Data Science Made Easy – Level 3

Using python™

Ramadurai Seshadri

August 2016

Ramadurai.seshadri@wipro.com

What You will Learn Today

Agenda for “Data Science Made Easy Using Python – Level 3”

1. What is Data Science and why is it important now?
2. What are the tools used by a Data Scientist for perform predictive analytics
3. What are the steps involved in building a predictive model
   - ✓ Hypothesis Generation
   - ✓ Data Preparation
   - ✓ Feature Engineering
   - ✓ Model Preprocessing
   - ➢Model Building
   - ➢Model Evaluation
4. How does a Data Scientist use Python
   - ✓ Using libraries - Pandas, Numpy, Matplotlib, Scipy, Statsmodels, Sci-Kit Learn
   - ✓ Machine Learning Algorithms

✓ Covered in previous class

Problem Definition and Hypotheses Generation

What is the business problem I am trying to solve?

1. Problem Statement example

   Predicting Revenues for a Chain Store. A Chain Store has revenue data for 2013 across 10 different stores and 1559 categories of Items. Their problem is they want to forecast revenues by store and by item for the next year. Can we design a prediction algorithm to help the Store?

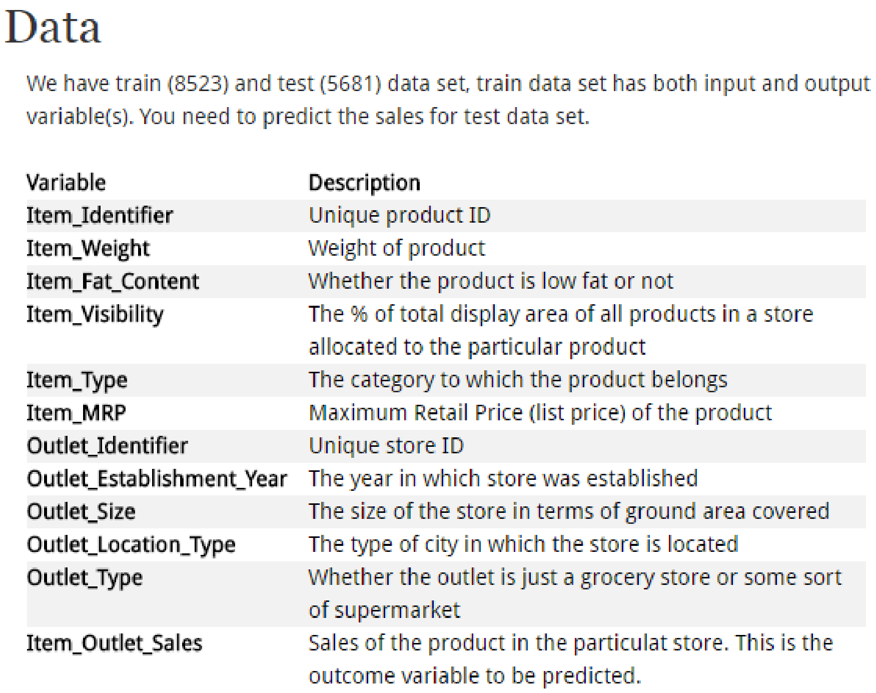
2. What is the Outcome I am trying to achieve or predict?

   - Outcome variable example

   Your task is to predict the Item Outlet Sales for the next year across Outlets and Items.

3. What kind of data do I have?

   - Store and Item Sales data*



Data

We have train (8523) and test (5681) data set, train data set has both input and output variable(s). You need to predict the sales for test data set.

Variable	Description
Item_Identifier	Unique product ID
Item_Weight	Weight of product
Item_Fat_Content	Whether the product is low fat or not
Item_Visibility	The % of total display area of all products in a store allocated to the particular product
Item_Type	The category to which the product belongs
Item_MRP	Maximum Retail Price (list price) of the product
Outlet_Identifier	Unique store ID
Outlet_Establishment_Year	The year in which store was established
Outlet_Size	The size of the store in terms of ground area covered
Outlet_Location_Type	The type of city in which the store is located
Outlet_Type	Whether the outlet is just a grocery store or some sort of supermarket
Item_Outlet_Sales	Sales of the product in the particulat store. This is the outcome variable to be predicted.

* source: AnalyticsVidhya.com

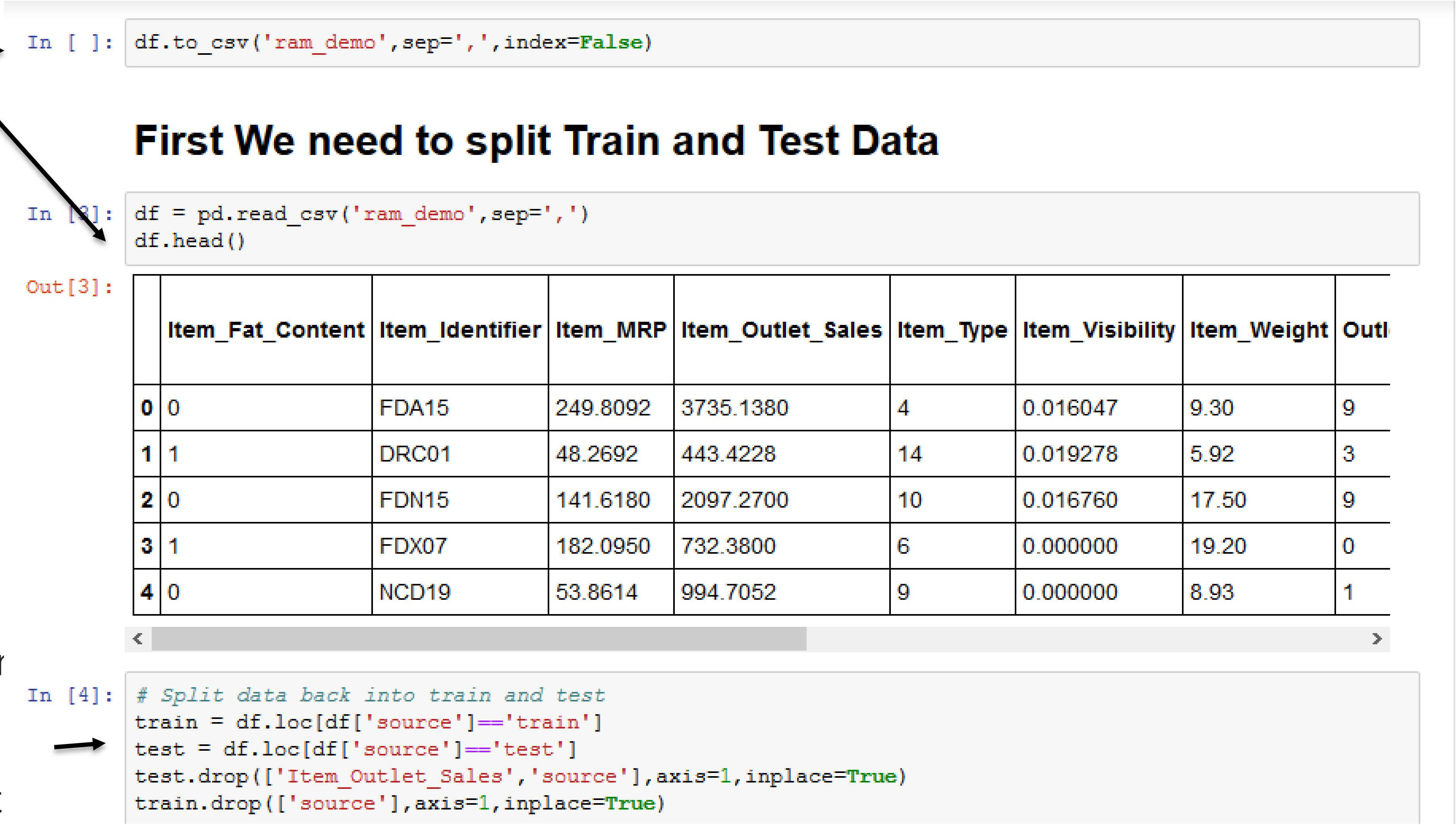
What we have done so far

Review the data set prepared for model building

Load Data from CSV format into a Python Pandas DataFrame

We have train (100) and test (50) data sets. Hence we must load them into separate Pandas Data Frames.

