

Machine Learning Project
AirBnb – Predicting New User Bookings

Write-up

Shraddha Dangare

Smitha Kannanaikkal

Manish Nanwani

PGP Sept 2017

Batch DF1709-CM

Background:

The dataset which we are working upon is the AirBnb Dataset, obtained from Kaggle, containing the information about new user activity recorded between a period of 2010 to 2014. The aim is to develop a classifier using this data to try and learn the destination countries selected by the people, and after having developed a model, we try to predict the destination country that a new user might select to travel to, and this new user data is collected after the time period of 2014.

1.Dataset Description:

The dataset provided by Airbnb contains a list of users along with their demographics, web session records, and some summary statistics. The whole dataset contains 5 csv files: train-users, test-users, sessions, countries, age-gender-bkts.

1) The train-users files contains 171239 training examples with 16 features, having information related to first booking done, account created, timestamp of first activity done, gender, age, sign up method, affiliate channel, ie, what kind of paid marketing, and affiliate provider used, ie, where the marketing is, first affiliate tracked, ie, whats the first marketing the user interacted with before the signing up.

2) The test-users have 43673 items and 15 properties. The values of country-destination are missing and that is the value we are asked to predict. The training and test sets are split by dates. In the test set, we are expected to predict coountry destination of all the new users with first activities after 4/1/2014.

3) sessions: The sessions file is the web sessions log records for users. The sessions file contains 5600850 examples and 6 properties: user-id, action, action-type, action-detail, device-type, secs-elapsed. There are actually 74610 different users in the file.

4) countries: The countries file contains statistics of destination countries in this dataset and their geometric information. It has information for 10 countries and their 7 different properties, such as longitude and latitude.

5) age-gender-bkts: This file contains statistics of users' age group, gender, country of destination. It consists of 420 examples and 5 properties.

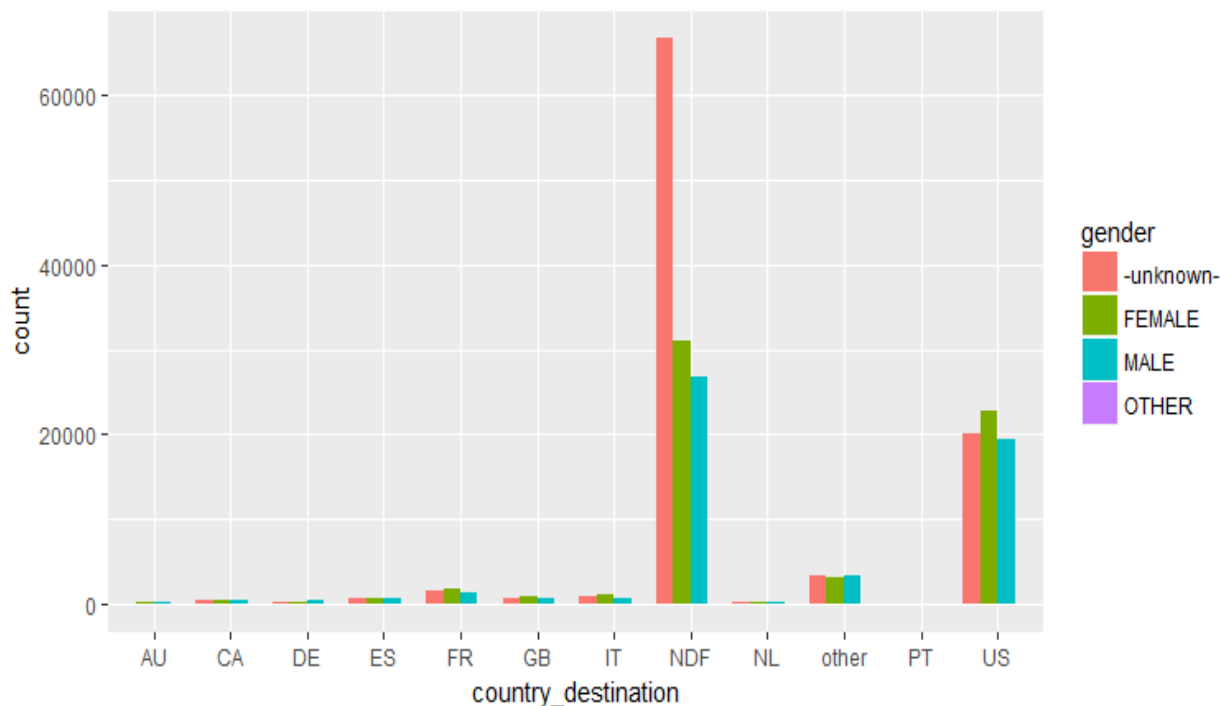
2. Exploratory Data Analysis:

