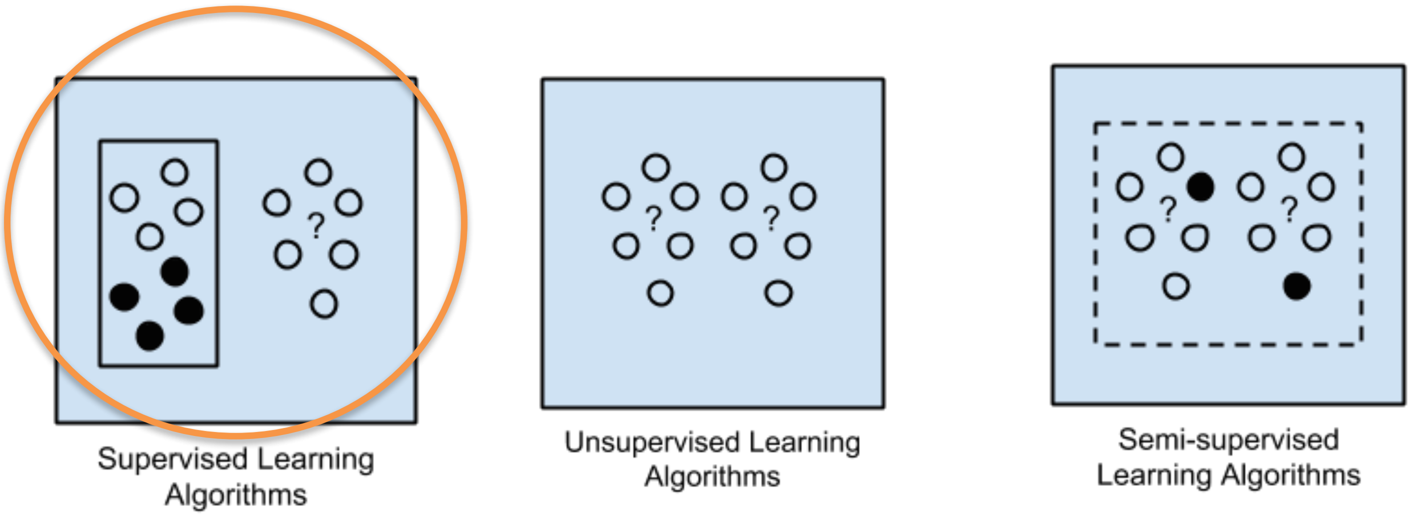
Data ^{X}

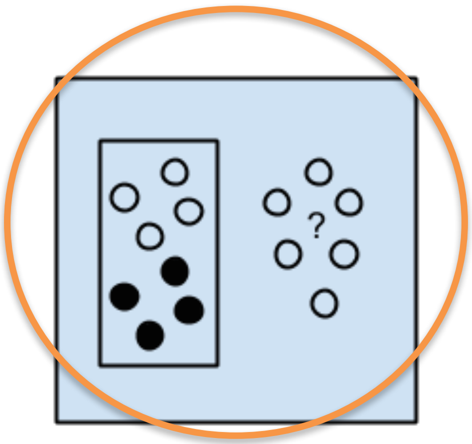
ML Summary and Next Steps in Illustrations

Data X: A Course on Data, Signals, and Systems

Ikhlaq Sidhu
Chief Scientist & Founding Director,
Sutardja Center for Entrepreneurship & Technology
IEOR Emerging Area Professor Award, UC Berkeley

Overview

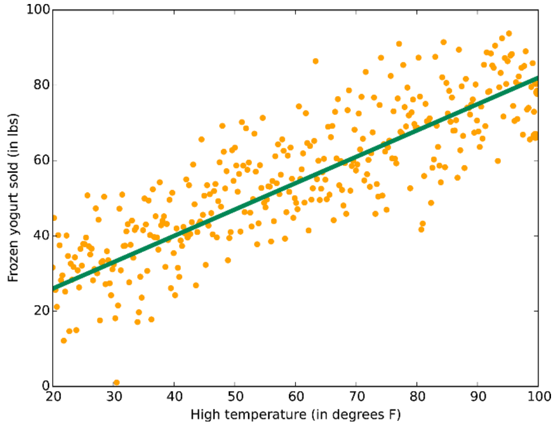

?
?
?
?
?
Supervised Learning Algorithms
Unsupervised Learning Algorithms
Semi-supervised Learning Algorithms



