in the United States, and it is the second leading cause of cancer death among women. Breast Cancer occurs as a results of abnormal growth of cells in the breast tissue, commonly referred to as a Tumor. A tumor does not mean cancer - tumors can be benign (not cancerous), pre-malignant (pre-cancerous), or malignant (cancerous). Tests such as MRI, mammogram, ultrasound and biopsy are commonly used to diagnose breast cancer performed.

**Expected outcome**

Given breast cancer results from breast fine needle aspiration (FNA) test (is a quick and simple procedure to perform, which removes some fluid or cells from a breast lesion or cyst (a lump, sore or swelling) with a fine needle similar to a blood sample needle). Since this build a model that can classify a breast cancer tumor using two training classification:

1= Malignant (Cancerous) - Present

0= Benign (Not Cancerous) -Absent

**Objective**

Since the labels in the data are discrete, the predication falls into two categories, (i.e. Malignant or benign). In machine learning this is a classification problem. Thus, the goal is to classify whether the breast cancer is benign or malignant and predict the recurrence and non-recurrence of malignant cases after a certain period. To achieve this we have used machine learning classification methods to fit a function that can predict the discrete class of new input

**ML Classification Method applied:**

SVM - Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification and regression analysis. It segregates two classes using a hyper plane.

**Identify data sources**

The Breast Cancer datasets is available [machine learning repository](https://archive.ics.uci.edu/ml/index.html) maintained by the University of California, Irvine.

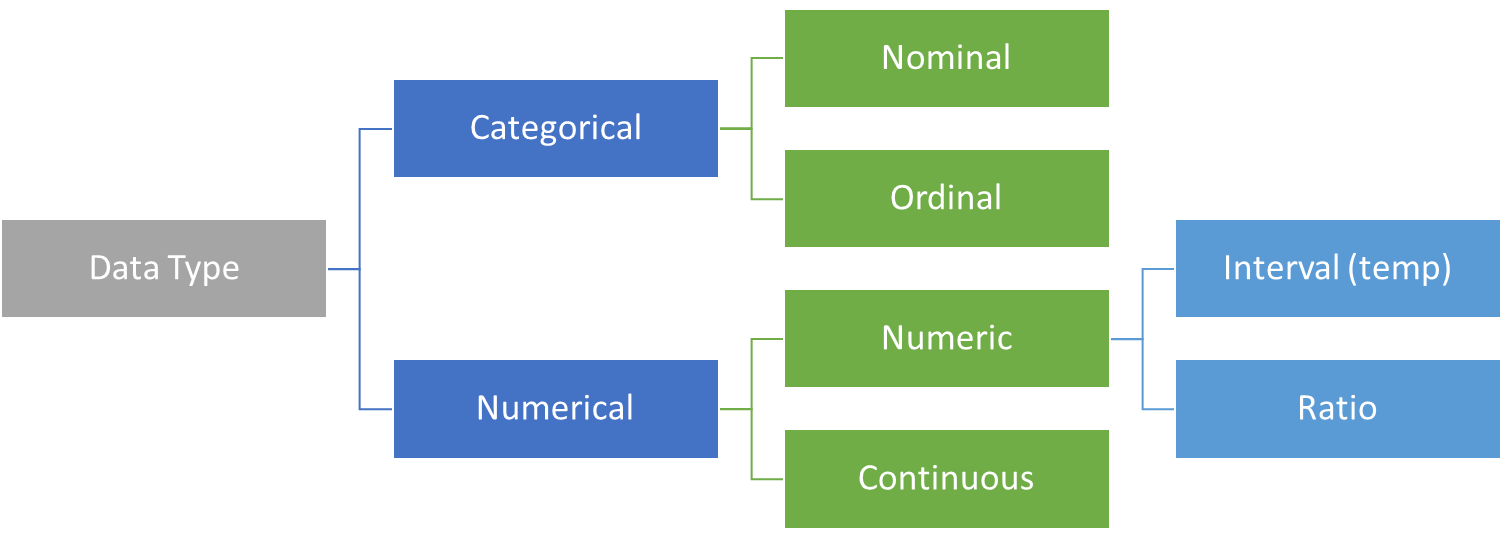
**NB\_2 Getting Started:**

1. **Load libraries and set options –** Import necessary libraries and offer explanation on each
2. **Obtaining Data- Data Set**

The next step is to use the UCI Machine Learning Repository to load the Breast Cancer dataset that had 14 attributes and 48,842 instances with which to build a model. The hypothesis is that it is possible predict the recurrence and non-recurrence of malignant cases after a certain period, based upon breast cancer tumor attributes is benign or malignant. The suggested task is to build a binary classifier that can determine from the Breast Cancer dataset information whether or not a tumor is benign or malignant. .

For this tutorial, we selected a [Census Income](http://archive.ics.uci.edu/ml/datasets/Census+Income) dataset that had 14 attributes and 48,842 instances.