After running summary we can see that there are NA /-unknown- values in the following attribute-

- Gender- '-unknown-'
- Age- NA
- First_affiliate_tracked- NA

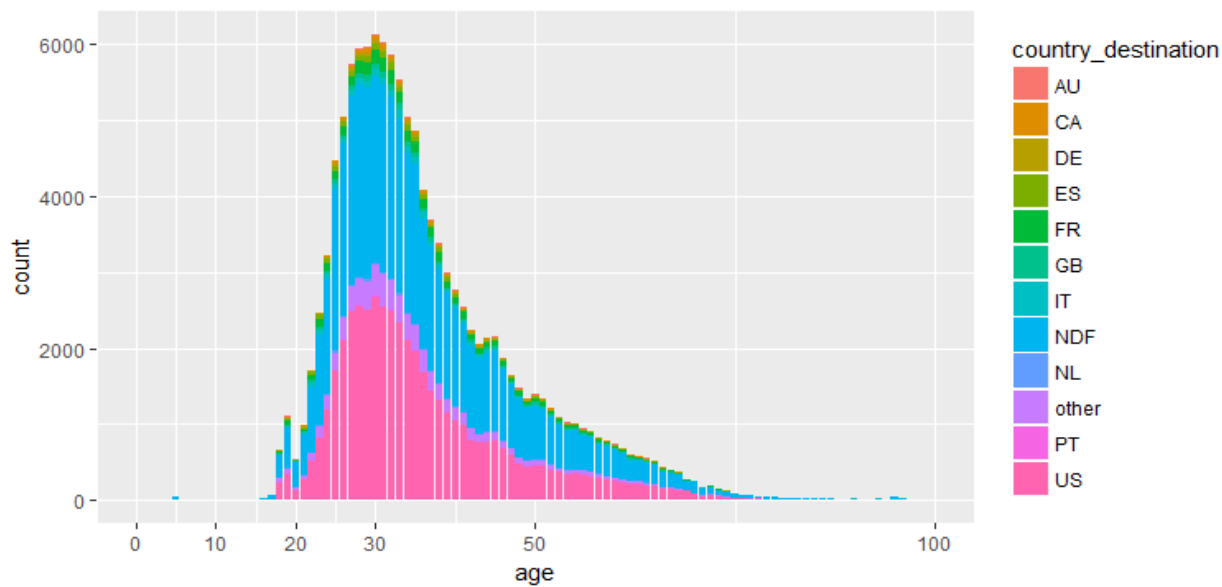
I. Identify NA's and handle NA-

1. Missing value Imputation of Gender



With the help of the graph, we impute the NA values for Gender as per the proportional distribution of Gender across the individual destination country.