Once loaded into Pandas Data Frames, we can split them into Model Ready Train data set to "train" the model and then "test" the model using the Model Ready Test data set.



```
In [ ]: df.to_csv('ram_demo',sep=',',index=False)
```

First We need to split Train and Test Data

```
In [3]: df = pd.read_csv('ram_demo',sep=',')
        df.head()
```

Out[3]:

	Item_Fat_Content	Item_Identifier	Item_MRP	Item_Outlet_Sales	Item_Type	Item_Visibility	Item_Weight	Outl
0	0	FDA15	249.8092	3735.1380	4	0.016047	9.30	9
1	1	DRC01	48.2692	443.4228	14	0.019278	5.92	3
2	0	FDN15	141.6180	2097.2700	10	0.016760	17.50	9
3	1	FDX07	182.0950	732.3800	6	0.000000	19.20	0
4	0	NCD19	53.8614	994.7052	9	0.000000	8.93	1

```
In [4]: # Split data back into train and test
        train = df.loc[df['source']=='train']
        test = df.loc[df['source']=='test']
        test.drop(['Item_Outlet_Sales','source'],axis=1,inplace=True)
        train.drop(['source'],axis=1,inplace=True)
```

Model Building

Process of deriving Insights from Data

Understanding Basics

Some Terms every Data Scientist should be Familiar with

Regression

If the target is a continuous value, then for node m representing a region R with N observations, a common criterion to minimize is the Mean Squared Error.

Classification

If the target is a Nominal or Categorical Variable, then we use Classification.

Classification trees, are used to separate a dataset into classes belonging to the response variable. Usually the response variable has two classes: Yes or No (1 or 0). If the target variable has ***more*** than 2 categories, then a variant of the algorithm, called C4.5, is used. For binary splits however, the standard CART procedure is used. Thus classification trees are used when the response or target variable is categorical in nature.

Given training vectors X and a label vector y, a decision tree recursively partitions the space such that the samples with the same labels are grouped together.

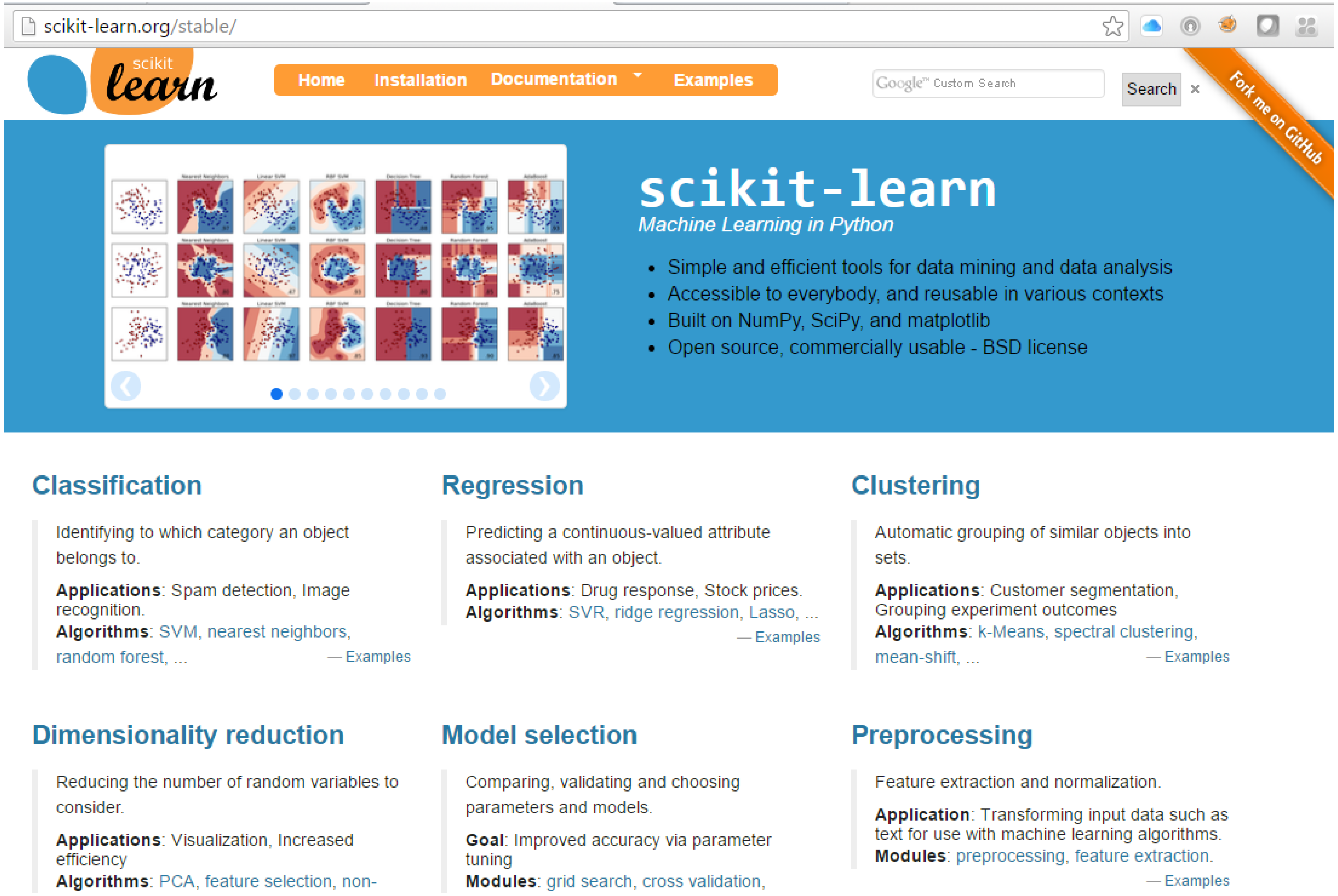
Sci-Kit Learn*

What is Sci-Kit Learn?

Scikit-learn was initially developed by David Cournapeau as a Google summer of code project in 2007. Later Matthieu Brucher joined the project and started to use it as apart of his thesis work. In 2010 INRIA got involved and the first public release (v0.1 beta) was published in late January 2010.

* Source: Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Screenshot of SK LEARN WEB SITE IN BROWSER



scikit-learn

Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.
Algorithms: SVM, nearest neighbors, random forest, ...

— Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.
Algorithms: SVR, ridge regression, Lasso, ...

— Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes
Algorithms: k-Means, spectral clustering, mean-shift, ...

— Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency
Algorithms: PCA, feature selection, non-

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning
Modules: grid search, cross validation,