Supervised Learning
Algorithms

```
#Setting up for Supervised learning
# First clean: use mapping + buckets

# X = matrix of data – e.g 1000 rows
# Y = In sample responses

# Typically we want to split in to
training data and test data

X_train = X[0:500]
Y_train = Y[0:500]
X_test = X[501:1000]
Y_test = Y[501:1000]
```

Linear Regression Illustration




```
#Setting Linear Regression in sklearn
from sklearn import linear_model

model= linear_model.LinearRegression()
model.fit(X_train, Y_train)

Y_pred_train =  model.predict(X_train)
Y_pred_test =  model.predict(X_test)


# Compare Y_pred_test with Y_test for
error.
```

Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

Logistic Regression Illustration

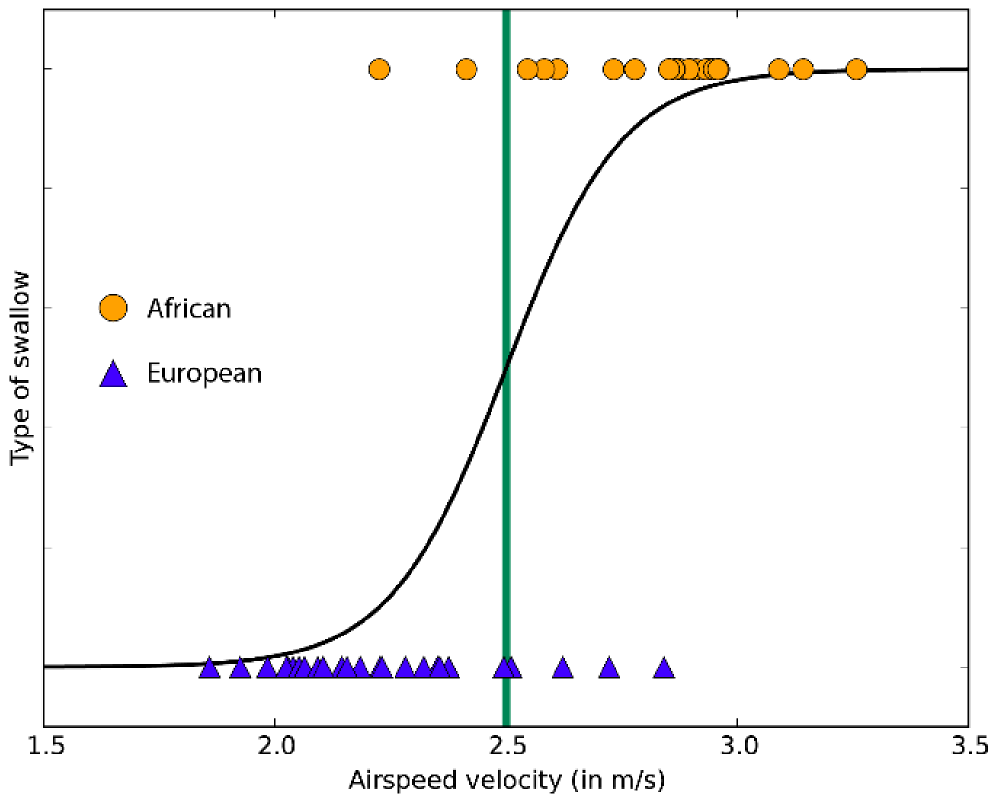
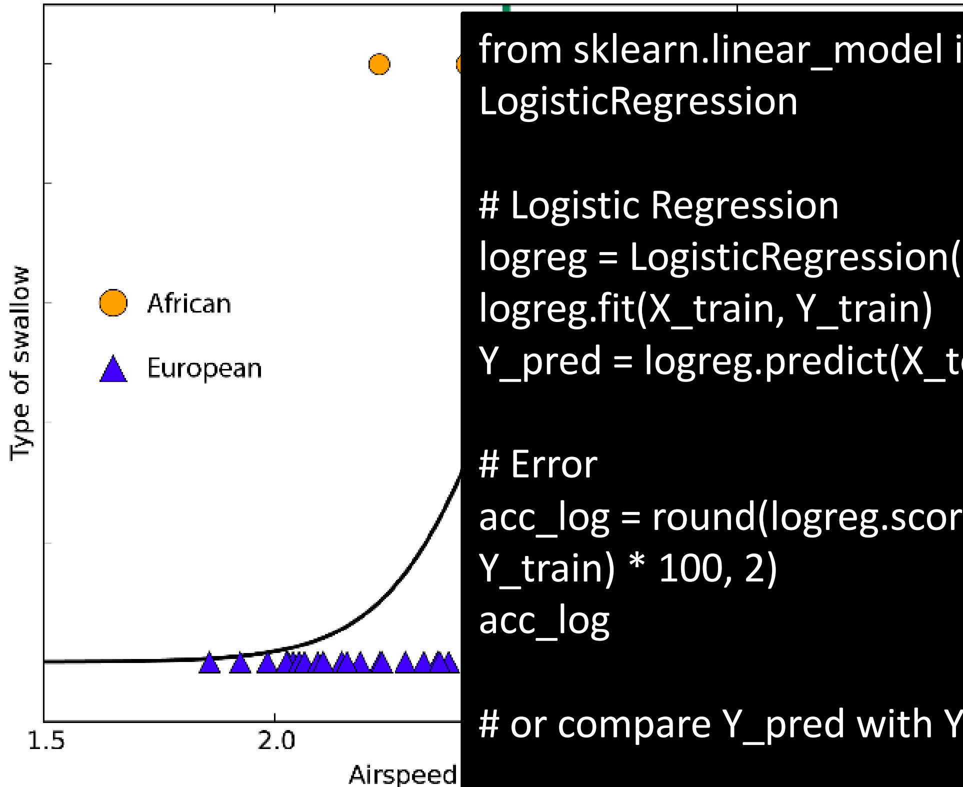


Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

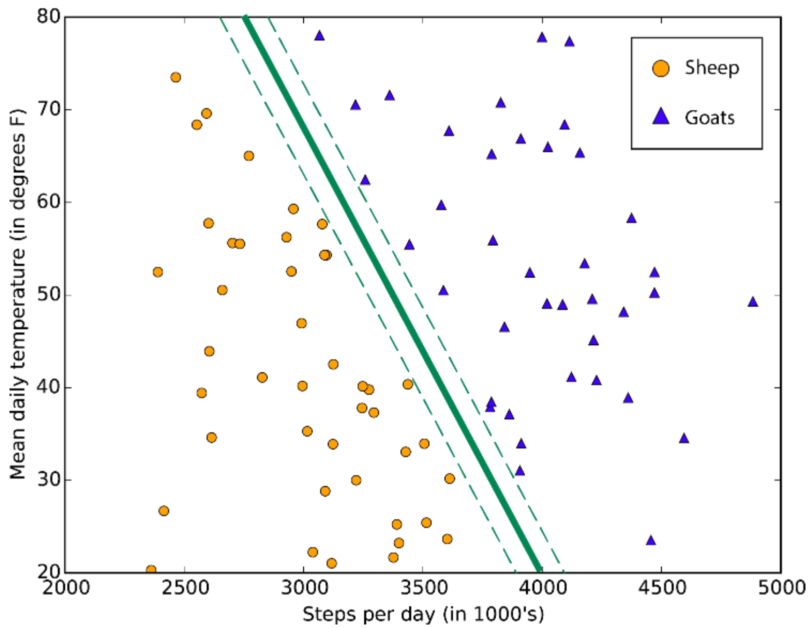
Logistic Regression Illustration




```
from sklearn.linear_model import
LogisticRegression

# Logistic Regression
logreg = LogisticRegression()
logreg.fit(X_train, Y_train)
Y_pred = logreg.predict(X_test)

# Error
acc_log = round(logreg.score(X_train,
Y_train) * 100, 2)
acc_log

# or compare Y_pred with Y_test
```

Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

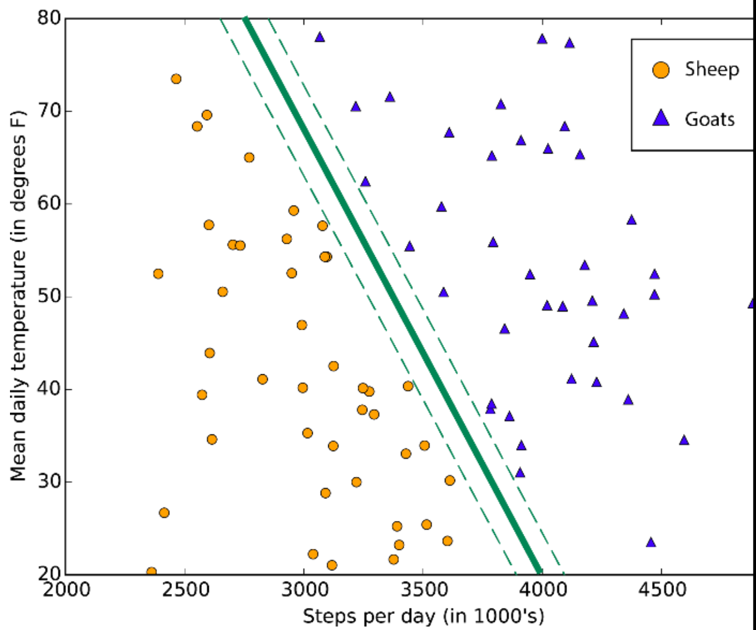
Support Vector Machine (SVM) Illustration



A typical support vector machine class boundary maximizes the margin separating two classes

Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

Support Vector Machine (SVM) Illustration



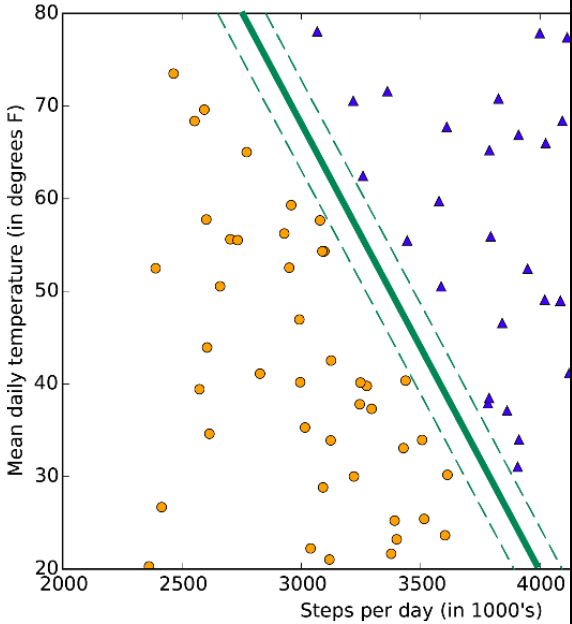
A typical support vector machine class boundary maximizes the margin separating two classes

```
from sklearn.svm import SVC,
LinearSVC

svc = SVC()
svc.fit(X_train, Y_train)
Y_pred = svc.predict(X_test)

# Error
acc_svc = round(svc.score(X_train,
Y_train) * 100, 2)
acc_svc

# or compare Y_pred with Y_test
```

Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

Data X

Support Vector Machine (SVM) Illustration

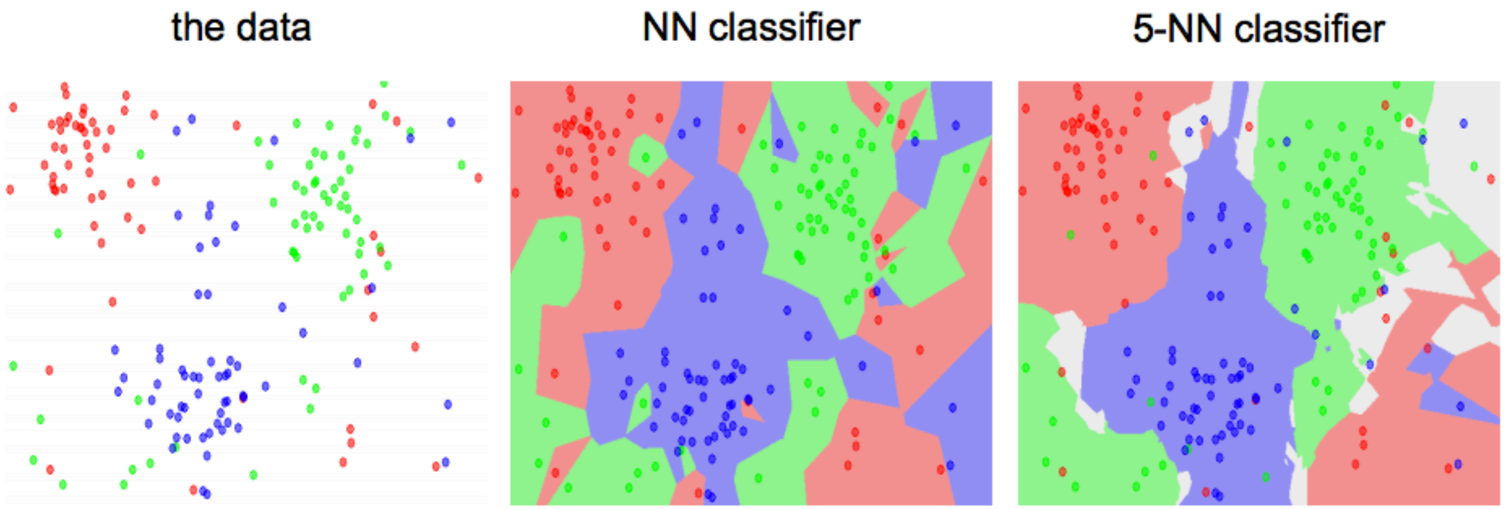


A typical support vector machine class boundary maximizes the m
classes

```
from sklearn.svm import SVC, LinearSVC

# Linear SVC
linear_svc = LinearSVC()
linear_svc.fit(X_train, Y_train)

Y_pred = linear_svc.predict(X_test)

# Error:
acc_linear_svc =
round(linear_svc.score(X_train, Y_train) *
100, 2)acc_linear_svc

# or compare Y_pred with Y_test
```

Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

KNN / K Means Illustration



KNN Method: Find the k nearest images and have them vote on the label (i.e. take the mode)

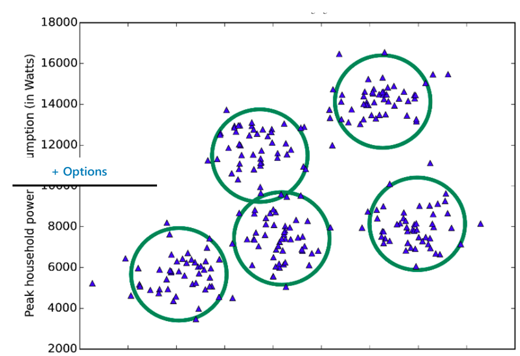
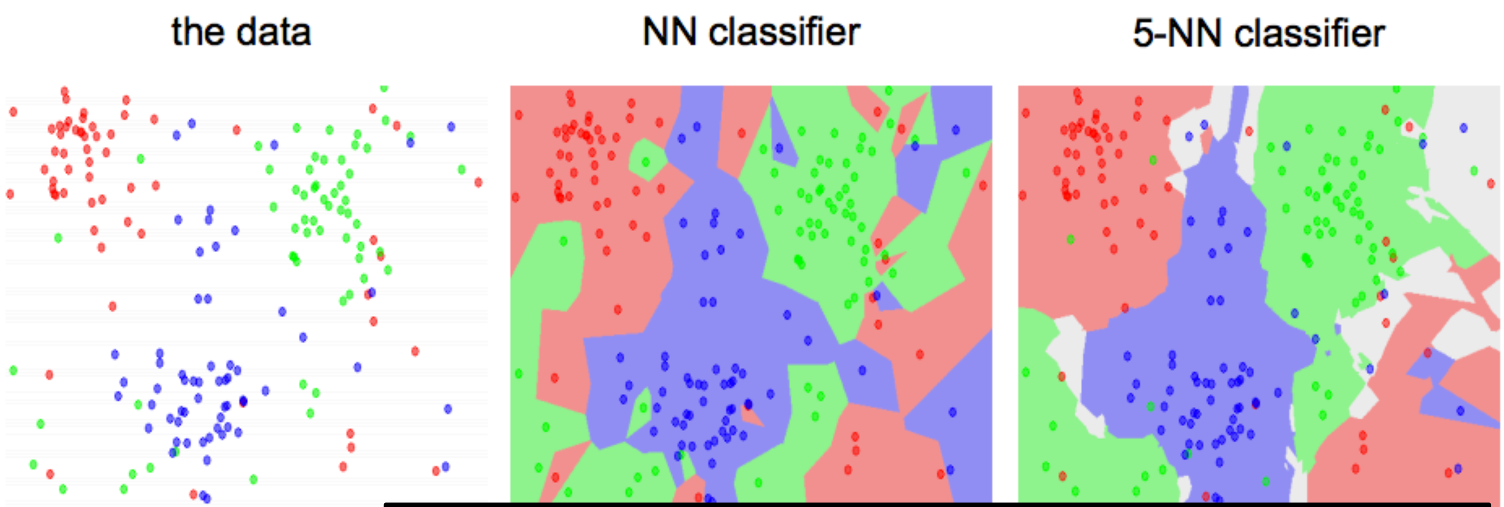


Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

K Means / KNN Illustration

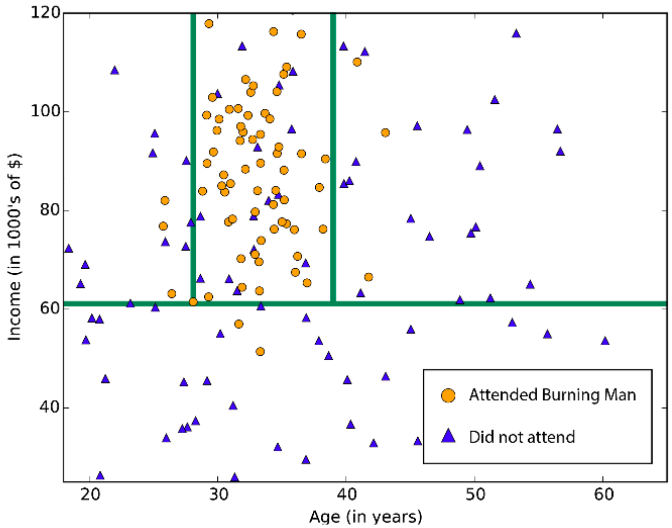


KNN Method: Find the k nearest images and have them vote on the label (i.e. take the mode)

```
from sklearn.neighbors import KNeighborsClassifier

knn = KNeighborsClassifier(n_neighbors = 3)
knn.fit(X_train, Y_train)
Y_pred = knn.predict(X_test)

acc_knn = round(knn.score(X_train, Y_train) * 100, 2)
acc_knn

# or compare Y_pred with Y_test
```

Illustration Source: ht

Decision Tree Illustration



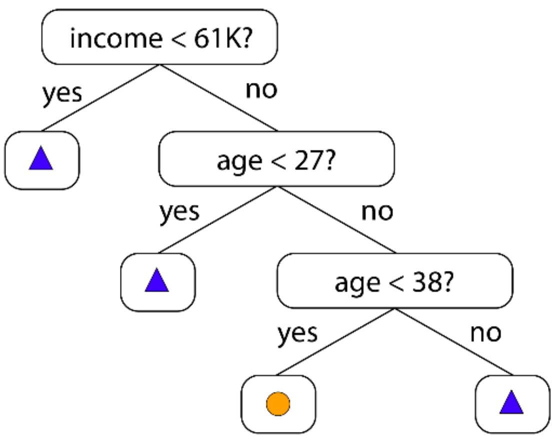
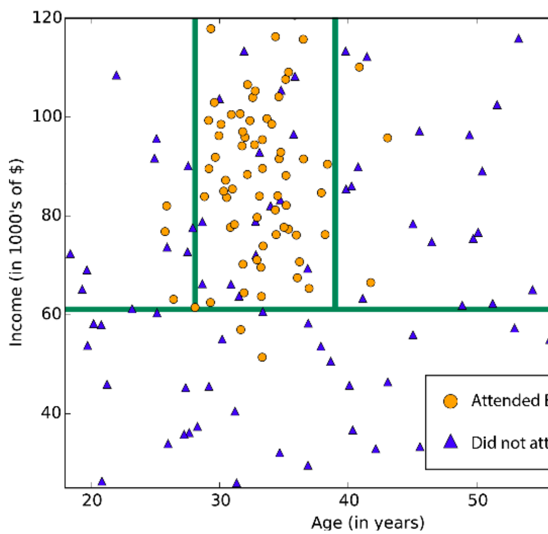


Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

DataX

Decision Tree Illustration




```
from sklearn import tree

decision_tree = DecisionTreeClassifier()
decision_tree.fit(X_train, Y_train)
Y_pred = decision_tree.predict(X_test)

# Error
acc_decision_tree =
round(decision_tree.score(X_train, Y_train) *
100, 2)
acc_decision_tree

# or compare Y_pred with Y_test
```

Illustration Source: https://docs.mic[redacted]ce

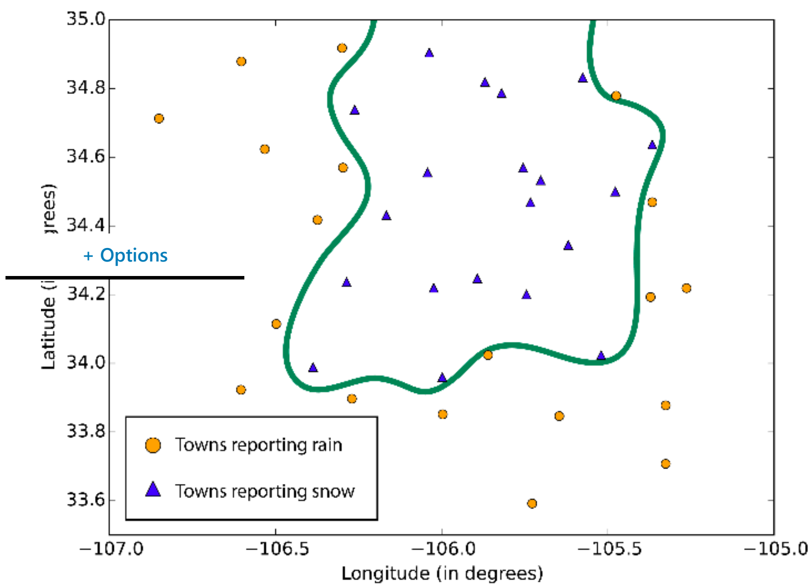
Our experiment with the Titanic Data Set

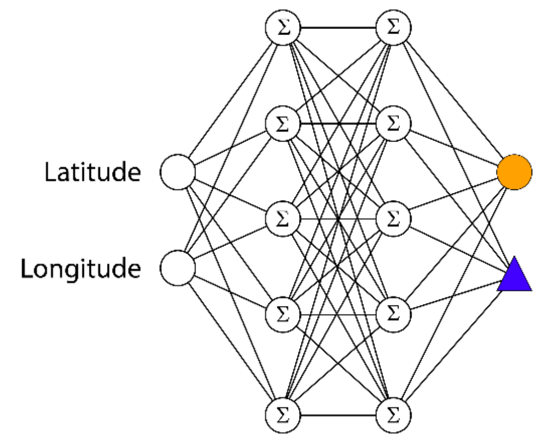
	Model	**Score**
	Random Forest	86.76
	Decision Tree	86.76
	KNN	84.74
	Support Vector Machines	83.84
	Logistic Regression	80.36
	Linear SVC	79.01
	Perceptron	78.00
	Naive Bayes	72.28
	Stochastic Gradient Decent	72.28

