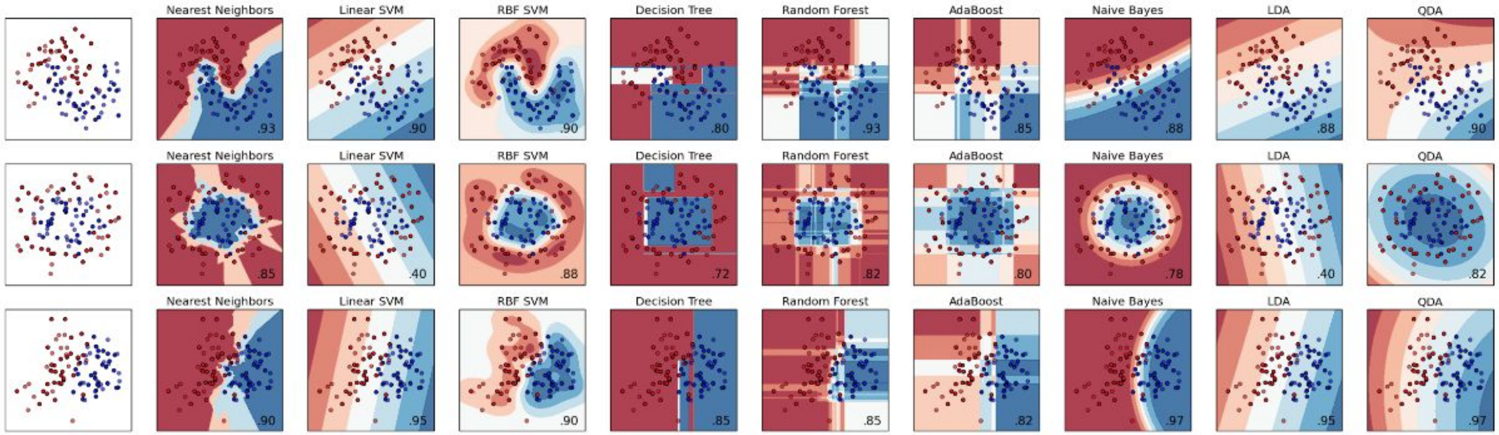
Scikit-Learn

By Ayush Arora

Introduction

- Machine Learning in Python
- Wide selection of various learning algorithms
- Easy to use, clean, yet powerful
- Numpy, Scipy, Matplotlib, IPython, Sympy, and Pandas



A comparison Scikit Learn's many Machine Learning models

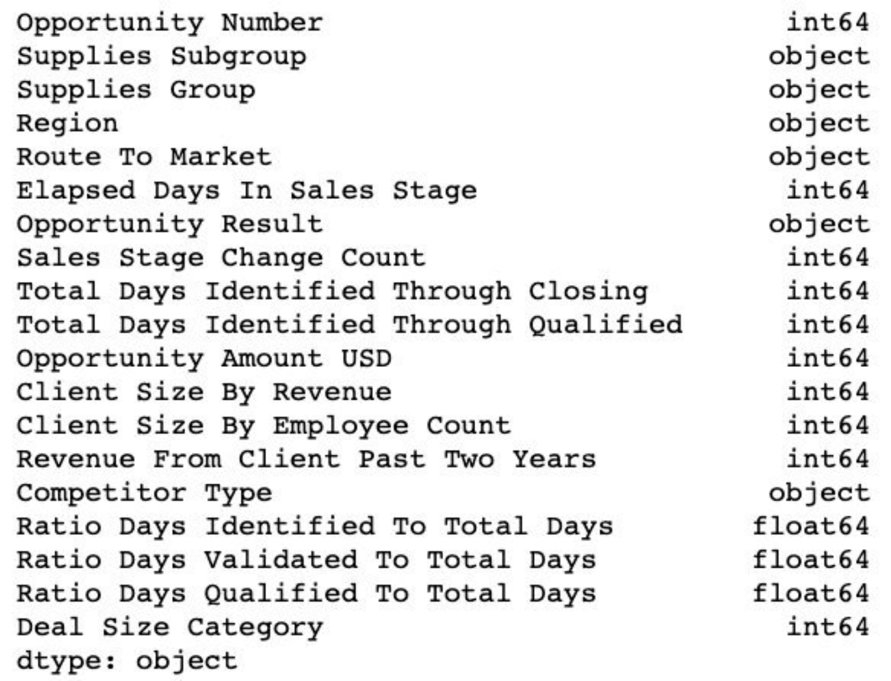
- Several Algorithms to create prediction models
  - Naive Bayes
  - Regression
  - Clustering

Goal

- Given a sales win & loss dataset from IMB
  - Sales campaign data of automotive parts
- Use scikit-learn to build a predictive model
  - Learn from existing dataset
  - Predict which sales campaign will result in loss? In a win?