Preprocessing

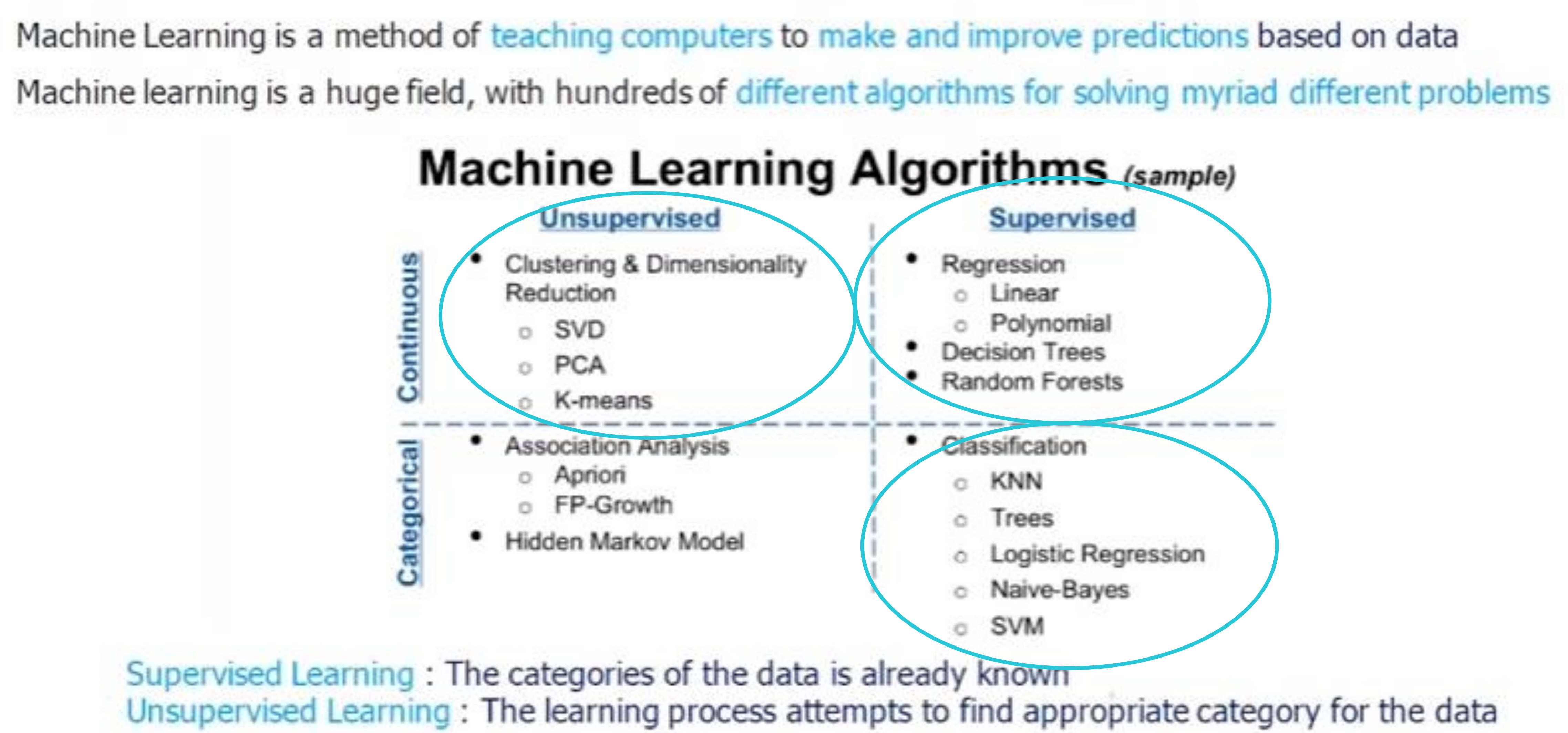
Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.
Modules: preprocessing, feature extraction.

— Examples

Model Building Techniques

How to know which modeling technique to use and when?



Machine Learning is a method of teaching computers to make and improve predictions based on data

Machine learning is a huge field, with hundreds of different algorithms for solving myriad different problems

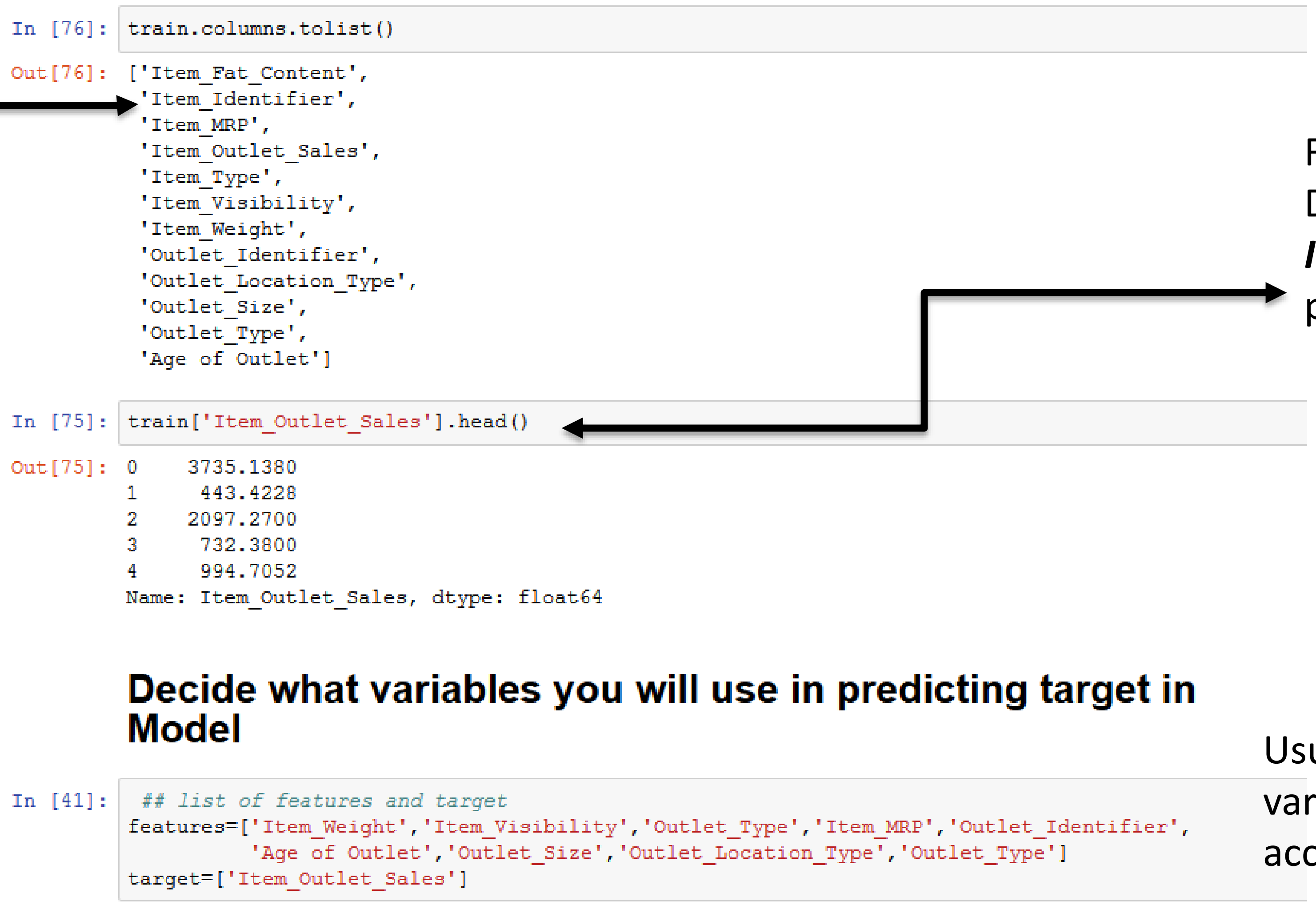
Machine Learning Algorithms *(sample)*

	Unsupervised	Supervised
Continuous	• Clustering & Dimensionality Reduction ○ SVD ○ PCA ○ K-means	• Regression ○ Linear ○ Polynomial • Decision Trees • Random Forests
Categorical	• Association Analysis ○ Apriori ○ FP-Growth • Hidden Markov Model	• Classification ○ KNN ○ Trees ○ Logistic Regression ○ Naive-Bayes ○ SVM

Supervised Learning : The categories of the data is already known
Unsupervised Learning : The learning process attempts to find appropriate category for the data

Select the Target and the Predictor Variables

You can select as many or as little as you feel appropriate to help solve the problem



```
In [76]: train.columns.tolist()

Out[76]: ['Item_Fat_Content',
          'Item_Identifier',
          'Item_MRP',
          'Item_Outlet_Sales',
          'Item_Type',
          'Item_Visibility',
          'Item_Weight',
          'Outlet_Identifier',
          'Outlet_Location_Type',
          'Outlet_Size',
          'Outlet_Type',
          'Age of Outlet']
```

```
In [75]: train['Item_Outlet_Sales'].head()

Out[75]: 0    3735.1380
         1     443.4228
         2    2097.2700
         3     732.3800
         4     994.7052
         Name: Item_Outlet_Sales, dtype: float64
```

Decide what variables you will use in predicting target in Model

```
In [41]:  ## list of features and target
         features=['Item_Weight','Item_Visibility','Outlet_Type','Item_MRP','Outlet_Identifier',
                   'Age of Outlet','Outlet_Size','Outlet_Location_Type','Outlet_Type']
         target=['Item_Outlet_Sales']
```

For example: In looking at current Dataset, there is a numeric variable ***Item_Outlet_Sales*** that needs to be predicted.

