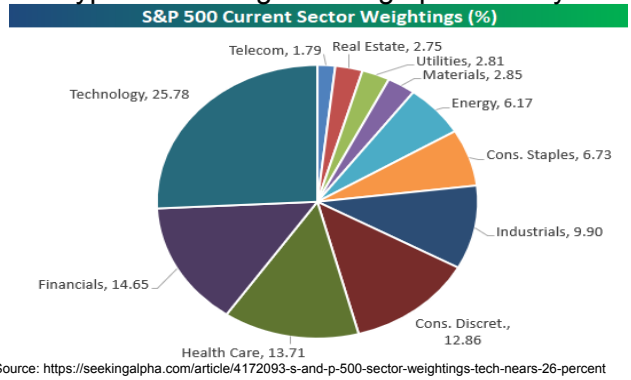
Extracting Types from Python Machine Learning Libraries

Ana Milanova, Collin Jones, Andrew Ma (RPI) Julian Dolby, Martin Hirzel (IBM)

Why extract types?

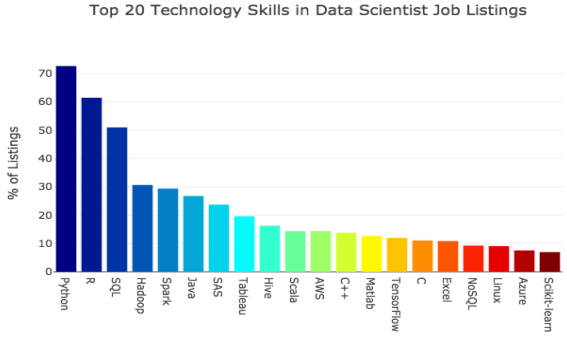
- AI skill demand exceeds supply, need to increase productivity
- AI for business must be trustworthy, need to avoid bugs
- Types avoid bugs with high productivity



Source: https://seekingalpha.com/article/4172093-s-and-p-500-sector-weightings-tech-nears-26-percent

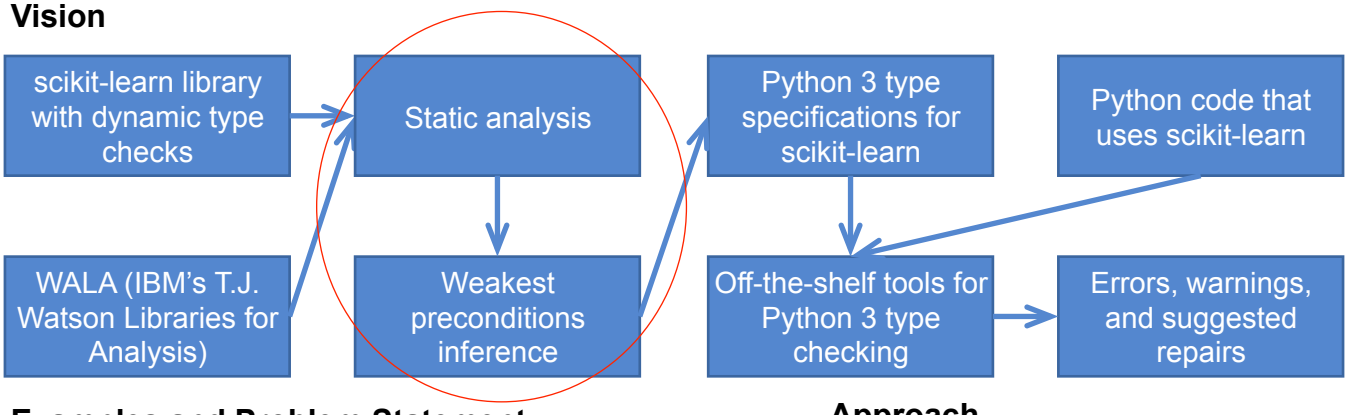
Why Python?

- Widely used for machine learning
- Until recently, no static type checks
- With Python 3, optional static types, but popular machine learning libraries such as scikit-learn do not use them yet



Source: https://towardsdatascience.com/the-most-in-demand-skills-for-data-scientists-4a4a8db896db

Vision



Examples and Problem Statement

Example 1:

```
// Error triggered by parameter dependences
// Path condition: svd_solver == "full" &&
//  !(0 <= n_components <= num_columns(X))
X = [[0,1],[-1,2],[3,4],[2,3]];
…
pca = PCA(n_components=4,
                    svd_solver="full");
pca.fit(X);
```

↓ _fit(X);

↓ _fit_full(X);

↓ ValueError!

Example 2 (adapted from web, qa.ru):

```
// Error triggered by input test_x: must be either a
// sparse matrix or a well-shaped (array) sequence
// Path condition:
      !isSparse(text_x) && !isArraySeq(test_x)
test_x = …;
…
lr = LogisticRegression(C=4, dual=True)
lr.fit(…)
… = lr.predict_proba(test_x)[:,1]
```

↓ _predict_proba(X) // in file logistic.py

↓ _predict_proba_lr(X) // in file base.py

↓ _decision_function(X) // in file base.py

↓ _check_array(X, accept_sparse="csr");
// in file validation.py

↓ ValueError!

- Errors occur far from where they are introduced
- Problem statement: "lift" path conditions that trigger error (deep in ML library code) to conditions on user inputs

Approach

- Symbolic execution techniques: weakest precondition inference using novel inter-procedural backward reasoning
  - Entails highly-precise analysis
  - scikit-learn is amenable to such highly-precise analysis
- Immediate scope: "lift" dynamically checked ValueErrors to conditions on user inputs. Catch errors early and suggest repairs
- Broader scope: generalized type extraction for ML libraries and Python