CIVIS

Software Library APIs: Lessons Learned from scikit-learn

Liz Sander, Data Scientist, Civis Analytics
GitHub: elsander
@sander_liz

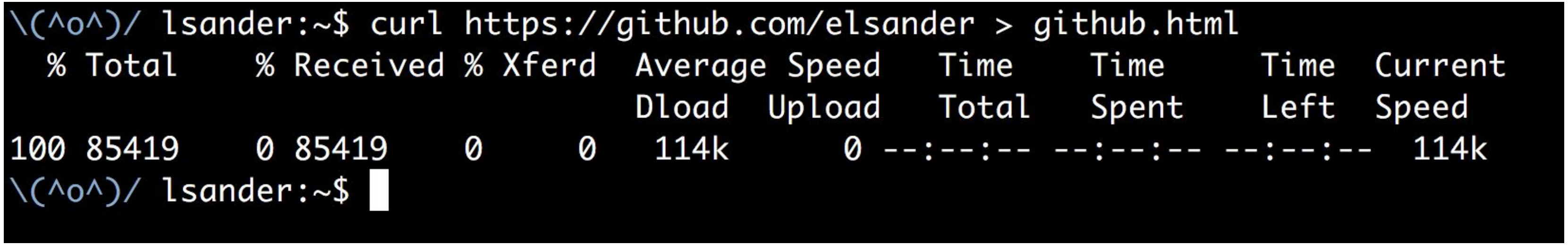
AGENDA

- Introduction to APIs
- Scikit-learn API
- Extending scikit-learn

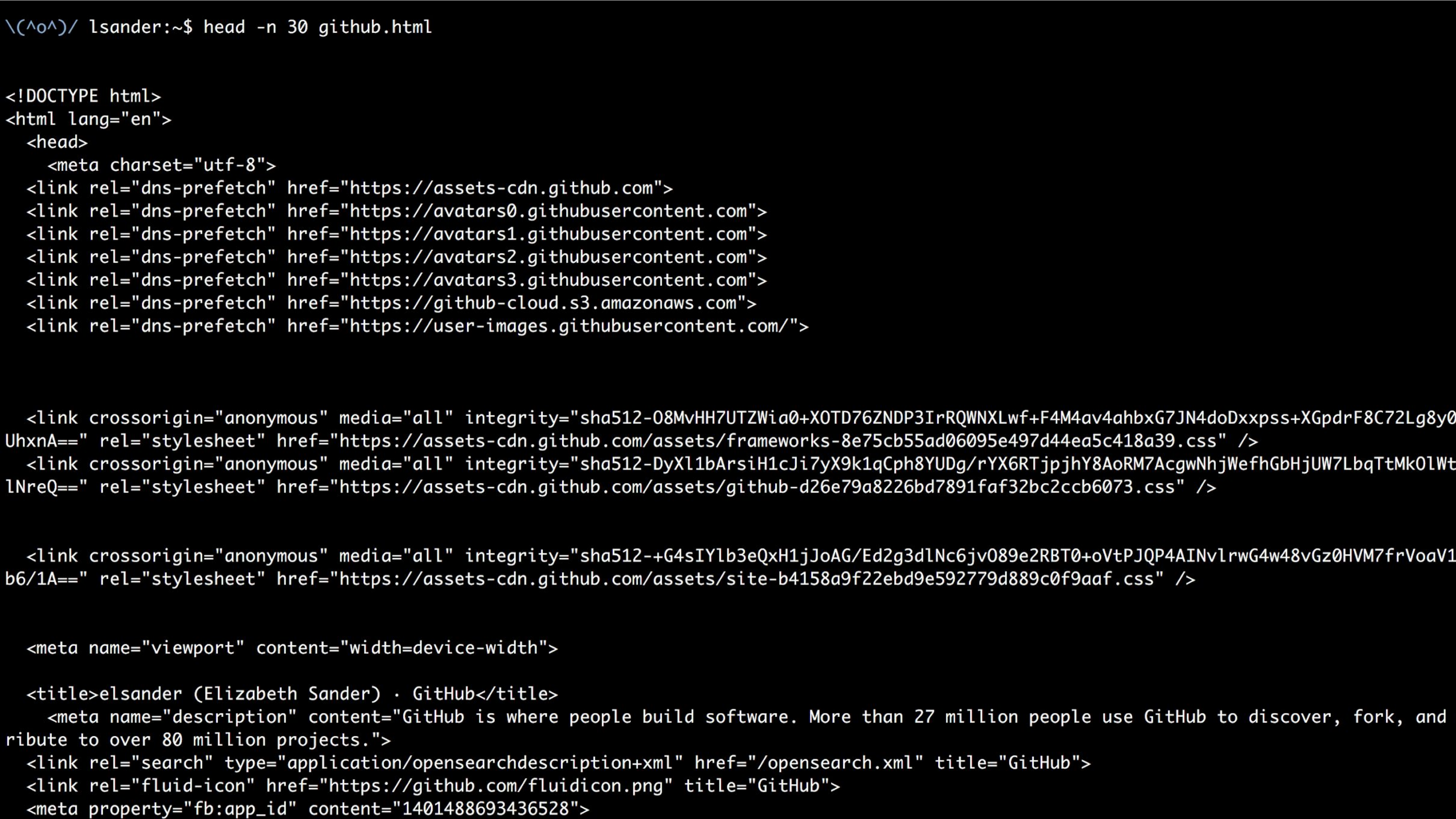
What is an API?

