ISA 414 – Managing Big Data

Lecture 12 – Data Analysis

Supervised Learning (Part I)

Dr. Arthur Carvalho

arthur.carvalho@miamioh.edu

MIAMI UNIVERSITY

Announcements

- Assignment 3
  - Now available on Canvas
  - Deadline: Wednesday,10/06, before 11:59 pm

Lecture Objectives

- Quick review of Homework 5 and 6

- Learn about predictive analytics
  - Classification and regression problems
    - Decision trees
    - Random forests

  - How to evaluate models
    - Training and test sets

Lecture Instructions

- Download the following files available on Canvas
  - *Lecture 12.ipynb*
  - *energy_data.csv*
  - *Description of the variables.xlsx*
- Place the above files inside the same folder
- Open the file *Lecture 12.ipynb* with VS Code

CRISP-DM

- Business Understanding phase
  - Important question: can a certain business problem be formulated as a data-analytics problem?
- Different types of problems/answers
  - Descriptive analytics: provides answers on "*what happened*"
    - Descriptive statistics, SQL queries, OLAP (business intelligence), …
  - Prescriptive analytics: provide answers on "*what could have happened*"
    - Optimization, simulation, …
  - Predictive analytics: provide answers on "*what will happen*"
    - Focus of this lecture

Business Understanding

- The solutions to the category of problems we will be working on are based on a paradigm called <u>supervised learning</u>
  - Assumption: there is a clear target variable whose values we are trying to <u>predict</u>

Our focus today → Classification

Next class → Regression

Problem	Supervised	Unsupervised
Classification	X	
Regression	X	
Clustering		X
Co-occurrence grouping		X
Profiling		X
…	…	…

Supervised Learning

- Classification (or class probability estimation)
  - Goal: determine which class a new observation (likely) belongs to
    - The target variable is qualitative (categorical/factor)
  - Example (Assignment 3): given a data set containing data about bank clients, can we build a statistical model that classifies future clients as “good” or “bad” in terms of paying back a loan?
    - Clear categorical target (two values: “good” and “bad”)

Supervised Learning

- Regression problems
  - Goal: predict the value of a target variable using a model structure that relates the target with informative attributes
    - The target variable is quantitative (numeric)
  - Example: what is the market price of a house in Oxford with 4 bedrooms, 3 bathrooms, built in 2016, …
    - Clear numeric target (price)

Supervised Learning

- Illustrative problem
  - An energy (electricity) provider has historical data with information about current clients
    - *E.g.*, Age, salary, marital status, *etc.*
  - Each client (observation) in the data set is associated with a category
    - Electricity tariff (*e.g.*, time-of-use tariff, peak tariff, flat tariff)
  - Business question:
    - Which electricity tariff should this company suggest to a new client?
      - Is this a classification or a regression problem?

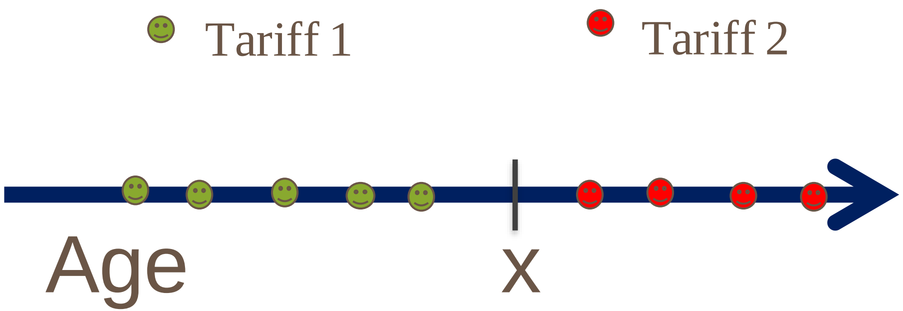
Classification

- Potential solution:
  - Translate this problem into a classification problem
  - Collect the data
    - *E.g.*, query internal databases
  - Build a *classifier*
    - A machine learning model that classifies clients based on tariffs
  - Classify the new client
    - Using the classifier to estimate which category the new client likely belongs to

Classification

- Consider the following hypothetical situation:
  - All clients younger than *x* in your data set prefer Tariff 1
  - All clients older than *x* in your data set prefer Tariff 2