You can exclude those you don’t want as a predictor variable

Usually the lesser the number of variables needed for the same accuracy, the better

Linear Regression

Linear Regression is a simple but powerful technique for modeling numeric outcomes

Linear Regression helps us understand the relationship between the Target variable and the Predictor variables, one variable at a time. When any one Predictor changes, what happens to the Target variable? That is what Linear Regression tells us.

Linear Regression is one of the most popular algorithms used for Prediction and Forecasting

Linear vs. Logistic Regression:

If the variable is continuous, Linear Regression is used.
If the variable is discrete, Logistic Regression is probably better

R^2 vs. Correlation:

R^2 is a measure of how much of the variance in *y* is explained by the model, *f*.
For optimal models, R^2 is the Square of the Correlation between the true and predicted outcomes. This relationship is not true for general model and target.

How do we measure the fit of the Model?

R Squared is the best measure

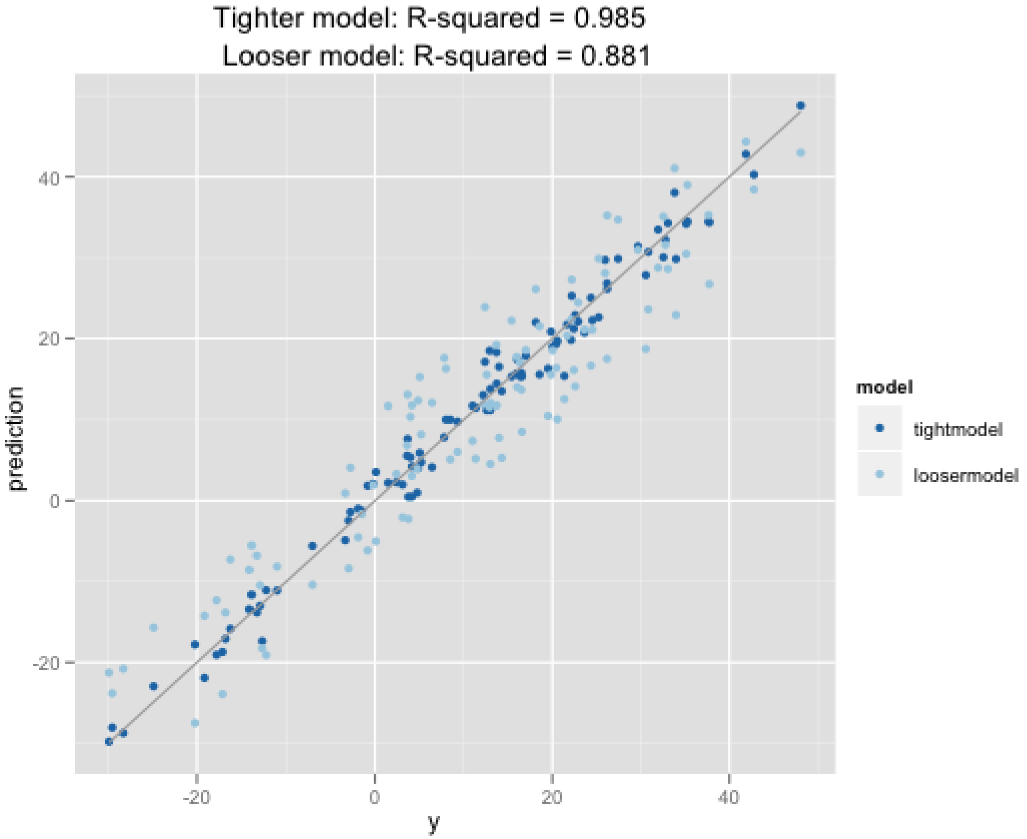


Chart: http://www.win-vector.com/blog/2011/11/correlation-and-r-squared/

Other Regression Modeling Techniques

There are at least 34 different Regression Modeling Techniques in the world

Do’s and Don’ts

- If you have a cyclical data set, to try and fit a line to it, may not be such a great idea!
- Neural Networks are a Non Linear Modeling Technique that can improve your model significantly
- Random Forests are also a good bet
- Gradient Boosting and Extreme Gradient Boosting improve upon Random Forests

- The method used to increase/decrease variables is called Mean Decrease in Gini Coefficient
- They use relative scoring of all the features

- Another way to reduce the number of variables is called “Principal Component Analysis”
- - PCA uses a table of Eigen Values (which are derived from Correlation Coefficients)
- Then you can create new variables called PC1, PC2, etc. that basically combine 2 or more variables into a single variable