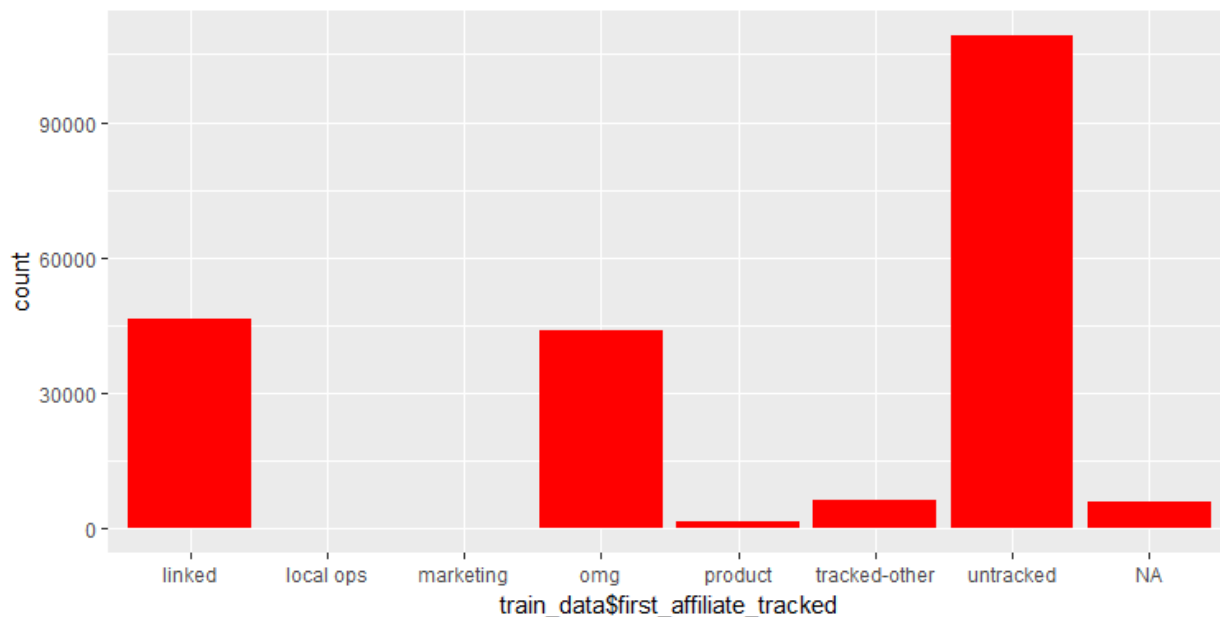
2. Missing Value Imputation of Age



From the summary of the age, we observed unusual values of age, so we capped the age limits, with the lower limits at 15 years and upper limit at 100 years.

Also we had certain values where there were birth years entered instead of the age, so we computed the age for those values as per the difference in the years, when the data was collected.

3. Missing value imputation of first Affiliate tracked



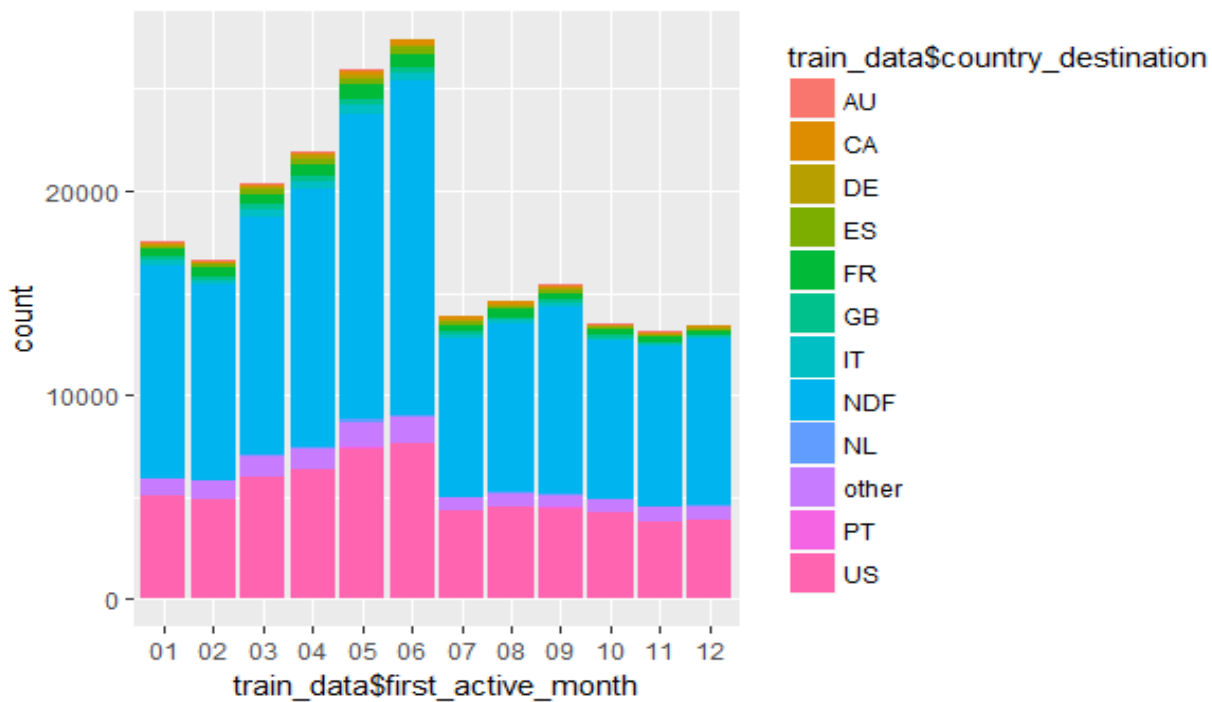
For this feature, we imputed NA values as untracked level.