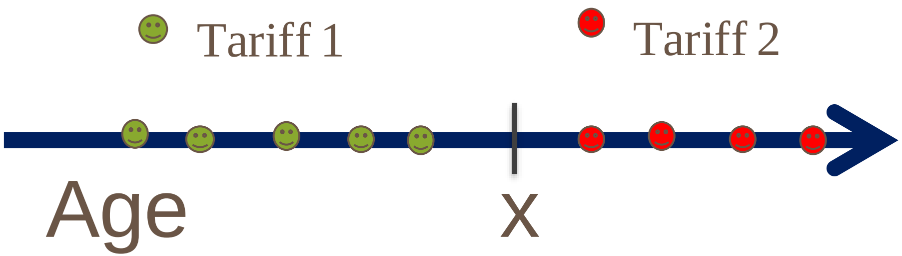
- What would be a good classifier?
  - Which tariff would you suggest to a new client?



MIAMI UNIVERSITY

Classification

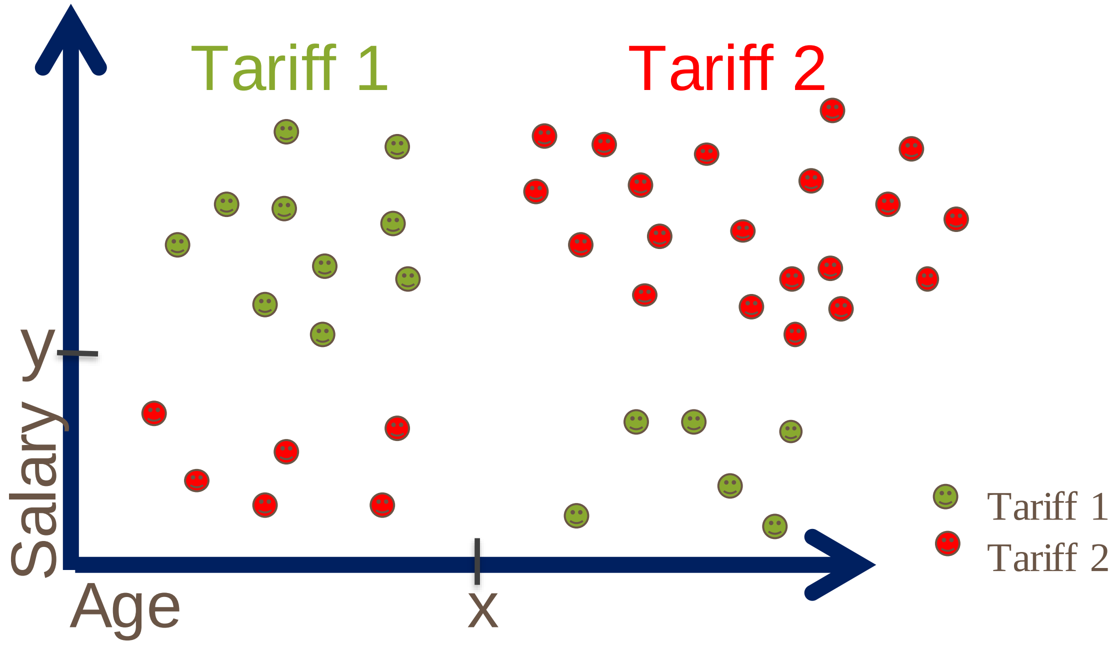
- Classifier:
  - New client’s age > x
    - Suggest Tariff 2
  - Otherwise
    - Suggest Tariff 1