More Accuracy
Generally more training time
More risk of overfitting

Less Accuracy
Generally less computation

Neural Network Illustration

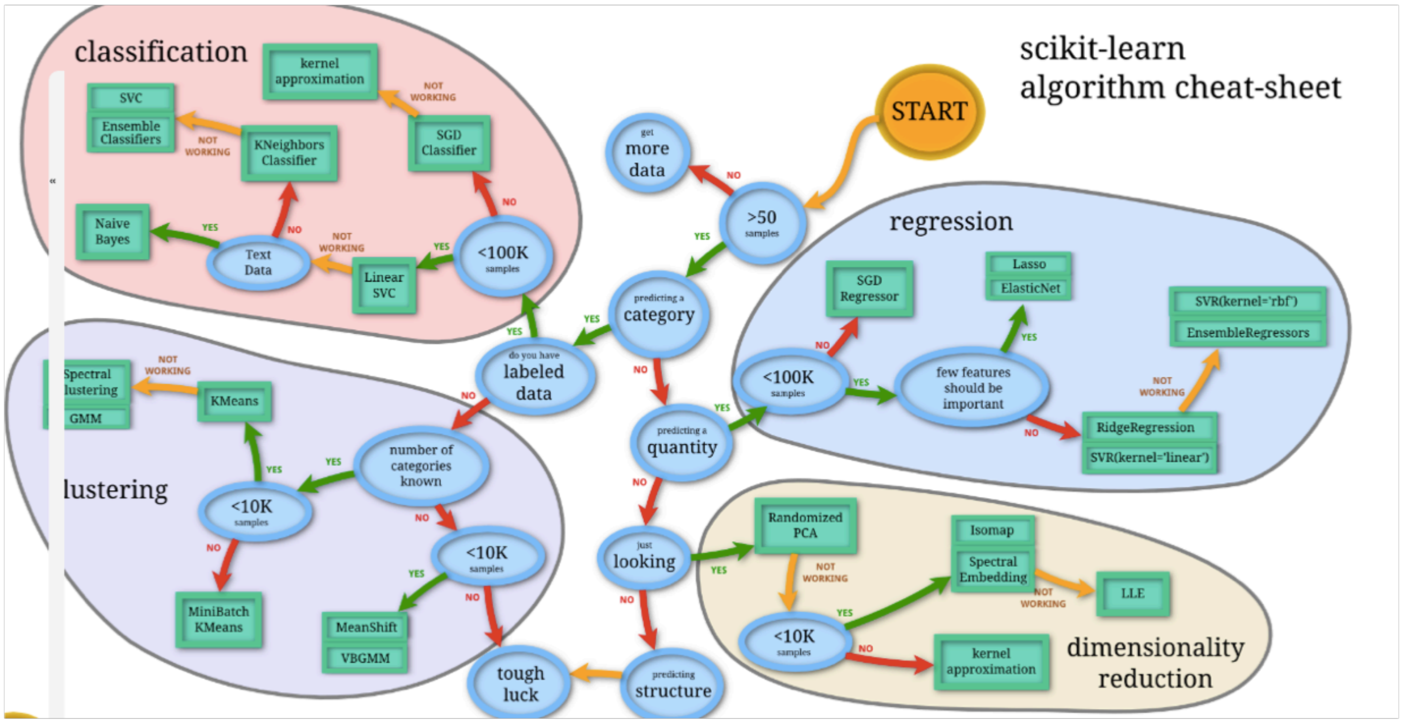




The boundaries learned by neural networks can be complex and irregular

Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

Scikit-Learn Algorithm



Algorithm	Accuracy	Training time	Linearity	Parameters	Notes
Two-class classification					
logistic regression		●	●	5	
decision forest	●	○		6	
decision jungle	●	○		6	Low memory footprint
boosted decision tree	●	○		6	Large memory footprint
neural network	●			9	Additional customization is possible
averaged perceptron	○	○	●	4	
support vector machine		○	●	5	Good for large feature sets
locally deep support vector machine	○			8	Good for large feature sets
Bayes' point machine		○	●	3	
Anomaly detection					
support vector machine	○	○		2	Especially good for large feature sets
PCA-based anomaly detection		○	●	3	
K-means		○	●	4	A clustering algorithm

Algorithm	Accuracy	Training time	Linearity	Parameters	Notes
Multi-class classification					
logistic regression		●	●	5	
decision forest	●	○		6	
decision jungle	●	○		6	Low memory footprint
neural network	●			9	Additional customization is possible
one-v-all	-	-	-	-	See properties of the two-class method selected
Regression					
linear		●	●	4	
Bayesian linear		○	●	2	
decision forest	●	○		6	
boosted decision tree	●	○		5	Large memory footprint
fast forest quantile	●	○		9	Distributions rather than point predictions
neural network	●			9	Additional customization is possible
Poisson			●	5	Technically log-linear. For predicting counts
ordinal				0	For predicting rank-ordering

Illustration Source: https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice

End of Section