AUTOMATED DOCUMENTATION OF END-TO-END EXPERIMENTS IN DATA SCIENCE

Sergey Redyuk, TU Berlin, Germany sergey.redyuk@tu-berlin.de
supervised by Volker Markl (TU Berlin) and Sebastian Schelter (NYU)

Abstract

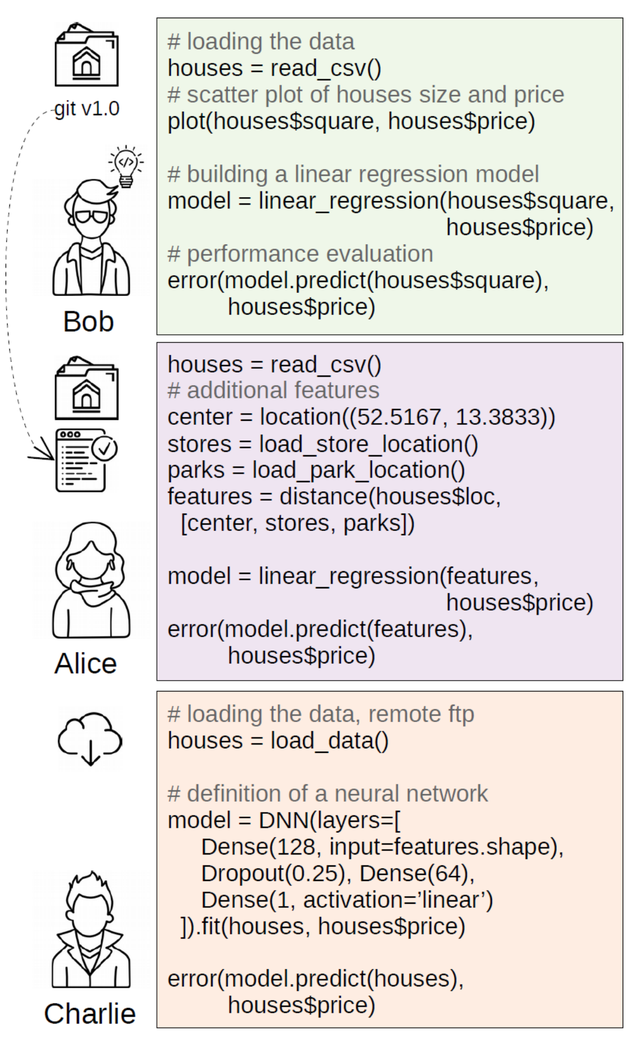
Data-oriented experiments are **hard to reproduce** due to rapid undocumented changes, abundance and inconsistency of tools and frameworks, multi-tenant environment

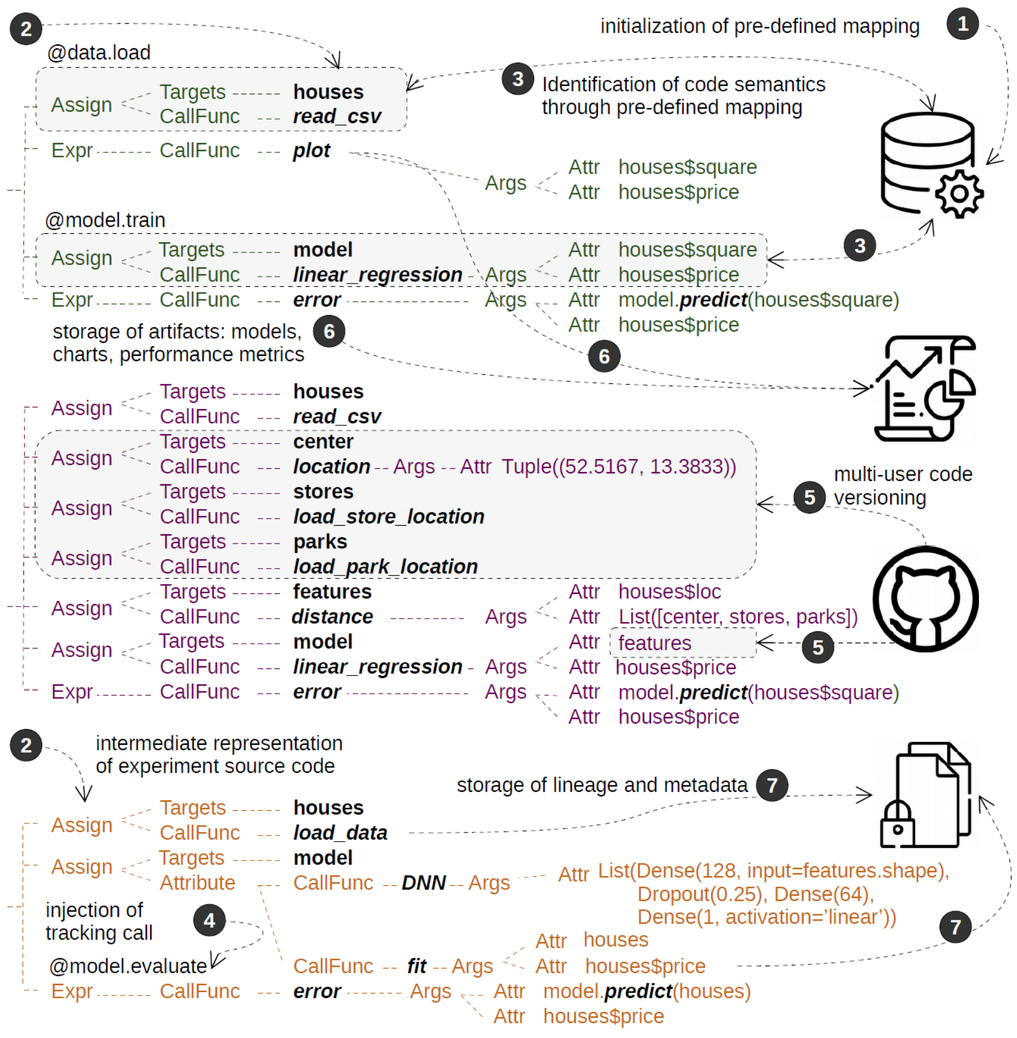
Goal: end-to-end tracking of workflow provenance and meta information, to achieve reproducibility, sharing and reuse of intermediate artifacts across teams and domains