MIAMI UNIVERSITY

Classification

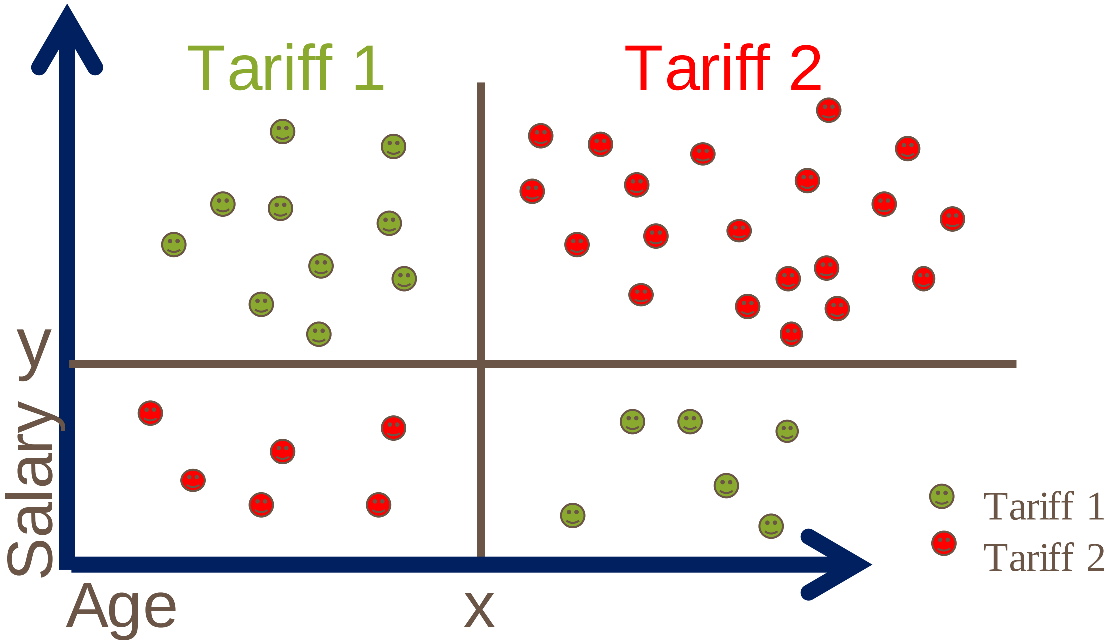
- What about this more realistic case?
  - Two variables



MIAMI UNIVERSITY

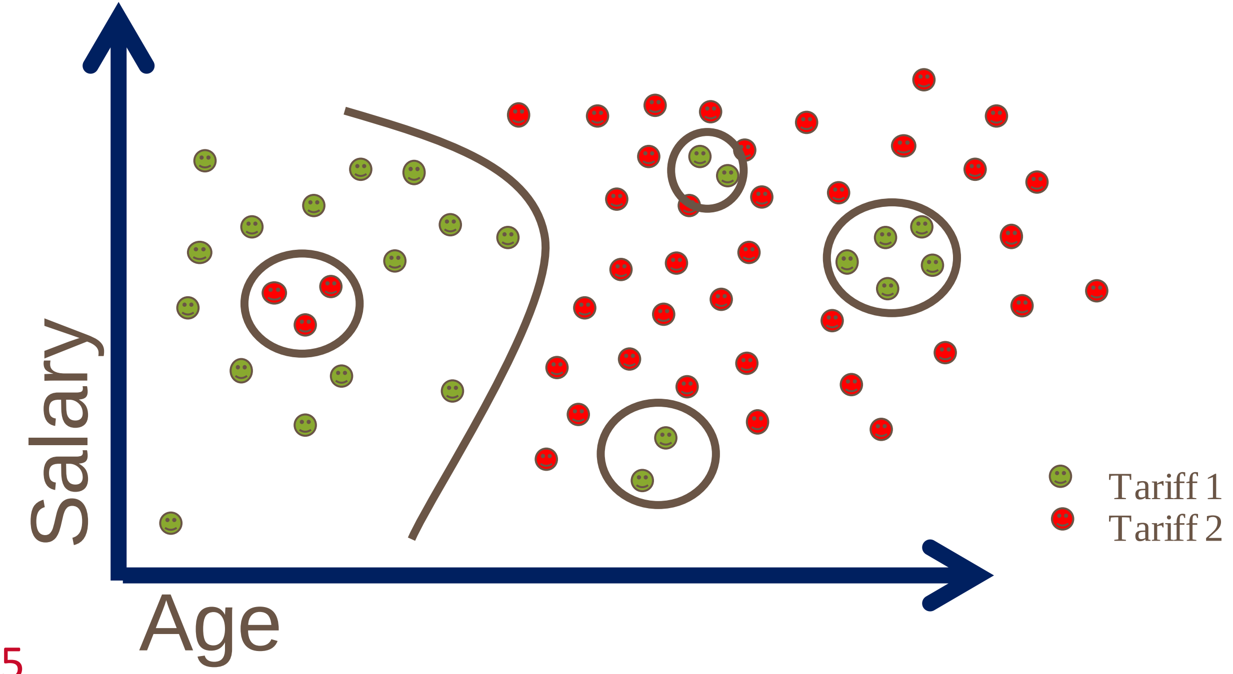
Classification

- Classifier:
  - New client's age > x AND new client's salary > y
    - Suggest Tariff 2
  - New client's age > x AND new client's salary < y
    - Suggest Tariff 1
  - New client's age < x AND new client's salary > y
    - Suggest Tariff 1
  - New client's age < x AND new client's salary < y
    - Suggest Tariff 2