Popular Regression Libraries in SK Learn

isotonic.IsotonicRegression

linear_model.ARDRegression

linear_model.LinearRegression

cross_decomposition.PLSRegression

ensemble.AdaBoostRegressor

ensemble.BaggingRegressor

ensemble.ExtraTreesRegressor

ensemble.GradientBoostingRegressor

ensemble.RandomForestRegressor

linear_model.PassiveAggressiveRegressor

linear_model.SGDRegressor

neighbors.KNeighborsRegressor

neighbors.RadiusNeighborsRegressor

tree.DecisionTreeRegressor

tree.ExtraTreeRegressor

Decision Tree for Regression

An Example of a Decision Tree from SKLEARN for Regression Purposes

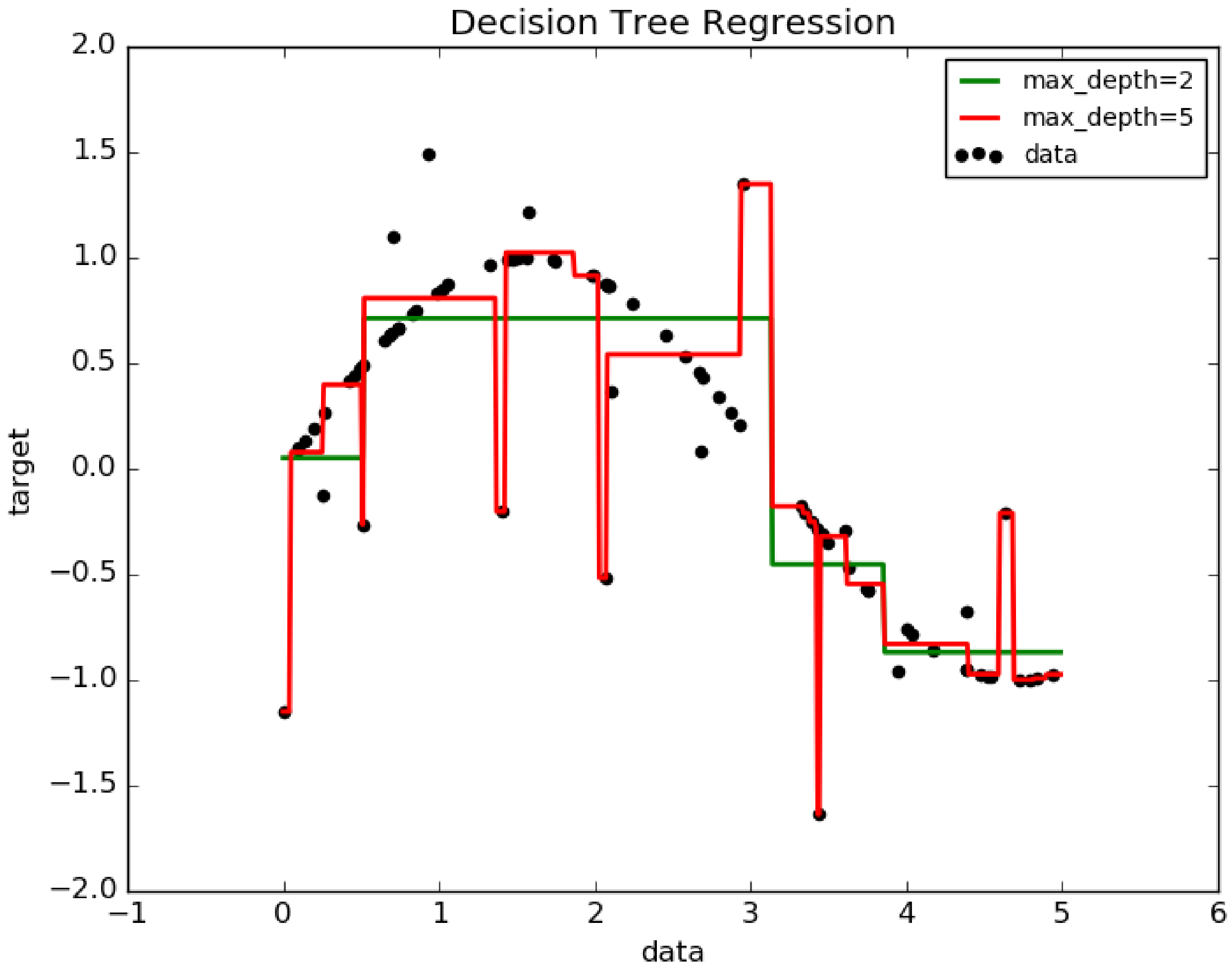
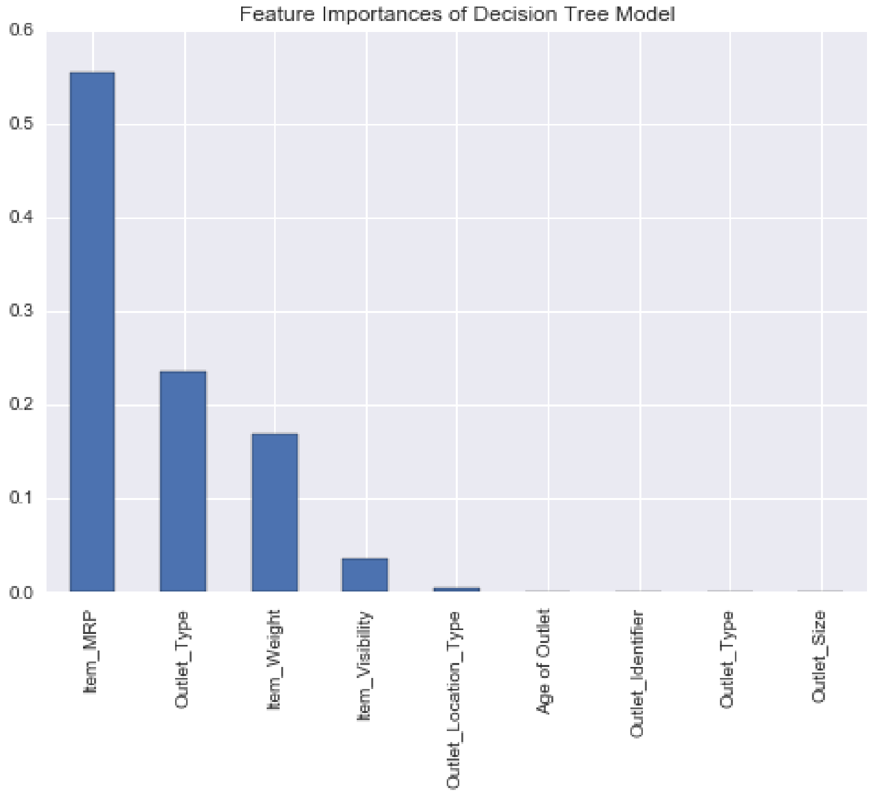


Chart: http://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html#example-tree-plot-tree-regression-py

The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.



The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance.

Decision Trees

Some advantages of decision trees are:

•Simple	•Simple to understand and to interpret. Trees can be visualized.
Very Little Prep	Requires little data preparation. Other techniques often require data normalization, dummy variables need to be created and blank values to be removed. Note however you must remove missing values.
Very Fast	The cost of using the tree (i.e., predicting data) is logarithmic in the number of data points used to train the tree.
All Data Types	Able to handle both numerical and categorical data. Other techniques can handle only one type of variable
Multiple Target Variables	Able to handle multi-output problems
Transparent	Uses a white box model. Easily explained by Boolean logic. It’s not a black box model (e.g., in an artificial neural network), which may be more difficult to interpret.
Easy Validation	Possible to validate a model using statistical tests. That makes it possible to account for the reliability of the model.
Performs well	Even under conditions that may violate the true model from which the data were generated.

Decision Trees

There are caveats to using Decision Trees in all circumstances

The disadvantages of decision trees include:

Over Fitting

You can create overly complex trees that do not generalize the data well (Overfitting) Mechanisms such as setting the minimum number of samples required at a leaf node or setting the maximum depth of the tree available

Very Sensitive

Can be “unstable” because small variations in the data might result in a completely different tree being generated. You can reduce tis by using them within an ensemble model.

Locally Optimized

“Greedy Algorithm” which works at local optimization but cannot guarantee that it is globally optimized. Again, use in an ensemble.

Some Missing

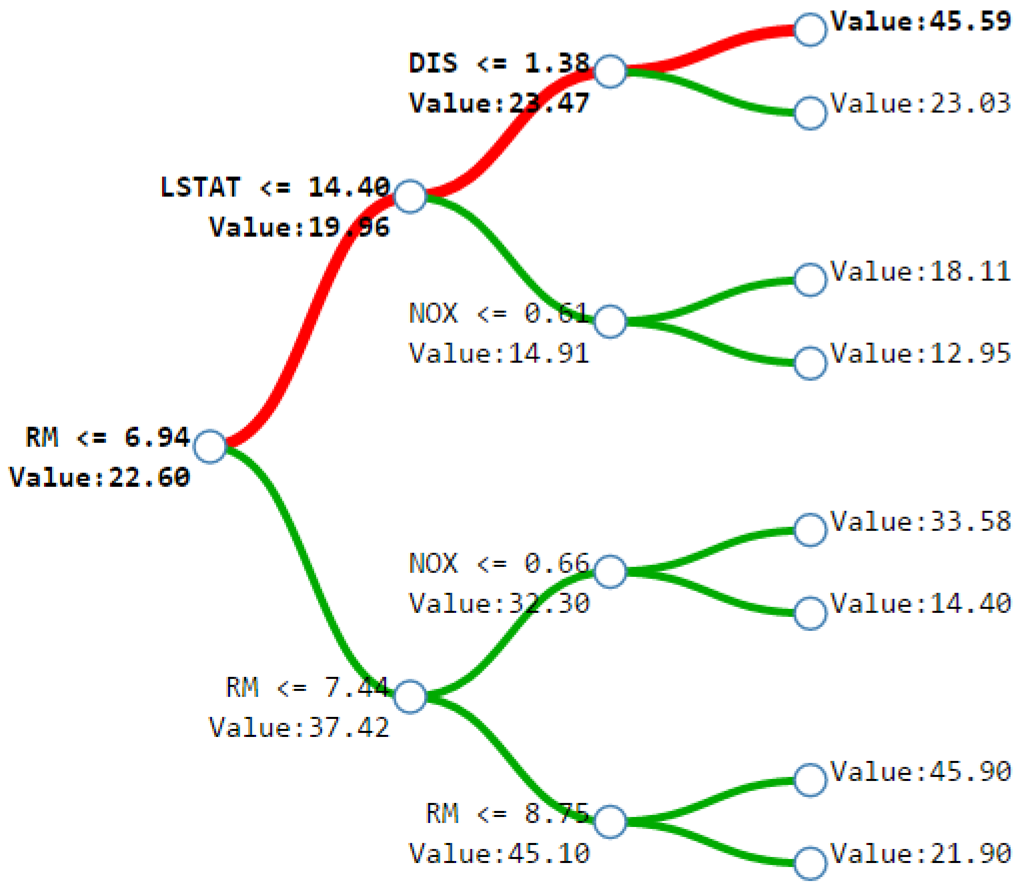
XOR, parity or multiplex problems

Can be biased

In favor of some dominant variables. It is therefore recommended to balance the dataset prior to fitting with the decision tree.

Ensemble Methods

Ensemble is a decision tree made with many different models. It has a higher success rate than a single decision tree since many decision trees are involved



Prediction: 45.59 ≈ 22.60 (trainset mean) - 2.64(loss from RM) + 3.52(gain from LSTAT) + 22.12(gain from DIS)

Chart: http://blog.datadive.net/interpreting-random-forests/

Boosting and Bagging

Ensemble methods combine the predictions of several different techniques in order to improve the generalizability / robustness over a single model.
Two families of ensemble methods are usually distinguished:

•In **Bagging methods**, the driving principle is to build several models independently and then to average their predictions. On average, the combined model is usually better than any of the single base model because its variance is reduced.

•**Example**: Random Forests, ...

•By contrast, in **boosting methods**, a base mode is built and then subsequent models built on it to reduce the bias of the combined model. The idea is to combine several weak models to produce a powerful model.

•**Example**: Extreme Gradient Boosting, ...

Random Forests

A diverse set of models is created by introducing randomness in the model construction. The prediction of the ensemble is given as the averaged prediction of the individual models.

Advantages

- Multiple types exist:
  - Pasting: When random subsets of the dataset are drawn as random subsets of the samples
  - Bagging: When samples are drawn with replacement
  - Random Subspaces: When random subsets of the dataset are drawn as random subsets of the features
  - Random Patches: when base estimators are built on subsets of both samples and features
- Parallelization: Allows the parallel construction of trees and the parallel computation of predictions through the n_jobs parameter

Disadvantages

- Random forests are typically treated as black boxes but they can be made transparent using new techniques

Model performance is computed based on the amount of "impurity" (typically variance in case of regression trees and gini coefficient or entropy in case of classification trees)

Classification Problems: An Example

We will use the Famous IRIS data set to see how classification works

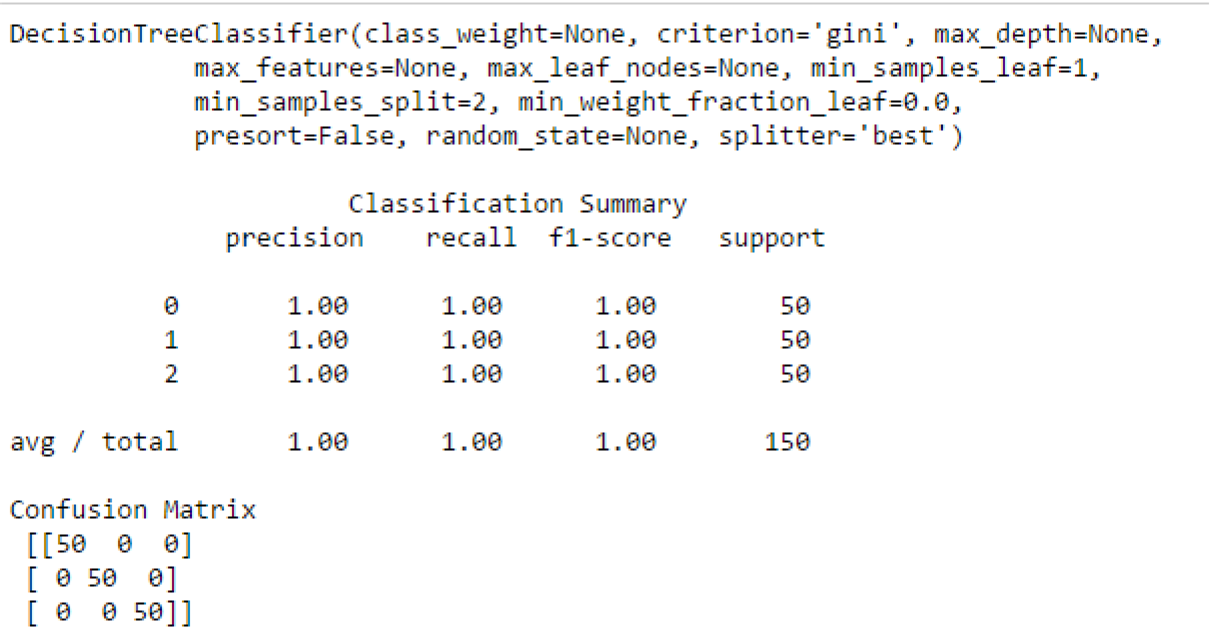
Input

```
# Decision Tree Classifier
import pandas as pd

from sklearn.datasets import load_iris
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics

# load the iris datasets
dataset = load_iris()
# fit a Classification and Regression Tree (CART) model
model = DecisionTreeClassifier()
model.fit(dataset.data, dataset.target)
print(model)
# make predictions
expected = dataset.target
predicted = model.predict(dataset.data)
# summarize the fit of the model
print('\n                    Classification Summary \n',
     metrics.classification_report(expected, predicted))
print('Confusion Matrix \n', metrics.confusion_matrix(expected, predicted))
```

Output



```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
            max_features=None, max_leaf_nodes=None, min_samples_leaf=1,
            min_samples_split=2, min_weight_fraction_leaf=0.0,
            presort=False, random_state=None, splitter='best')

                   Classification Summary
              precision    recall  f1-score   support

          0       1.00      1.00      1.00        50
          1       1.00      1.00      1.00        50
          2       1.00      1.00      1.00        50

avg / total       1.00      1.00      1.00       150

Confusion Matrix
 [[50  0  0]
 [ 0 50  0]
 [ 0  0 50]]
```

Clustering Problems: An example using Iris

Clustering is about finding groups of similar data in an unlabeled data set. It is an example of Unsupervised Learning.

There are a number of methods. I will list the few major ones I know here:

Hierarchical Clustering
K-Means
Mean Shift
Affinity propagation
Spectral clustering

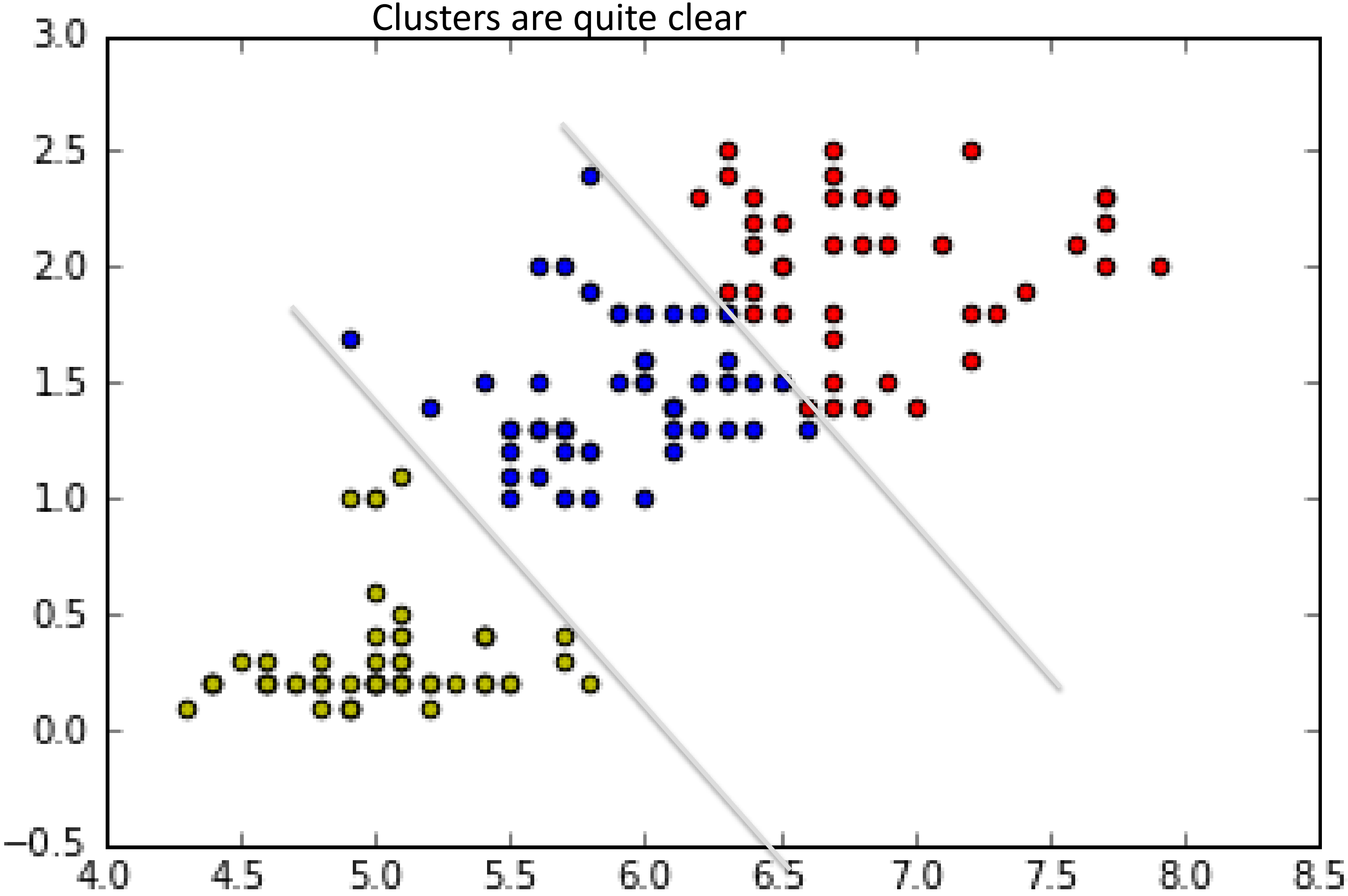
Cluster analysis is a popular classification technique frequently used to analyze market research data which divides customers into groups.