Contributions

- automated tracking of workflow provenance and metadata at runtime via source code analysis
- extraction of a declarative high-level representation of a data science experiment
- design and population of an experiment database to enable search and reuse capabilities over previous experiments

Use Case

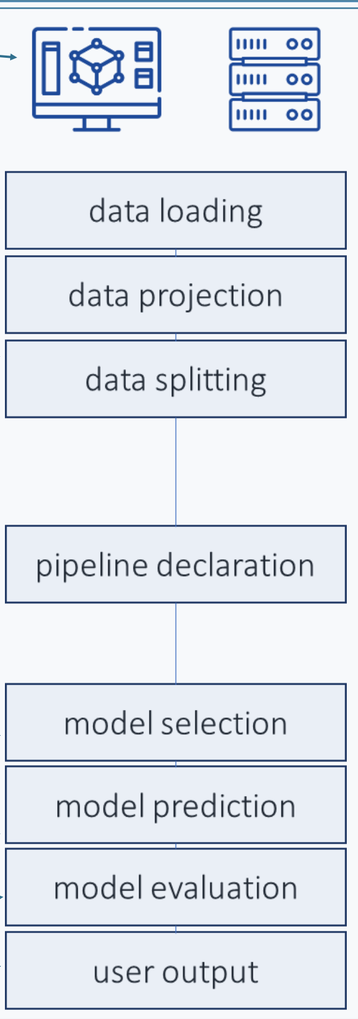




1. populating a predefined mapping that matches function signatures to a high-level data science task (manually, based on user documentation);
2. extracting stringent intermediate representation of the source code;
3. querying the predefined mapping to identify source code semantics;
4. capturing detected workflow provenance and corresponding meta information, required for reproducibility;
5. automated multi-tenant version control system to detect changes in source code and assign them to a particular high-level data science task;
6. persisting the intermediate artifacts, e.g. ML models, figures, user output;
7. persisting workflow provenance and metadata associated with a run of the experiment.

Open Questions and Challenges

```
1  from sklearn.model_selection import train_test_split, GridSearchCV
2  …
3  data_filepath, target = '…', 'class'
4  data = pd.read_scv(data_filepath)
5  columns = list(data)
6
7  X, y = data[[col for col in columns if col != target]], data[target]
8  X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=.1)
9
10 categorical_cols = ['workclass', 'occupation', 'marital_status']
11 numeric_cols = ['hours_per_week', 'age']
12 feature_transformation = ColumnTransformer(transformers=[
13    ('cat_features', OneHotEncoder(handle_unknown='ignore'), categorical_cols),
14    ('scaled_numeric', StandardScaler(), numeric_cols)
15 ])
16 pipeline = Pipeline([
17    ('features', feature_transformation),
18    ('learner', SGDClassifier(max_iter=1000, tol=1e-3))
19 ])
20 param_grid = {
21    'learner__alpha': [0.0001, 0.001, 0.01, 0.1]
22 }
23
24 search = GridSearchCV(pipeline, param_grid, cv=5)
25 model = search.fit(X_train, y_train)
26 predicted = model.predict(X_test)
27 acc = accuracy_score(y_test, predicted)
28 print("TRAIN. accuracy: %.4f" % (acc))
```



Evaluation

- manually annotated example scripts where ground truth is available, e.g. logical structure of the experiment, ML model declaration, hyperparameters etc.
- example scripts collected from public repositories where ground truth can be partially inferred, e.g. jupyter notebooks that are logically splitted into blocks
- metrics: achieved reproducibility (binary), overhead as a diff in execution time

Semantics

- difficult to infer, e.g. low-level implementations
- the task on arbitrary source code is so far infeasible
- experiment source code is a combination of high-level APIs (e.g. pandas ecosystem, keras) and low-level imperative statements
- solutions: (i) frequent pattern mining, and (ii) evaluation of control flow

We would like to acknowledge the support of the Helmholtz Einstein International Berlin Research School in Data Science (HEIBRiDS), Max Delbruck Center for Molecular Medicine (MDC), TU Berlin, and the Berlin Center for Machine Learning (BZML) 01IS18037A. We thank Sebastian Schelter (NYU) and Volker Markl (TU Berlin) for guidance and valuable contributions.

Berliner Zentrum für MASCHINELLES LERNEN
Technische Universität Berlin
HEIBRIDS Helmholtz Einstein International Berlin Research School in Data Science
MDC MAX-DELBRÜCK-CENTRUM FÜR MOLEKULARE MEDIZIN IN DER HELMHOLTZ-GEMEINSCHAFT