The Dataset

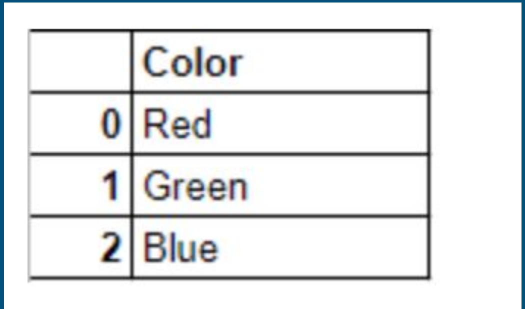
- 78,024 rows
  - 17,627 wins
  - 60,398 losses
- 19 features
- Target Variable
  - Opportunity Result (win/loss)
- Used Seaborn/Matplotlib
  - For Visualization



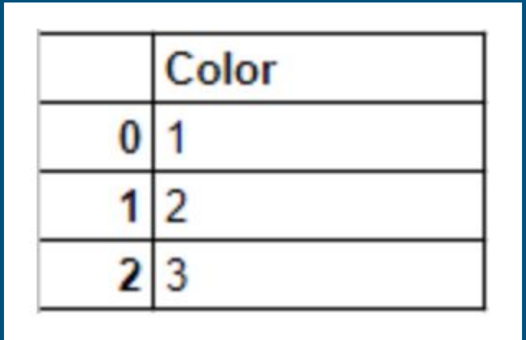
```
Opportunity Number                              int64
Supplies Subgroup                              object
Supplies Group                                 object
Region                                         object
Route To Market                                object
Elapsed Days In Sales Stage                     int64
Opportunity Result                             object
Sales Stage Change Count                        int64
Total Days Identified Through Closing           int64
Total Days Identified Through Qualified         int64
Opportunity Amount USD                          int64
Client Size By Revenue                          int64
Client Size By Employee Count                   int64
Revenue From Client Past Two Years              int64
Competitor Type                                object
Ratio Days Identified To Total Days           float64
Ratio Days Validated To Total Days            float64
Ratio Days Qualified To Total Days            float64
Deal Size Category                              int64
dtype: object
```

Preprocessing Data

- Many of the columns contain string data
- Must be converted into numerical data before using ML algorithms
  - LabelEncoder().fit_transform() function
- Example