**DATA TYPES**

****

***Structured data*** is data which is a form of data which has a high degree or organization such as numerical or categorical data. Temperature, phone numbers, gender are examples of structured data.

***Unstructured data* is** data in a form which doesn’t explicitly have structure we are used to. Examples of unstructured data are photos, images, audio, language text and many others. There is an emerging field called [Deep Learning](http://datascienceguide.github.io/deep-learning/) which is using a specialized set of algorithms which perform well with unstructured data.

 The two common types of structured we commonly deal with are categorical variables (which have a finite set of values) or numerical values (which are continuous).

***Categorical Variables:*** Categorical variables can also be nominal or ordinal.

* ***Nominal*** data has no intrinsic ordering to the categories. For example gender (Male, Female, Other) has no specific ordering.
* ***Ordinal*** data as clear ordering such as three settings on a toaster (high medium and low). A frequency table (count of each category) is the common statistic for describing categorical data of each variable, and a bar chart or a waffle chart (shown below) are two visualizations which can be used.

***Numeric Variables****:* Numeric or continuous variables can be any value within a finite or infinite interval (temperature, height, weight.

There are two types of numeric variables are interval and ratios.

* Interval variables have numeric scales and the same interpretation throughout the scale, but do not have an absolute zero. eg temperature in Fahrenheit or Celcius can meaningfully be subtracted or added (difference between 10 degrees and 20 degrees is the same difference as 40 to 50 degrees) but cannot be multiplied. For example, a day which is twice as hot may not be twice the temperature
* ratio scale of measurement is the most informative scale. It is an interval scale with the additional property that its zero position indicates the absence of the quantity being measured.

***Transforming Data***: There are a couple of techniques:

* ***Binning (Numerical to Categorical) -Binning***
* ***Encoding (Categorical to Numerical); Continuization***
  + Encoding or continuation is the transformation of categorical variables to binary or numerical counterparts. An example is to treat male or female for gender as 1 or 0. Categorical variables must be encoded in many modeling methods (e.g., linear regression, SVM, neural networks)
  + Two main types of encoding are Binary and Target-based (<http://www.saedsayad.com/encoding.htm>)
  + Binary encoding is a special case of encoding where the value is set to a 0 or 1 to indicate absence or presence of a category.
  + One hot encoding is a special case where multiple categories are each binary encoded. Given we have k categories, this will create k extra features (thus increasing the dimensionality of the data)

**Notebook 2: Exploratory Data Analysis**

**Now that we have a good intuitive sense of the data,** Next step involves taking a closer look at attributes and data values. In this section, I am getting familiar with the data, which will provide useful knowledge for data pre-processing.

**Objectives of Data Exploration**

Exploratory data analysis (EDA) is a very important step which takes place after [**feature engineering**](http://datascienceguide.github.io/feature-engineering) and **acquiring data** and it should be done before any modeling. This is because it is very important for a data scientist to be able to understand the nature of the data without making assumptions. The results of data exploration can be extremely useful in grasping the structure of the data, the distribution of the values, and the presence of extreme values and interrelationships within the data set

http://onlinestatbook.com/2/introduction/levels\_of\_measurement.html

The purpose of EDA is:

* to use summary statistics and visualizations to better understand data,
* find clues about the tendencies of the data, its quality and to formulate assumptions and the hypothesis of our analysis
* For data preprocessing to be successful, it is essential to have an overall picture of your data

Basic statistical descriptions can be used to identify properties of the data and highlight which data values should be treated as noise or outliers.

* Descriptive statistics is the process of condensing key characteristics of the data set into simple numeric metrics. Some of the common metrics used are **mean, standard deviation, and correlati**on.
* Visualization is the process of projecting the data, or parts of it, into Cartesian space or into abstract images. In the data mining process, data exploration is leveraged in many different steps including preprocessing, modeling, and interpretation of results.
* https://solomonmessing.wordpress.com/2012/03/04/visualization-series-insight-from-cleveland-and-tufte-on-plotting-numeric-data-by-groups/

**Descriptive/Summary Statistics.**

Summary statistics are measurements meant to describe data. In the field of descriptive statistics, there are many summary measurements, (<http://www.saedsayad.com/numerical_variables.htm>)



**Visual Feature Analysis (https://www.dashingd3js.com/why-data-visualizations)**

Graphic displays of basic statistical descriptions are helpful for the visual inspection of data, which is useful for data preprocessing. These include *quantile plots, quantile–quantile plots, histograms*, and scatter plots. Such graphs. The first three of these show univariate distributions (i.e., data for one attribute), while scatter plots show bivariate distributions (i.e., involving two attributes).

Examples of visualizations for

* numeric data are line charts with error bars, histograms, box and whisker plots
* categorical data bar charts and waffle charts, histograms
* Bivariate data are scatter charts or combination charts. The tutorial on exploratory data analysis goes over many of these visualizations.

