Evaluation for Data Scientist – Focus on Python

Competence framework for evaluating the final project (qualitative evaluation)

Name

Date

Step		Minimum standard	Optimal standard
Preparation	Load data	creates DataFrames from csv_file	uses optional function parameters to minimize work steps
EDA	Understand your data	uses mean value, median, minimum, maximum and standard deviation when appropriate	also carries out data exploration on appropriate subsets of the entire data set. (grouping)
	Visualize Distributions	☐ applies at least one of the following diagram types to all numeric columns: ***Box plot, histogram, distribution plot, scatter plot*** ☐ Applies at least one of the following diagram types to all numeric columns: ***Bar chart, Pie chart***	☐ visualizes at least 2 subsets of the entire data set in the same figure to compare the distribution of the subsets with one another ☐ uses dimensionality reduction methods to visualize high-dimensional data
	Detect Outliers	☐ Identifies outliers visually	☐ Identifies outliers using machine learning methods (e.g. DBSCAN, RANSAC)
	Check Correlations	☐ explores correlations between features	☐ visualizes correlations
Split	Training/test split	☐ performs a training/test split as early as possible	☐ The test data set is only used once to the evaluate the model quality.
Data Cleaning	Transforming data types	☐ converts ***int, float, str, datetime*** into other formats depending on the situation	☐ uses string manipulation methods ☐ defines own data cleaning functions and applies them.
	Data Imputation	☐ Decides according to the situation whether to delete, replace or mark missing values ☐ Applies data imputation with '***mean***' or '***mode***'	☐ uses external sources, robust metrics or modelling methods for data imputation.
	Dealing with Outliers	☐ Decides according to the situation whether to delete, replace or mark missing values ☐ Only removes outliers from the training set	

Step		Minimum standard	Optimal standard
Resampling	Over/Undersample	☐ uses the ***class_weights*** parameter for unbalanced target categories	☐ Implements explicit over/undersampling methods depending to the situation
Feature Engineering	Polynomials & Interactions	☐ Creates new features by combining existing features	☐ uses polynomials and interaction terms to generate features ☐ Integrates data from external sources to enrich the data set
	Encoding	☐ applies ONE-hot encoding to nominal features ☐ applies label encoding according to the situation	☐ implements a way to deal with unknown categories outside the training set
	Data Scaling	☐ uses data transformation methods (standardization, MinMax scaling) appropriately	
	Dimensionality Reduction	☐ uses dimensionality reduction methods to create new features depending on the situation	☐ interprets the composition of the principal components
	Feature Selection	☐ uses model-immanent feature importance methods ☐ carries out justified feature selection	☐ uses model agnostic methods to assess the feature importance if necessary ☐ proceeds systematically when evaluating features.
Modeling	Model Training	☐ uses a simple classification model (KNN, Logistic Regression, Decision Tree)	☐ evaluates the use of ensemble classifiers ☐ Uses bagging and/or boosting methods ☐ Evaluates the use of ANN
	Evaluation Metrics	☐ Performs at least one training-validation split in the training data ☐ Uses a single evaluation metric ☐ Justifies the choice of evaluation metric based on the nature of the problem	☐ Uses a weighted selection of several evaluation metrics

Step		Minimum standard	Optimal standard
		makes qualitative statements about the quality of the implemented model... ... and justifies this based on the evaluation metric(s)	
	Hyperparameter Tuning	manually tests several value combinations for hyperparameters	uses cross-validation in model evaluation uses GridSearch to automate hyperparameter tuning uses pipelines to modify the data (e.g. scaling, PCA ...) as well as to optimize the model
	Model Selection	explicitly determines which of the tested models is the final model	evaluates model performance against a baseline model or dummy model
Model Interpretation	Model Interpretation	explains in the consultation the influence of a hyperparameter on how a ML model functions	explains the influence of a model parameter on the prediction result
	Sanity checks	a prediction is made for **each** data point in the test set (*'**aim-set**'*). The original of the data set provided is not modified. the process presented minimizes ***data leakage***	the data pipeline is able to generate a prediction for a single data point with partially missing data. ***Data leakage*** does not occur.
	Visualizing Features Importances & Predictions	Uses suitable diagram types for the visualization all visualizations contain at least one meaningful title... ... and axis labels	creates an interactive visualization visualizations of the results follow the principles of data storytelling

Step		Minimum standard	Optimal standard
Clean Up	Create Functions	the prediction is generated with a single function call.	
	Refractor Code	structures the project submission using headings and comments. uses meaningful names for variables, functions and classes	Code mainly follows the PEP 8 standard
Storytelling	Presenting your Find-ings	Leads the way through the **functioning** code in the project consultation Explains important modelling decisions (e.g. dealing with outliers, choice of metrics, feature selection...) answers questions technically	Delivers an introductory summary: the solved problem the solution they worked out model performance of the final model