1 Module 3: Introduction to Data Analysis

Contents

1.1 The Data Science Lifecycle

- The Lifecycle:
  - Start with a science question
  - Obtain data clean it
  - Interpret and understand the data
  - Understand the world
  - Make actionable recommendations

1.2 The Cross-Industry Standard Process for Data Mining (CRISP-DM) Framework

- Business Understanding
  - Determine Business Objectives:
    - Background & Success criteria
  - Assess Situation:
    - Requirements, Assumptions & Constraints
    - Risks & Contingencies
    - Terminology
    - Cost & Benefits

  - Determine Data Mining Goals:
    - Data mining goals & success criteria
  - Produce Project Plan:
    - Initial assessment of tools and techniques

- Data Understanding
  - Collect initial data
  - Describe data
  - Explore data
  - Verify data quality

- Data Preparation
  - Select Data:
    - Rational for inclusion/exclusion
  - Clean Data
  - Construct Data:
    - Derived data
    - Generated records
  - Integrate/Format Data

- Modeling
  - Select Modeling Technique:
    - Modeling techniques & model assumptions
  - Generate Test Design
  - Build Model:
    - Parameter setting
    - Model description
  - Assess Model:
    - Assess & revise parameter settings

- Evaluation
  - Evaluate Results:
    - Assess results wrt business success criteria

  - Review Process
  - Determine Next Steps:
    - i.e. Possible actions or a decision

- Deployment
  - Plan Deployment
  - Plan Monitoring and Maintenance
  - Produce Final Report/Presentation
  - Review Project & Documentation

1.3 pandas Basics

- Multi-param query requires "&" and parentheses:
  df[(df[str]=='str') & (df[float]==float)]
- df.query() for multiparam:
  e.g. df.query('Entity == "China" and Year == 2017')
- df.isin():
  df[df.isin(list)]
- df.query() for isin:
  e.g. df.query('Entity in @list_of_countries')

1.4 pandas Aggregation Operations

- df.groupby(parameter_to_sum:str) 'groups' the dataframe by that parameter
  - df.groupby(param:str).agg(sum)
  - .agg(max) and .agg(min)
  - .agg(function) – define and use a function to aggregate groupby

1.5 pandas Sorting with Aggregation

- df.sort_values(parameter, ascending=bool)

- Can combine .sort_values() and .groupby()
  e.g. df.sort_values('gdp').groupby('Year').agg(lambda x: x.iloc[0])
  Retrives the lowest gdp every year! (NOTE that the .agg element is equivalent to .first())

1.6 pandas Indexing

- .set_index(column_name) or .set_index([col_names ,...])
  (Notice that using .groupby(col) automatically indexes by col)

- .reset_index()

1.7 pandas Filtering

- df.dropna() Drops any row with NaN

- df.filter(function)
  e.g. function = lambda x: max(x) > 10
  (Returns bool for whether the column's max is >10)