Credit Scoring with Python

Credit Scoring with Python

A **catalog of python packages** that can be used for building a Credit Scorecard.

Motivation and Scope

The objective is to assist with the development of digital Credit Scoring processes that are built around open source software. There is currently no single python framework that covers the full Model Development and Model Validation of Credit Scoring Models, especially not to a standard that would be required if such models where to used in actual production environments. The catalog aims to associate the available functionality of various existing packages with the various steps of the model development / validation process.

The focus of the the catalog is on the variety of statistical scoring models that can be developed quantitatively using historical performance data.

Out of scope are:

- low data approaches such as Expert Based Models
- models that use Financial Market Information such as observed credit spreads
- Credit Rating Model

Catalog of Python Libraries

Data Collection

Data Collection is a highly context dependent process (depends on the existing systems, databases and their schemas, operating environments etc that hold credit data). Hence it is not possible to pin down concrete packages that would be sufficient in every case. The table is thus only indicative

Procedure	Pandas	Scikit-learn	Other	Remarks
Connect to SQL database	read_sql		SQLAlchemy	In-memory only workflows, see Dask for scaling
Connect to NoSQL database			pymongo	
Load from csv / tsv files	read_csv	numpy arrays only	csv package	sklearn.datasets.load_files
Load from json files	read_json	numpy arrays only	json	
Load from xls / xlsx / ods files	read_excel		xlrd, openpyxl	
Merge, join, transform operations	dataframe operations			

Data Review

Risk Data Review is a collection of procedures that aim to

- establish Data Quality and, where appropriate, identify actions that will improve it
- perform Exploratory Data Analysis to help generate insights about the available data sets

The objective of these procedures is to create a collection of data objects that will conceptually support the next step of identifying useful features.

Procedure	Pandas	Scikit-learn	Other	Remarks
Descriptive Statistics	DataFrame.describe, pandas_profiling		stats.describe	
Visualization	matplotlib API, pandas.plotting	matplotlib API	Yellowbrick	

Data Cleansing

Data Cleansing is a collection of procedures that aim to

- Imputing missing Values
- Correcting wrong Representations (e.g. Date, Percentages)
- Correcting wrong Values (Summations)

The objective of these procedures is to create a collection of data objects that will practically support the next step of identifying useful features by making sure that all quantitative data meet quality requirements for automated processing.

Procedure	Pandas	Scikit-learn	Other	Remarks
Missing Data	fillna (via numpy datatypes)	sklearn.impute		numpy.NaN is the fundamental missing data datatype

Feature Selection and Engineering

Feature engineering is the process of using domain knowledge of the data to create features that make machine learning algorithms work. In the context of credit scoring, a feature is usually a Credit Score Factor, that is a variable (attribute, characteristic) that has a potential relationship with the outcome variable. The objective of these procedures is to create a *master data table* that meets the criteria for supporting the main model development step.

Procedure	Pandas	Scikit-learn	Other	Remarks
Feature Selection		sklearn.feature_selection		
Standardization / Normalization		sklearn.preprocessing		
Transformations		sklearn.pipeline	featuretools	

Model Selection and Fit

Model selection and fit is the process of selecting a suitable model (and model hyperparameters) and the subsequent estimation of the model. We assume here that the selection process in made manually

Procedure	Pandas	Scikit-learn	Other	Remarks
Model Fit		scikit-learn.model.fit		

Model Validation

Model Validation of a Credit Scorecard aims to contain the Model Risk associated with using the scorecard (the potential for error in the development and implementation of the model and/or the application or interpretation of model results). The nature of possible errors is linked to the nature of the model and its production use (e.g. Type I / Type II classification errors when accepting a new client)

Procedure	Pandas	Scikit-learn	Other	Remarks
Cross -Validation		sklearn.cross_validation		
Accuracy Score		sklearn.metrics		

NB: Model Selection, Fit and Validation may be an iterative process

Model Deployment

Model Deployment entails (in its most basic form) to make available the credit scorecard to users. It is very common that developed scorecards are re-programmed in other languages when deployed in production. Here we only cover some options of using python also as a deployment platform.

Procedure	Pandas	Scikit-learn	Other	Remarks
As a desktop application			PyQT, wxWidgets, Kivy	
Attaching to a web service			Flask, Bottle, Django	

Further Notes

Criteria for Inclusion

- We make no differentiation by license (provided it is an open source license)
- We make no detailed assessment of maturity / testing (we will revisit this in due course)
- Initial preference is to the "well known" / widely used projects of the python ecosystem
- For some tasks there might be multiple packages that offer the same functionality. In those instances we will want to catalogue all good alternatives.

Method

The structure of the catalog is to decompose the required functionality in a roughly linear fashion following the steps of the Risk Model Lifecycle (see also How to Build a Credit Scorecard). In practice these steps might be performed by different teams, at different times, in different sequences, using different tools etc. **It is not for this entry to document best practices in this respect.**

Issues and Challenges

- Actual credit scorecards may vary significantly in structure as they need to operate in different organizational, operational and regulatory contexts. The aim here is to capture a typical quantitative and in particular machine learning oriented development workflow that uses modern open source libraries available for that purpose
- This entry is not a credit model catalog. There is a separate entry for that purpose, although in the model development segment we do list the packages that offer relevant models for credit scorecard development
- This entry is not a workflow for complete end-to-end credit scorecard development. How to Build a Credit Scorecard is a separate entry that covers that task
- Many tasks in the catalog can be coded from scratch (using e.g numpy) instead of using an existing python library. Using a package introduces additional dependencies but also reduces the risk of code errors and speeds up development.

See Also

- Credit Scorecard
- How to Build a Credit Scorecard
- Credit Scoring Models
- Open Source Data Quality Software
- Open Source Risk Management Software