**Univariate data (One variable)**

The two visualizations used to describe univariate (1 variable) data is the box plot and the histogram. The box plot can be used to show the minimum, maximum, mean, median, quantiles and range.

**Bivariate data (Two Variables)**

When plotting the relation between two variables, one can use a **scatter plot**. A scatter plot is one of the most effective graphical methods for determining if there appears to be a relationship, pattern, or trend between two numeric attributes. To construct a scatter plot.

The scatter plot is a useful method for providing a first look at bivariate data to see clusters of points and outliers, or to explore the possibility of correlation relationships. Two attributes, X, and Y, are correlated if one attribute implies the other. Correlations can be positive, negative, or null (uncorrelated)



[**http://blog.districtdatalabs.com/an-introduction-to-machine-learning-with-python**](http://blog.districtdatalabs.com/an-introduction-to-machine-learning-with-python)

[**http://blog.districtdatalabs.com/data-exploration-with-python-3**](http://blog.districtdatalabs.com/data-exploration-with-python-3)

[**http://blog.districtdatalabs.com/data-exploration-with-python-2**](http://blog.districtdatalabs.com/data-exploration-with-python-2)

**http://blog.districtdatalabs.com/data-exploration-with-python-1**

**http://proquest.safaribooksonline.com.ezproxy.library.tamu.edu/book/databases/business-intelligence/9780128014608/chapter-3dot-data-exploration/s0010\_html\_3#X2ludGVybmFsX0h0bWxWaWV3P3htbGlkPTk3ODAxMjgwMTQ2MDglMkZzMDAzNV9odG1sXzMmcXVlcnk9KChQeXRob24lMjBNYWNoaW5lJTIwTGVhcm5pbmclMjBCbHVlcHJpbnRzKSk=**

[**https://github.com/CommerceDataService/tutorial-predicting-income/blob/master/predicting\_income\_with\_census\_data\_pt1.md**](https://github.com/CommerceDataService/tutorial-predicting-income/blob/master/predicting_income_with_census_data_pt1.md)

Following are a list of libraries, you will need for any scientific computations and data analysis:

* **NumPy**stands for Numerical Python. The most powerful feature of NumPy is n-dimensional array. This library also contains basic linear algebra functions, Fourier transforms,  advanced random number capabilities and tools for integration with other low level languages like Fortran, C and C++
* **SciPy** stands for Scientific Python. SciPy is built on NumPy. It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.
* **Matplotlib** for plotting vast variety of graphs, starting from histograms to line plots to heat plots.. You can use Pylab feature in ipython notebook (ipython notebook –pylab = inline) to use these plotting features inline. If you ignore the inline option, then pylab converts ipython environment to an environment, very similar to Matlab. You can also use Latex commands to add math to your plot.
* **Pandas** for structured data operations and manipulations. It is extensively used for data munging and preparation. Pandas were added relatively recently to Python and have been instrumental in boosting Python’s usage in data scientist community.
* **Scikit Learn** for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of effiecient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.
* **Statsmodels** for statistical modeling. Statsmodels is a Python module that allows users to explore data, estimate statistical models, and perform statistical tests. An extensive list of descriptive statistics, statistical tests, plotting functions, and result statistics are available for different types of data and each estimator.
* **Seaborn** for statistical data visualization. Seaborn is a library for making attractive and informative statistical graphics in Python. It is based on matplotlib. Seaborn aims to make visualization a central part of exploring and understanding data.
* **Bokeh** for creating interactive plots, dashboards and data applications on modern web-browsers. It empowers the user to generate elegant and concise graphics in the style of D3.js. Moreover, it has the capability of high-performance interactivity over very large or streaming datasets.
* **Blaze** for extending the capability of Numpy and Pandas to distributed and streaming datasets. It can be used to access data from a multitude of sources including Bcolz, MongoDB, SQLAlchemy, Apache Spark, PyTables, etc. Together with Bokeh, Blaze can act as a very powerful tool for creating effective visualizations and dashboards on huge chunks of data.
* **Scrapy** for web crawling. It is a very useful framework for getting specific patterns of data. It has the capability to start at a website home url and then dig through web-pages within the website to gather information.
* **SymPy** for symbolic computation. It has wide-ranging capabilities from basic symbolic arithmetic to calculus, algebra, discrete mathematics and quantum physics. Another useful feature is the capability of formatting the result of the computations as LaTeX code.
* **Requests** for accessing the web. It works similar to the standard python library urllib2 but is much easier to code. You will find subtle differences with urllib2 but for beginners, Requests might be more convenient.
* **os** for Operating system and file operations
* **networkx**and **igraph** for graph based data manipulations
* regular expressions for finding patterns in text data
* **BeautifulSoup** for scrapping web. It is inferior to Scrapy as it will extract information from just a single webpage in a run.

Now that we are familiar with Python fundamentals and additional libraries, lets take a deep dive into problem solving through Python. Yes I mean making a predictive model! In the process, we use some powerful libraries and also come across the next level of data structures. We will take you through the 3 key phases: