# Data Science Internship Assessment

## Problem Statement

Novelis Data Science team is building a data science platform which provides framework for end-to-end implementation of machine learning pipelines. The framework will be built using python in an object oriented programming style. Furthermore, we want to implement everything using the scikit learn API, all implementations confined to fit, transform, and predict methods. This will allow us to implement machine learning pipelines using the scikit learn pipelines. As such, the assessment will evaluate candidates’ ability to transform codebase prepared in procedural programming to object oriented programming, specifically using the scikit learn API as much as possible.

## Instructions

Assessment folder contains codebase prepared in procedural programming to solve a classical binary classification problem. Among multiple notebooks, there is a notebook named “2DataScienceAssessment\_Fraud\_Detection\_FeatureEngineering.ipynb” which prepares a python script for feature engineering.

Candidates are required to transform each task in the code (e.g., calculating the fraud ratio, label encoding etc.) into separate classes containing fit, predict, transform, and fit\_transform methods. Please make a note that some of these methods do not necessarily be executed in every class. Finally, candidates are required to combine these classes using sklearn’s pipeline functionality into a feature engineering pipeline.

Eventually, with the transformed notebook, and hence the python script, for feature engineering, candidates should be able to replicate the results comparable to “Final\_Notebook\_fraud\_detection.ipynb”.

As examples, candidates may refer to

* [https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html [scikit-learn.org]](https://urldefense.proofpoint.com/v2/url?u=https-3A__scikit-2Dlearn.org_stable_modules_generated_sklearn.pipeline.Pipeline.html&d=DwMFaQ&c=1BiSaOvZRfbfOVK9K1qpLA&r=bqFVSSy_iDChJT-FjnVeihBB7ieX-2ricQHEc_NQp7wRO9_6o1-q6T70az2B5mrK&m=QV2tojive52BpDn2rz75B-ysjSqApw_WcID0mJyqTpw&s=EiMBDguxBdGOWOn2uHt3oXshm1We8De4hmXjDMJ2WoU&e=)
* [https://scikit-learn.org/dev/developers/contributing.html#rolling-your-own-estimator [scikit-learn.org]](https://urldefense.proofpoint.com/v2/url?u=https-3A__scikit-2Dlearn.org_dev_developers_contributing.html-23rolling-2Dyour-2Down-2Destimator&d=DwMFaQ&c=1BiSaOvZRfbfOVK9K1qpLA&r=bqFVSSy_iDChJT-FjnVeihBB7ieX-2ricQHEc_NQp7wRO9_6o1-q6T70az2B5mrK&m=QV2tojive52BpDn2rz75B-ysjSqApw_WcID0mJyqTpw&s=YwQ-yB0frENT9sKpRp9CS61-O997j88_9KCSkRea3U8&e=)
* [http://danielhnyk.cz/creating-your-own-estimator-scikit-learn/ [danielhnyk.cz]](https://urldefense.proofpoint.com/v2/url?u=http-3A__danielhnyk.cz_creating-2Dyour-2Down-2Destimator-2Dscikit-2Dlearn_&d=DwMFaQ&c=1BiSaOvZRfbfOVK9K1qpLA&r=bqFVSSy_iDChJT-FjnVeihBB7ieX-2ricQHEc_NQp7wRO9_6o1-q6T70az2B5mrK&m=QV2tojive52BpDn2rz75B-ysjSqApw_WcID0mJyqTpw&s=MQ9h-IaOWA4qnw5VaI19ZeUXlo-Z2jnR92O5OjjhxVU&e=)

## Steps

1. Run “Final\_Notebook\_fraud\_detection.ipynb”. This will create two joblib files in the working directory.
2. Transform each task, e.g. fraud ratio calculation, label encoding etc., “2DataScienceAssessment\_Fraud\_Detection\_FeatureEngineering.ipynb” into separate classes with fit, transform, predict, and fit\_transform methods. Note that not all of these methods need to be executed in every class. Save the new notebook as “2DataScienceAssessment\_Fraud\_Detection\_FeatureEngineering\_OOP.ipynb”. Remember to change the filename in the first cell of this notebook from “myScripts/featureengineering.py” to “myScripts/featureengineering\_OOP”.py
3. Create a copy of “Final\_Notebook\_fraud\_detection.ipynb” as “Final\_Notebook\_fraud\_detection\_OOP.ipynb” and import the new class you created instead of feature engineering.
4. Run “Final\_Notebook\_fraud\_detection\_OOP.ipynb” and compare results from step 1.