II. Feature Engineering:

1. first_active_month:

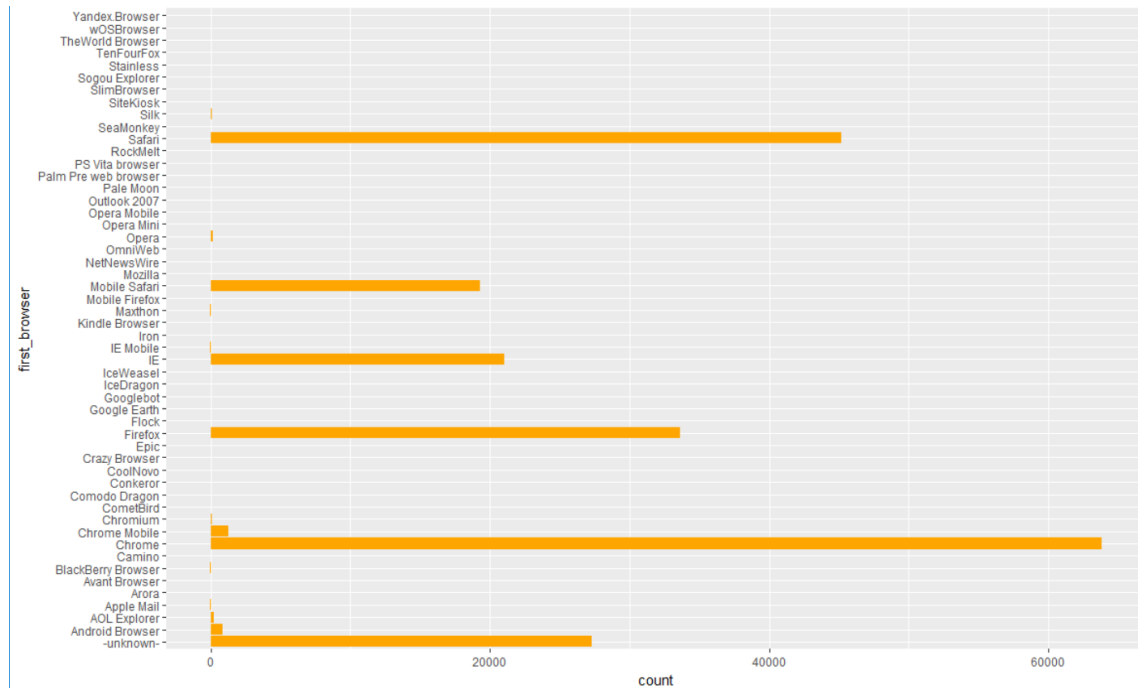
We extracted months from the timestamp.

```
timestamp_first_active
Min.      :2.009e+13
1st Qu.   :2.012e+13
Median    :2.013e+13
Mean      :2.013e+13
3rd Qu.   :2.014e+13
Max.      :2.014e+13
```



From the Graph, we binned the Months into two broad groups, summer/spring and fall/winter.

2. First browser



There are more than 50 categories of browsers. Although the majority use only 6, therefore we will club all the minor browsers into the “other” category.

3. Merging multiple data sources

age-gender-bkts-

We used the age gender csv file, which has the age buckets for all the countries destination, gender-wise. So we use the mean value for that destination country, having the maximum population coming from that age bucket, to impute into the Na values in the train data for each destination country.

sessions-

The session csv file has the log of all the activities for every individual id, so for every id, we combined all the activities together and extracted the total seconds elapsed by each user id for a given session.