It can be used identify demographic or psychographic characteristics of customers with similar buying patterns or to highlight differences between groups of products.

Clustering Must Do's:

Always send in only after filling in missing values
Always send in only after removing outliers
Always perform Z-Score or Other Transforms to the Data
Always send in only Numeric Variables
Always send in only after doing Principal Component Analysis to find the 2 most important variables
You can do 2 variables at a time if you want see visually in 2D
You can have any number of clusters within those 2 variables

Clusters are quite clear



With 2 variables

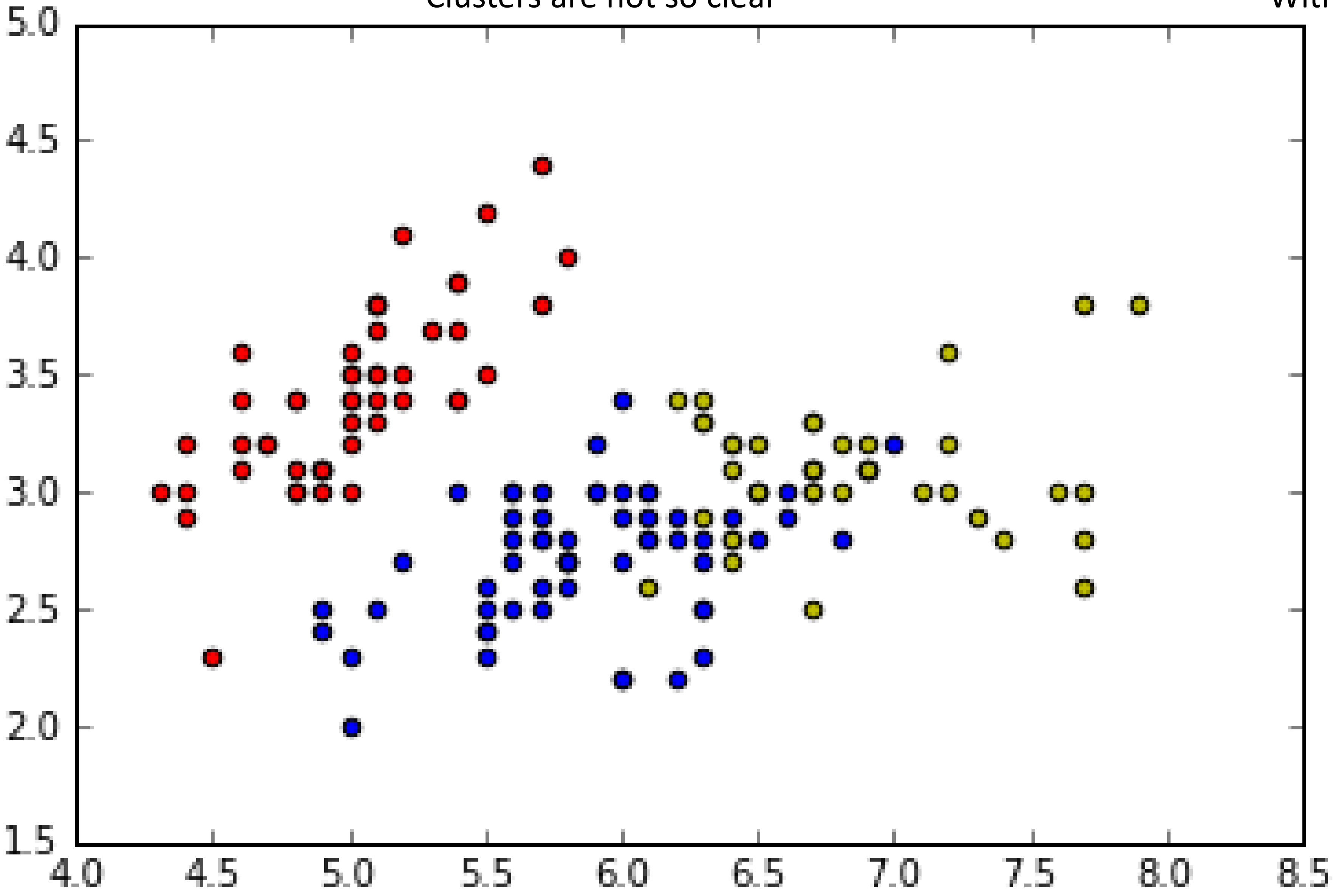
Iris Data Set
Clustering using K Means
Used only 2 variables out of the 4 available

Clustering in 2 D

Clusters are not so clear

With 4 variables

This View is slightly Different because how do you plot in 4 dimensions when 4 variables are involved? So we plotted using only the first 2 variables



The same Iris data set

Clustered using K Means with all 4 variables

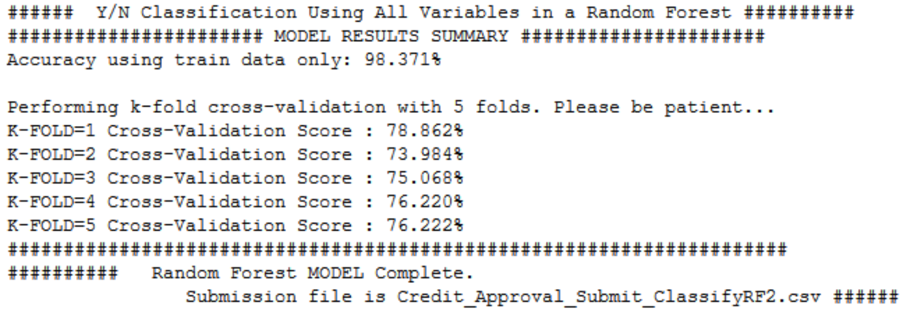
Clustering in 4 D?