Classification

- Keep in mind that real-life is a mess



MIAMI UNIVERSITY

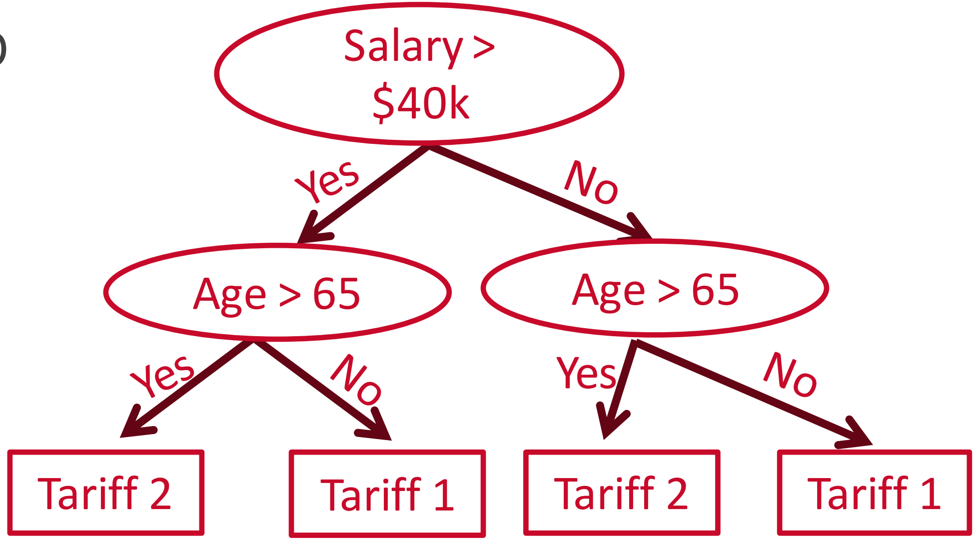
Decision Trees

- The previous approach of segmenting the attribute space is precisely what a technique called *Decision Trees* does
  - Create "if-then" rules

    E.g., IF 'Salary > $40K' AND

    'Age > 65'

    THEN suggest Tariff 2



Decision Trees

- Formally, decision trees create hyperrectangles
- Categories are the “leaves”
  - Bottom of the tree
- There are many algorithms for building decision trees from a data set
  - Beyond the scope of this course
  - General idea: choose a variable at each level that best splits the data set
    - *E.g.*, reduce entropy (information gain)

MIAMI UNIVERSITY

Decision Trees

- Henceforth, we shall heavily use the sklearn and pandas modules in our modeling efforts
  - Note that sklearn has several submodules that serve different purposes
    - We shall import them as we progress
  - Install the required modules

```
pip install sklearn
pip install pandas
```

  - Load pandas and the data set we use today

```
import pandas as pd
energy_data = pd.read_csv("energy_data.csv")
```

Decision Trees

- Before analyzing the data, always check whether each attribute is of the right type
  - If it is not, then change attribute types
  - Note that pandas store strings as objects

    energy_data.dtypes
- **sklearn** does not allow for non-numeric variables when building models
  - Its preprocessing submodule has several functions to transform categorical into continuous variables
    - Example: OrdinalEncoder(), OneHotEncoder(), LabelEncoder()

Decision Trees

- Oftentimes (not always) one should encode categorical variables as **dummies**
  - Except for the **target** variable, which should have its values replaced by numbers

- Let's derive dummies for the predictors
  - pandas offer a simpler way to derive dummies than sklearn

```
energy_data = pd.get_dummies(energy_data,
                             columns = ["MaritalStatus", "IncomeLevel", "DwellingArea",
                                        "HasChildren", "SolarRoof", "ShiftableLoad",
                                        "AttitudeSustainability" ],
                             drop_first = True)
```

Decision Trees

- Let’s derive recode the target variable
  - (Sub)module preprocessing in sklearn

```
from sklearn import preprocessing

enc = preprocessing.LabelEncoder()

energy_data["Tariff"] = enc.fit_transform(energy_data["Tariff"])
```


MIAMI UNIVERSITY

Decision Trees

- Building the tree model
  - One must create two sets of columns
    - The feature (independent) variables

```
x = energy_data.drop(columns=["Tariff"])
```

    - The target (dependent) variable

```
y = energy_data["Tariff"]
```

  - Next, it is time to create and fit a model

```
model = DecisionTreeClassifier()
model = model.fit(x,y)
```