What is an API?

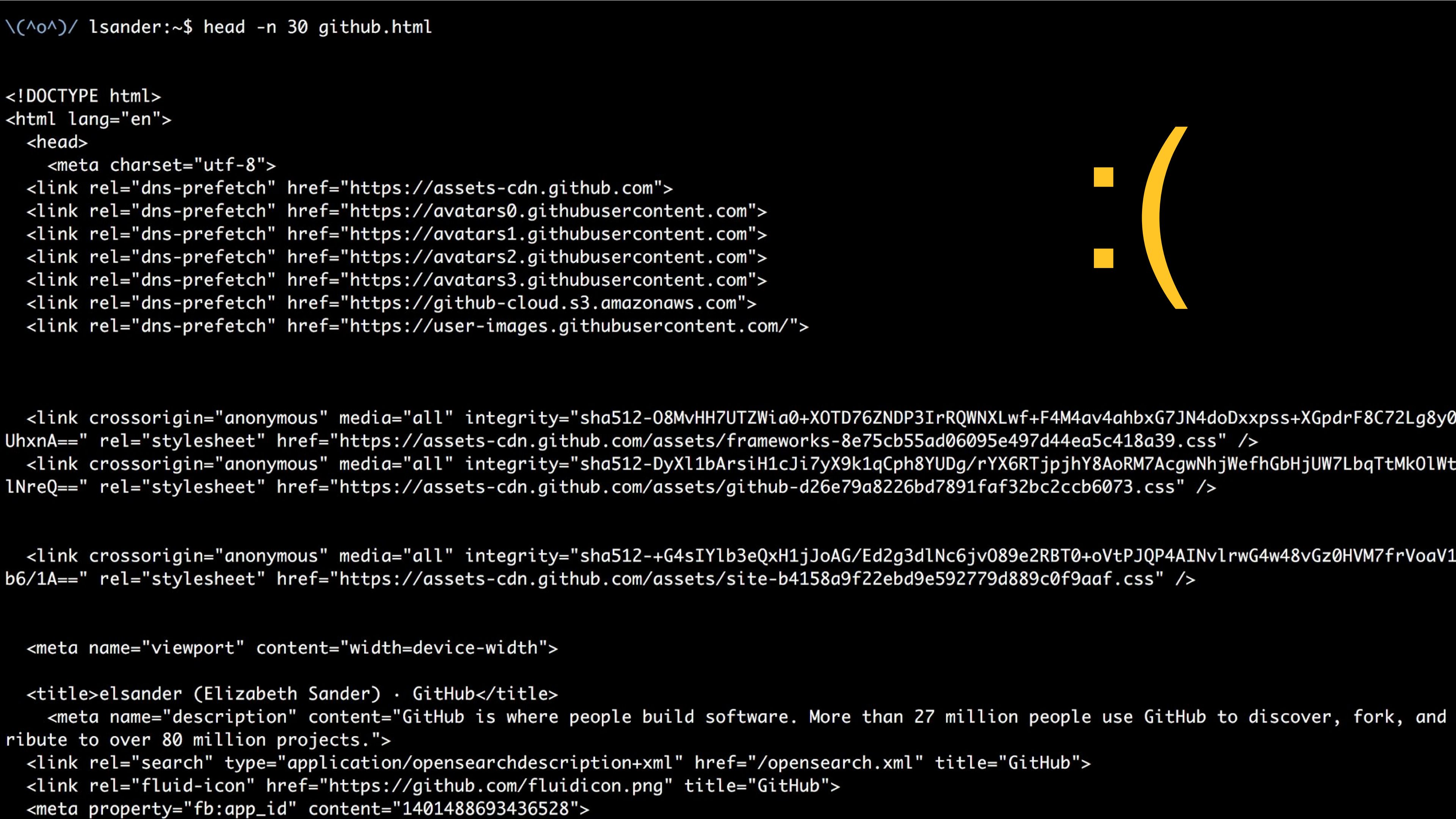
- A website API is an interface between website and developer



```
\(^o^)/ lsander:~$ curl https://github.com/elsander > github.html
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 85419    0 85419    0     0   114k      0 --:--:-- --:--:-- --:--:--  114k
\(^o^)/ lsander:~$ 
```

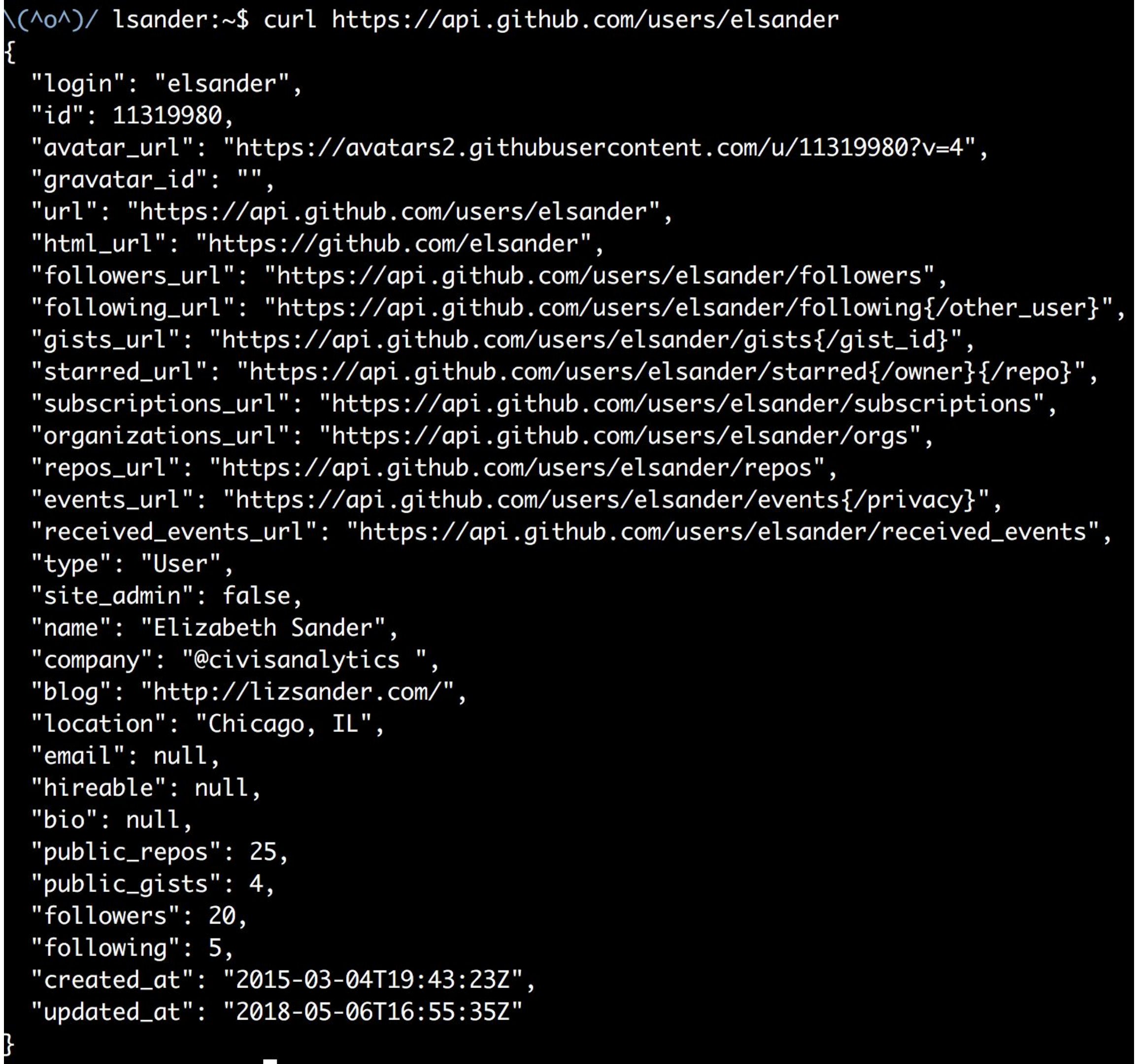


```
\(^o^)/ lsander:~$ head -n 30 github.html


<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
  <link rel="dns-prefetch" href="https://assets-cdn.github.com">
  <link rel="dns-prefetch" href="https://avatars0.githubusercontent.com">
  <link rel="dns-prefetch" href="https://avatars1.githubusercontent.com">
  <link rel="dns-prefetch" href="https://avatars2.githubusercontent.com">
  <link rel="dns-prefetch" href="https://avatars3.githubusercontent.com">
  <link rel="dns-prefetch" href="https://github-cloud.s3.amazonaws.com">
  <link rel="dns-prefetch" href="https://user-images.githubusercontent.com/">



  <link crossorigin="anonymous" media="all" integrity="sha512-O8MvHH7UTZWia0+XOTD76ZNDP3IrRQWNXLwf+F4M4av4ahbxG7JN4doDxxpss+XGpdrF8C72Lg8y0
UhxnA==" rel="stylesheet" href="https://assets-cdn.github.com/assets/frameworks-8e75cb55ad06095e497d44ea5c418a39.css" />