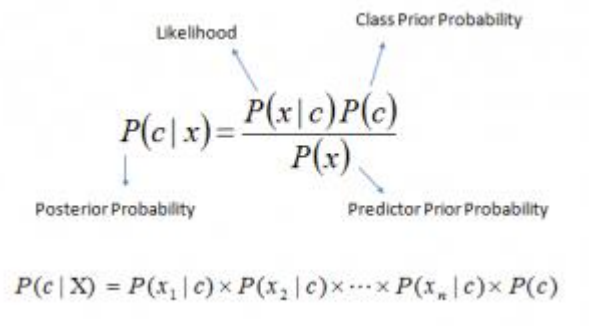
4. Algorithm Used:

1. Naïve Bayes:

Naïve Bayes is a classification technique based on naïve bayes with an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. For example, a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, all of these properties independently contribute to the probability that this fruit is an apple and that is why it is known as 'Naive'.

Naive Bayes model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability $P(c|x)$ from $P(c)$, $P(x)$ and $P(x|c)$. Look at the equation below:



The diagram shows the Bayes' Theorem equation $P(c|x) = \frac{P(x|c)P(c)}{P(x)}$ with four labels and arrows: 'Likelihood' points to $P(x|c)$, 'Class Prior Probability' points to $P(c)$, 'Posterior Probability' points to $P(c|x)$, and 'Predictor Prior Probability' points to $P(x)$. Below the equation is the expanded formula: $P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$.

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Posterior Probability Likelihood Class Prior Probability Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

- $P(c|x)$ is the posterior probability of *class* (c , *target*) given *predictor* (x , *attributes*).
- $P(c)$ is the prior probability of *class*.
- $P(x|c)$ is the likelihood which is the probability of *predictor* given *class*.
- $P(x)$ is the prior probability of *predictor*.

Pros and Cons of Naive Bayes :

Pros:

- It is easy and fast to predict class of test data set. It also perform well in multi class prediction
- When assumption of independence holds, a Naive Bayes classifier performs better compare to other models like logistic regression and you need less training data.
- It perform well in case of categorical input variables compared to numerical variable(s). For numerical variable, normal distribution is assumed (bell curve, which is a strong assumption).

Cons:

- If categorical variable has a category (in test data set), which was not observed in training data set, then model will assign a 0 (zero) probability and will be unable to make a prediction. This is often known as “Zero Frequency”. To solve this, we can use the smoothing technique. One of the simplest smoothing techniques is called Laplace estimation.
- On the other side naive Bayes is also known as a bad estimator, so the probability outputs are not to be taken too seriously.
- Another limitation of Naive Bayes is the assumption of independent predictors. In real life, it is almost impossible that we get a set of predictors which are completely independent.

2. Logistic Regression

Logistic Regression is a classification algorithm. It is used to predict a binary outcome (1 / 0, Yes / No, True / False) given a set of independent variables. To represent binary / categorical outcome, we use dummy variables. You can also think of logistic regression as a special case of linear regression when the outcome variable is categorical, where we are using log of odds as dependent variable. In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function.

Derivation of Logistic Regression Equation

The fundamental equation of generalized linear model is:

$$g(E(y)) = \alpha + \beta x_1 + \gamma x_2$$

Here, $g()$ is the link function, $E(y)$ is the expectation of target variable and $\alpha + \beta x_1 + \gamma x_2$ is the linear predictor (α, β, γ to be predicted). The role of link function is to ‘link’ the expectation of y to linear predictor.

Important Points

1. GLM does not assume a linear relationship between dependent and independent variables. However, it assumes a linear relationship between link function and independent variables in logit model.
2. The dependent variable need not to be normally distributed.
3. It does not use OLS (Ordinary Least Square) for parameter estimation. Instead, it uses maximum likelihood estimation (MLE).
4. Errors need to be independent but not normally distributed.

Let's understand it further using an example:

We are provided a sample of 1000 customers. We need to predict the probability whether a customer will buy (y) a particular magazine or not. As you can see, we've a categorical outcome variable, we'll use logistic regression.

To start with logistic regression, I'll first write the simple linear regression equation with dependent variable enclosed in a link function:

$$g(y) = \beta_0 + \beta(\text{Age}) \quad \text{---- (a)}$$

Note: For ease of understanding, I've considered 'Age' as independent variable.

In logistic regression, we are only concerned about the probability of outcome dependent variable (success or failure). As described above, g() is the link function. This function is established using two things: Probability of Success(p) and Probability of Failure(1-p). p should meet following criteria:

1. It must always be positive (since $p \geq 0$)
2. It must always be less than equals to 1 (since $p \leq 1$)

Now, we'll simply satisfy these 2 conditions and get to the core of logistic regression. To establish link function, we'll denote g() with 'p' initially and eventually end up deriving this function.