MIAMI UNIVERSITY

Model Evaluation

- How do we know our model is any good?
  - One possible way: split the original data set into two parts
    - Train the model using one data set (training set)
    - Test the model using the complementary data set (test set)
      - Use the trained model to predict the target values in the test set, and compare the predictions against the true values
      - The higher the number of times the predictions agree with the true values, the more accurate the classifier is

Model Evaluation

- Analogy: exam
  - A professor gives you a study guide
    - Set of problems with answers
  - What if the professor asks you the same questions in the exam?
    - The professor is not really <u>testing</u> your knowledge
    - The professor is testing whether you are capable of <u>memorizing</u> answers
  - What if the professor asks you slightly different questions about the same material in the exam?
    - The professor is now <u>testing your knowledge</u> (generalization power)
  - <u>Study guide = training</u>; <u>Exam = testing</u>

Model Evaluation

- Let's redo what we did before and evaluate our model
  - Randomly split the original data into two data frames
    - **Training set** (66% of the observations)
    - **Test set** (34% of the observations)
      - The 66/34 division is just one common way of doing it
  - Train the model using the training set
  - Evaluate the model using the test set

Model Evaluation

- Let's use the submodule model_selection in sklearn to split a data frame

```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.34)
```

- Building a decision tree using the training set

```
model = DecisionTreeClassifier()
model = model.fit(x_train,y_train)
```

Model Evaluation

- Evaluating a decision tree using the test set
  - The metrics submodule contains several metrics to evaluate statistical models
  - We shall use the overall accuracy metric
    - Percentage of correctly classified instances

```
from sklearn import metrics
```

  - Step 1: use the model to predict the class of each observation in the test set

```
y_pred = model.predict(x_test)
```

  - Step 2: calculate how often the predictions in the model agree with the true class in the test set

```
metrics.accuracy_score(y_test, y_pred)
```

  - An accuracy of, say, 0.60 means that the model is expected to correctly classify 60% of future instances

Decision Trees

- Strengths
  - Simple to understand and interpret
  - Performs reasonably well for big data sets
- Drawbacks
  - Learning the optimal tree is not always computationally feasible
  - Might result in a high bias towards the training set
    - Tendency to **overfit**

Classification

- There are many models for classification
  - SVM, random forests, GBM, logistic regression…
  - Which one is the best?
    - In theory, all algorithms are equally good in expectation !!!
      - No Free-Lunch Theorem
  - Common approach when <u>predictive accuracy is the only important factor</u>
    - Build and evaluate multiple classifiers
    - Perform statistical analysis on the obtained results to determine the most accurate model

Classification

- Some models perform well for certain problems, and poorly for other problems
- Another common approach: combine several models
  - Ensemble learning
  - Compensate poor individual performance
    - (Almost) free lunch
  - Diversity matters
    - Formally, one wants the errors produced by the individual models to be as uncorrelated as possible

Random Forests

- Ensemble model
- Idea: build several decision trees semi-randomly
  - Each tree individually classifies a new observation
  - The most popular predicted category is chosen as the outcome of the model

MIAMI UNIVERSITY

Random Forests

- Building a random forest (200 trees) using the training set
  - Submodule sklearn.ensemble

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=200)
model = model.fit(x_train,y_train)
```

- Evaluating a random forest model using the test set

```
y_pred = model.predict(x_test)
metrics.accuracy_score(y_test, y_pred)
```

Random Forests

- One of the most popular models in forecasting competitions (alongside GBM and Neural Networks)
  - Knowledge Discovery in Databases (KDD)
  - Kaggle.com
- Strengths
  - Tackles the bias problem with single decision trees
- Drawbacks
  - No longer easy to interpret and explain the results

Classification

- This lecture summarized what is often taught across many data mining classes
- Keep in mind that:
  - There are many different statistical models for different types of problems
  - There are many different evaluation metrics other than using the percentage of correctly classified observations
    - *E.g.*, specificity, sensitivity, ROC area
  - There are many different ways of estimating model errors
    - K-fold cross validation, nested cross validation

Summary

- Summary
  - Data-analytics problems: classification problem
  - Decision trees and random forests
  - Evaluation: training and test sets
- Useful references
  - http://www.r2d3.us/visual-intro-to-machine-learning-part-1/
  - https://docs.google.com/presentation/d/1kSuQyW5DTnkVaZEjGYCkfOxvzCqGEFzWBy4e9Uedd9k/edit
- Next class: regression problems