Model Evaluation

Process of deriving Insights from Data

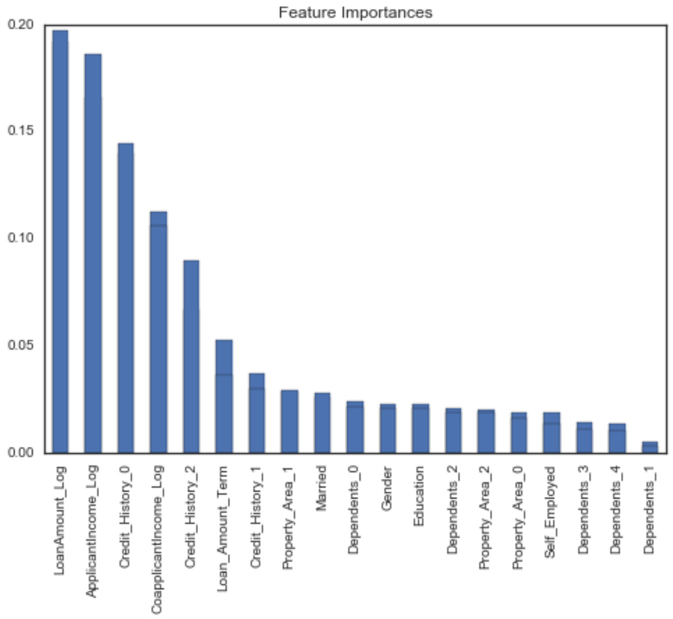
Model Evaluation

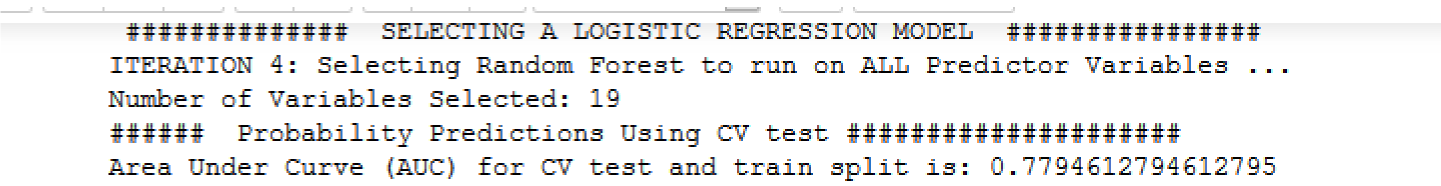
How to Evaluate results of a Model



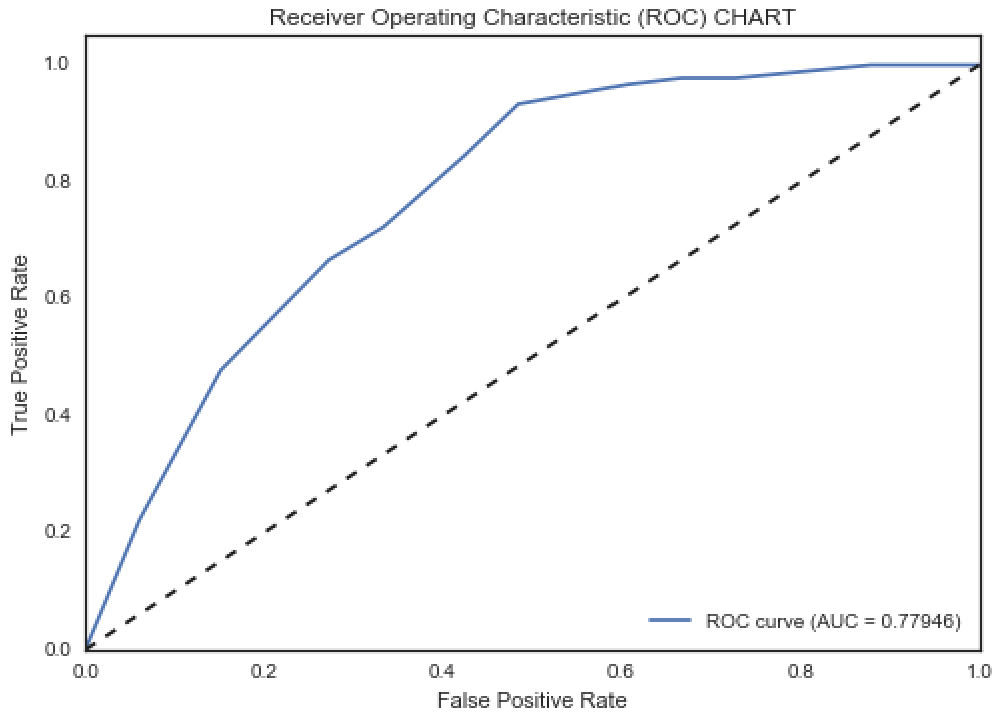
```
######  Y/N Classification Using All Variables in a Random Forest ##########
###################### MODEL RESULTS SUMMARY #####################
Accuracy using train data only: 98.371%

Performing k-fold cross-validation with 5 folds. Please be patient...
K-FOLD=1 Cross-Validation Score : 78.862%
K-FOLD=2 Cross-Validation Score : 73.984%
K-FOLD=3 Cross-Validation Score : 75.068%
K-FOLD=4 Cross-Validation Score : 76.220%
K-FOLD=5 Cross-Validation Score : 76.222%
######################################################################
##########   Random Forest MODEL Complete.
               Submission file is Credit_Approval_Submit_ClassifyRF2.csv ######
```





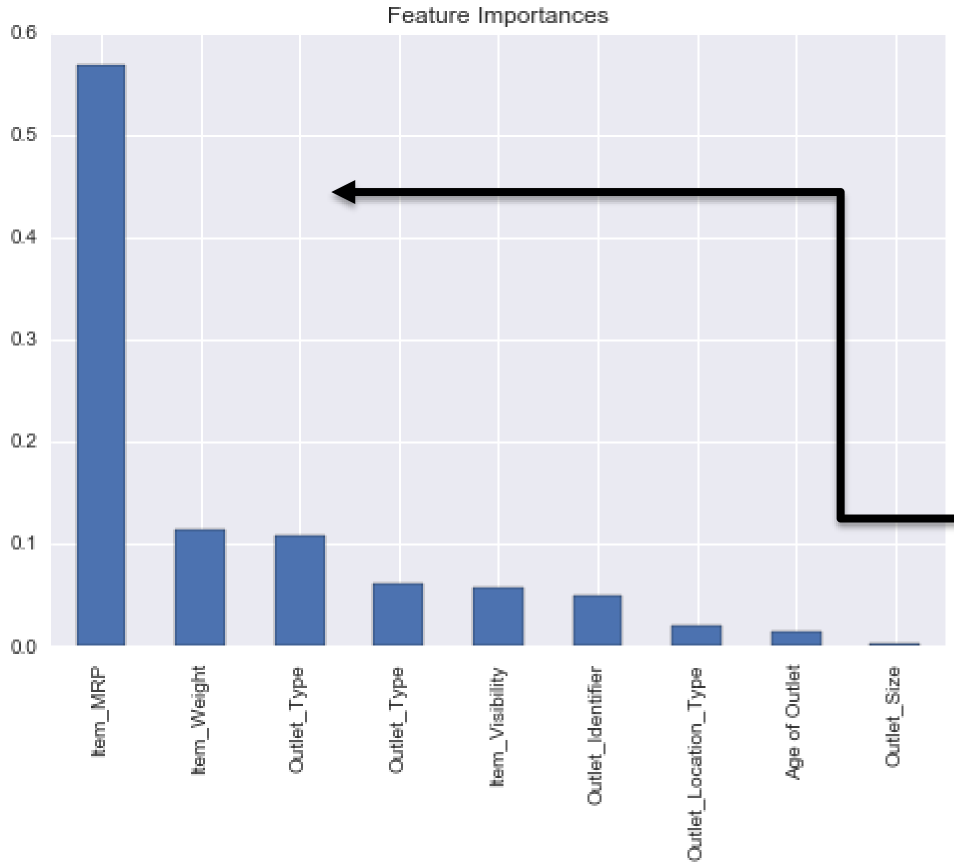
```
##############  SELECTING A LOGISTIC REGRESSION MODEL  ################
ITERATION 4: Selecting Random Forest to run on ALL Predictor Variables ...
Number of Variables Selected: 19
######  Probability Predictions Using CV test #####################
Area Under Curve (AUC) for CV test and train split is: 0.7794612794612795
```



```
##########   Random Forest MODEL Complete.
```

Variable Importance

How to know Importance of a Variable in predicting outcome



Python Code

```
coef1 = pd.Series(model.feature_importances_,
features).sort_values(ascending=False)
coef1.plot(kind='bar', title='Feature Importances')
```

The **expected fraction of the samples** they contribute to is used as an estimate of the **relative importance of the features**

The higher the number the better.

Sometimes highest number of the Std Parameter is "normalized" to 100 and the rest of the variables are compared to that 100.