Since probability must always be positive, we'll put the linear equation in exponential form. For any value of slope and dependent variable, exponent of this equation will never be negative.

$$p = \exp(\beta_0 + \beta(\text{Age})) = e^{(\beta_0 + \beta(\text{Age}))} \quad \text{----- (b)}$$

To make the probability less than 1, we must divide p by a number greater than p. This can simply be done by:

$$p = \exp(\beta_0 + \beta(\text{Age})) / \exp(\beta_0 + \beta(\text{Age})) + 1 = e^{(\beta_0 + \beta(\text{Age}))} / e^{(\beta_0 + \beta(\text{Age}))} + 1 \quad \text{----- (c)}$$

Using (a), (b) and (c), we can redefine the probability as:

$$p = e^y / 1 + e^y \quad \text{--- (d)}$$

where p is the probability of success. *This (d) is the Logit Function*

If p is the probability of success, $1-p$ will be the probability of failure which can be written as:

$$q = 1 - p = 1 - (e^y / 1 + e^y) \quad \text{--- (e)}$$

where q is the probability of failure

On dividing, (d) / (e), we get,

$$\frac{p}{1-p} = e^y$$

After taking log on both side, we get,

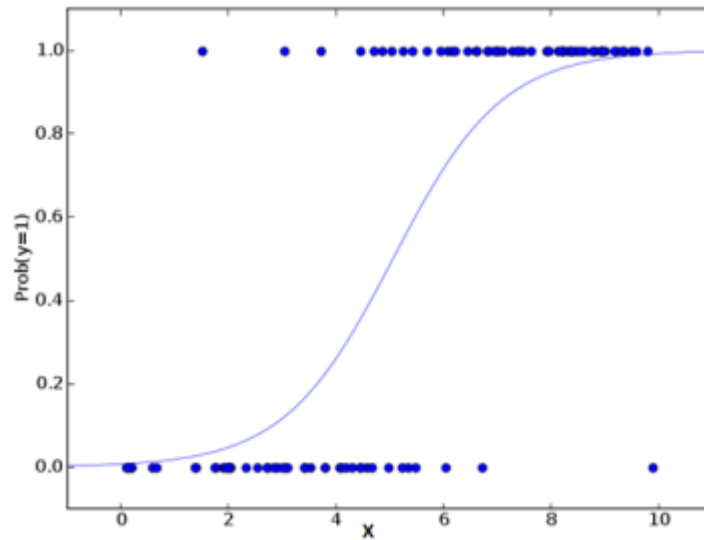
$$\log \left(\frac{p}{1-p} \right) = y$$

$\log(p/1-p)$ is the link function. Logarithmic transformation on the outcome variable allows us to model a non-linear association in a linear way.

After substituting value of y , we'll get:

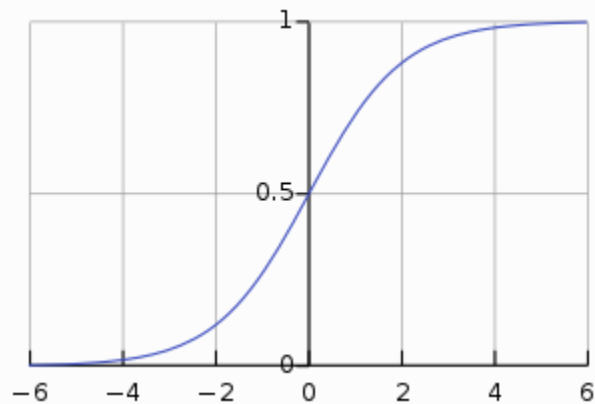
$$\log \left(\frac{p}{1-p} \right) = \beta_0 + \beta(\text{Age})$$

This is the equation used in Logistic Regression. Here $(p/1-p)$ is the odd ratio. Whenever the log of odd ratio is found to be positive, the probability of success is always more than 50%. A typical logistic model plot is shown below. You can see probability never goes below 0 and above 1.