	Color
0	Red
1	Green
2	Blue

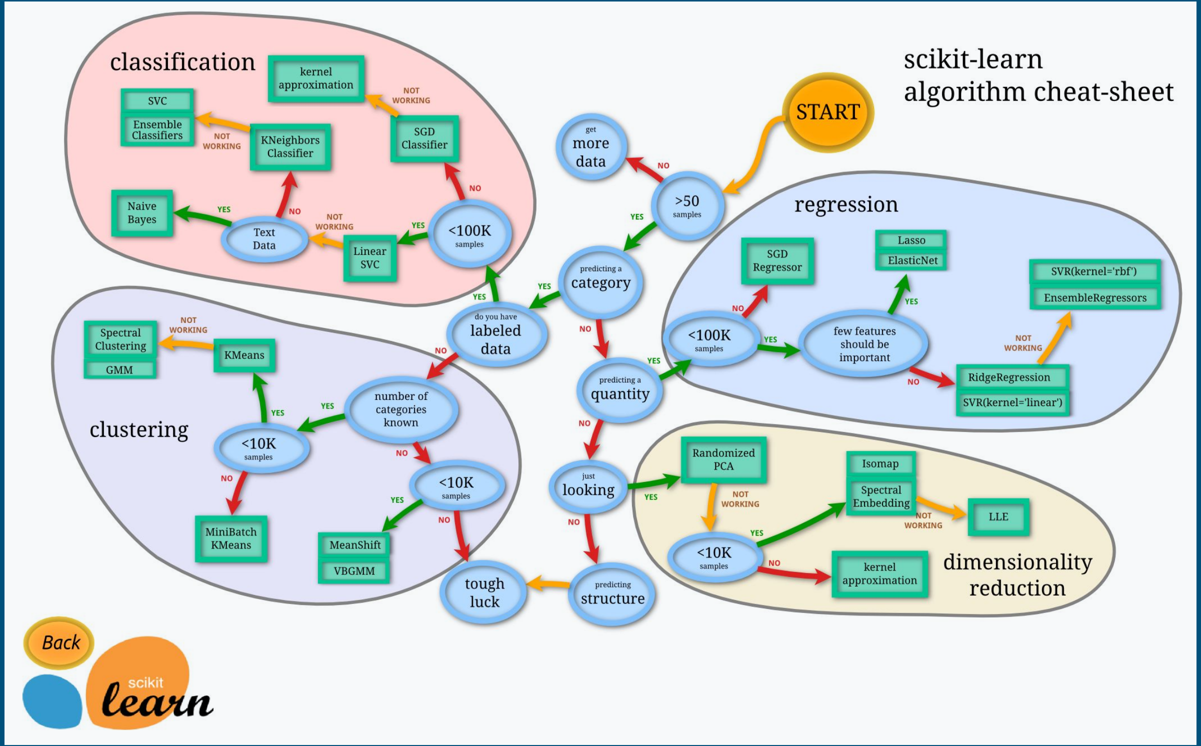


	Color
0	1
1	2
2	3

Training & Testing Set

- Separate target variable from dataset
- ML Algorithm needs to be trained on a set to learn relationship between features and target variable
- Training Set
  - Contains target variable
  - Used to train algorithm and build model
- Testing Set
  - Contains only features
  - Model predicts target variable
  - Compared with target variable afterwards to see how accurate the model is
- Train_test_split function is used

Building The Model

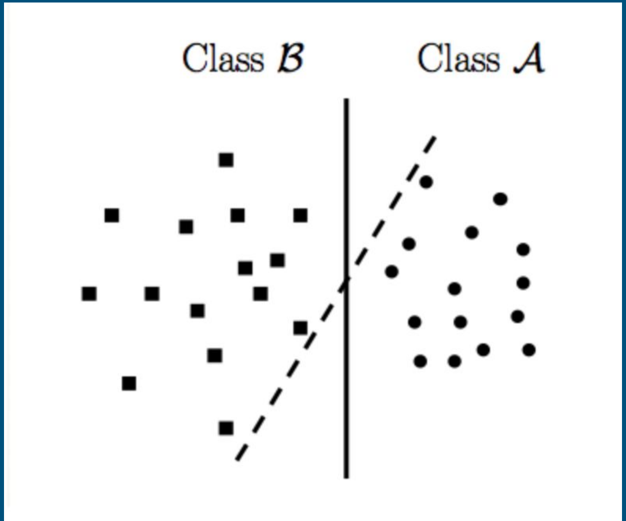


Naive-Bayes

- Classification Algorithm
- Mathematically Complicated
- Calculates probability of connection between feature and target variable
  - Selects feature with highest probability as the one to base its predictions on
- Use GaussianNB

LinearSVC

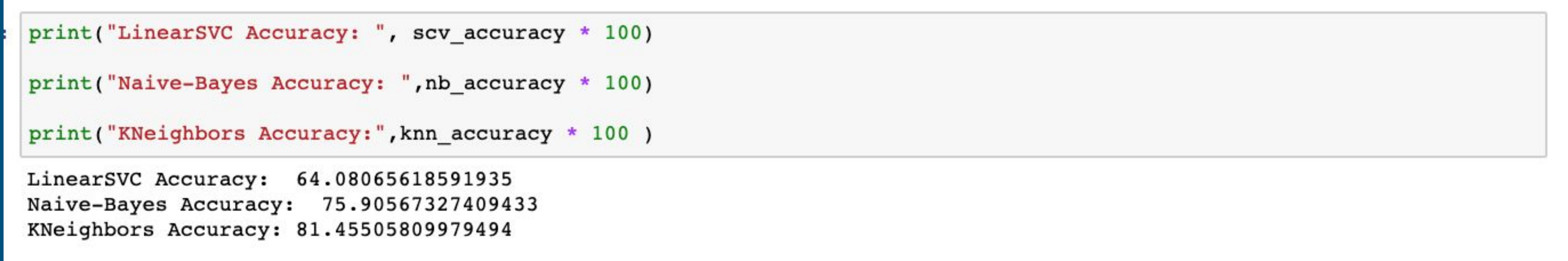
- Another classification algorithm
- Creates line(s) of distinction to divide data according to its target variable
- Use LinearSVC



K-Neighbors Classifier

- Most Complex
- Previous algorithms create straight lines of distinction
- Uses calculus to create curvy distinctions to better fit the data
- Use KNeighborsClassifier

Comparing Accuracy Scores



```
print("LinearSVC Accuracy: ", scv_accuracy * 100)

print("Naive-Bayes Accuracy: ",nb_accuracy * 100)

print("KNeighbors Accuracy:",knn_accuracy * 100 )
```

```
LinearSVC Accuracy:  64.08065618591935
Naive-Bayes Accuracy:  75.90567327409433
KNeighbors Accuracy: 81.45505809979494
```

- KNeighbors turned out to be the most accurate
- Scikit Learn has numerous algorithms that are easy to plug in once the data is preprocessed and ready for prediction modeling

Resources

- https://www.dataquest.io/blog/sci-kit-learn-tutorial/
- https://scikit-learn.org/stable/tutorial/machine_learning_map/
- https://towardsdatascience.com/an-introduction-to-scikit-learn-the-gold-standard-of-python-machine-learning-e2b9238a98ab
- https://www.ibm.com/communities/analytics/watson-analytics-blog/sales-win-loss-sample-dataset/

Questions?