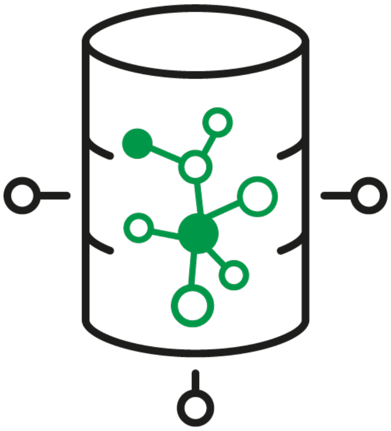

Cadent Data Science

Python Coding Benchmark

Cadent's data science team is responsible for research and development of machine learning models in a variety of business areas. In order to work more effectively as a team, we share a set of core tools from the python ecosystem, primarily, pandas for data manipulation, scikit-learn for machine learning, numpy & scipy for scientific computing and matplotlib & seaborn for data visualization.

This benchmark is designed to verify your ability to apply your existing data skills within our team's core tool kit. You are welcome to use online resources as references, but please stick to the guidelines provided. This phase of the interview process is better served by a good solution following the methods outlined than a great solution that deviates from them. Also, while we use a broader selection of tools on the team, the restrictions on package usage for this benchmark is intentional and submissions that do not comply will be excluded from consideration.

Assignment

Train a Machine Learning Model of your choosing to predict survival in the *Titanic Dataset*. Please limit packages used to *pandas*, *sklearn*, *numpy*, *scipy*, and *matplotlib/seaborn.*

Deliverable

The final deliverable to "hand in" is a jupyter notebook (ipython notebook) that follows the data science process to train and evaluate a model. Performance will be assessed via adherence to the guidelines and the ability to explain one's reasoning in a follow-up session.

Requirements

- Use the **Jupyter** notebook IDE
- Use **pandas** to read the csv data
- Clear labeling of the notebook's sections
- Concise, commented code
- MUST use the public **Titanic dataset**
  - from the Kaggle Competition
- Minor amount of feature engineering-- no need to go overboard.
- Final Evaluation on a held out test set; include **precision / recall curve**
- Must use **nested cross validation** on the training set for model selection to estimate generalization:
  - Tune model parameters with **GridSearchCV**
  - Evaluate a classification metric with **cross_val_score**
- Comments comparing the nested CV score and the final evaluation score.
- Any "scientific" data preprocessing (i.e. PCA) classes must be placed in a Pipeline, **sklearn.pipeline**, with the final classifier