This is a **binary classification** problem as each tumor is either ($y = 1$) or ($y = 0$). These are the 2 classes here. This probability function is the '**Sigmoid Function**' which is :

In order to map predicted values to probabilities, we use the sigmoid function. The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.



$$f(x) = \frac{1}{1 + e^{-(x)}}$$

- $F(x)$ = output between 0 and 1 (probability estimate)

3. Linear Discriminant Analysis

Logistic regression is a classification algorithm traditionally limited to only two-class classification problems.

If you have more than two classes then Linear Discriminant Analysis is the preferred linear classification technique.

In this post you will discover the Linear Discriminant Analysis (LDA) algorithm for classification predictive modeling problems.

Limitations of Logistic Regression

Logistic regression is a simple and powerful linear classification algorithm.

Two-Class Problems. Logistic regression is intended for two-class or binary classification problems. It can be extended for multi-class classification, but is rarely used for this purpose.

- **Unstable With Well Separated Classes.** Logistic regression can become unstable when the classes are well separated.
- **Unstable With Few Examples.** Logistic regression can become unstable when there are few examples from which to estimate the parameters.

Linear Discriminant Analysis does address each of these points and is the go-to linear method for multi-class classification problems. Even with binary-classification problems, it is a good idea to try both logistic regression and linear discriminant analysis.

Representation of LDA Models

The representation of LDA is straight forward.

It consists of statistical properties of your data, calculated for each class. For a single input variable (x) this is the mean and the variance of the variable for each class. For multiple variables, this is the same properties calculated over the multivariate Gaussian, namely the means and the covariance matrix.

Learning LDA Models

LDA makes some simplifying assumptions about your data:

1. That your data is Gaussian, that each variable is is shaped like a bell curve when plotted.
 2. That each attribute has the same variance, that values of each variable vary around the mean by the same amount on average.
- With these assumptions, the LDA model estimates the mean and variance from your data for each class. It is easy to think about this in the univariate (single input variable) case with two classes.

The mean (μ) value of each input (x) for each class (k) can be estimated in the normal way by dividing the sum of values by the total number of values.

$$\mu_k = 1/n_k * \sum(x)$$

Where μ_k is the mean value of x for the class k , n_k is the number of instances with class k . The variance is calculated across all classes as the average squared difference of each value from the mean.

$$\sigma^2 = 1 / (n-K) * \sum((x - \mu)^2)$$

Where σ^2 is the variance across all inputs (x), n is the number of instances, K is the number of classes and μ is the mean for input x .

How to Prepare Data for LDA

This section lists some suggestions you may consider when preparing your data for use with LDA.

- **Classification Problems.** This might go without saying, but LDA is intended for classification problems where the output variable is categorical. LDA supports both binary and multi-class classification.
- **Gaussian Distribution.** The standard implementation of the model assumes a Gaussian distribution of the input variables. Consider reviewing the univariate distributions of each attribute and using transforms to make them more Gaussian-looking (e.g. log and root for exponential distributions and Box-Cox for skewed distributions).
- **Remove Outliers.** Consider removing outliers from your data. These can skew the basic statistics used to separate classes in LDA such the mean and the standard deviation.
- **Same Variance.** LDA assumes that each input variable has the same variance. It is almost always a good idea to standardize your data before using LDA so that it has a mean of 0 and a standard deviation of 1.

Extensions to LDA

Linear Discriminant Analysis is a simple and effective method for classification. Because it is simple and so well understood, there are many extensions and variations to the method. Some popular extensions include:

- **Quadratic Discriminant Analysis (QDA):** Each class uses its own estimate of variance (or covariance when there are multiple input variables).
- **Flexible Discriminant Analysis (FDA):** Where non-linear combinations of inputs is used such as splines.
- **Regularized Discriminant Analysis (RDA):** Introduces regularization into the estimate of the variance (actually covariance), moderating the influence of different variables on LDA.

The original development was called the Linear Discriminant or Fisher's Discriminant Analysis. The multi-class version was referred to Multiple Discriminant Analysis. These are all simply referred to as Linear Discriminant Analysis now.

Inference:

-Algorithm Comparison

Naïve Bayes

Accuracy:0.532

Multinomial LR

Accuracy:0.606

LDA

Accuracy:0.67

We observe that we get better accuracy by LDA over Multinomial Logistic Regression and Naïve-Bayes.

Naïve-Bayes is generally used as base algorithm to define a benchmark of minimum accuracy, and the same can be observed from the accuracy results obtained.