  <link crossorigin="anonymous" media="all" integrity="sha512-DyXl1bArsiH1cJi7yX9k1qCph8YUDg/rYX6RTjpjhY8AoRM7AcgwNhjWefhGbHjUW7LbqTtMkOlWt
lNreQ==" rel="stylesheet" href="https://assets-cdn.github.com/assets/github-d26e79a8226bd7891faf32bc2ccb6073.css" />


  <link crossorigin="anonymous" media="all" integrity="sha512-+G4sIYlb3eQxH1jJoAG/Ed2g3dlNc6jvO89e2RBT0+oVtPJQP4AINvlrwG4w48vGz0HVM7frVoaV1
b6/1A==" rel="stylesheet" href="https://assets-cdn.github.com/assets/site-b4158a9f22ebd9e592779d889c0f9aaf.css" />


  <meta name="viewport" content="width=device-width">

  <title>elsander (Elizabeth Sander) · GitHub</title>
    <meta name="description" content="GitHub is where people build software. More than 27 million people use GitHub to discover, fork, and
ribute to over 80 million projects.">
  <link rel="search" type="application/opensearchdescription+xml" href="/opensearch.xml" title="GitHub">
  <link rel="fluid-icon" href="https://github.com/fluidicon.png" title="GitHub">
  <meta property="fb:app_id" content="1401488693436528">
```

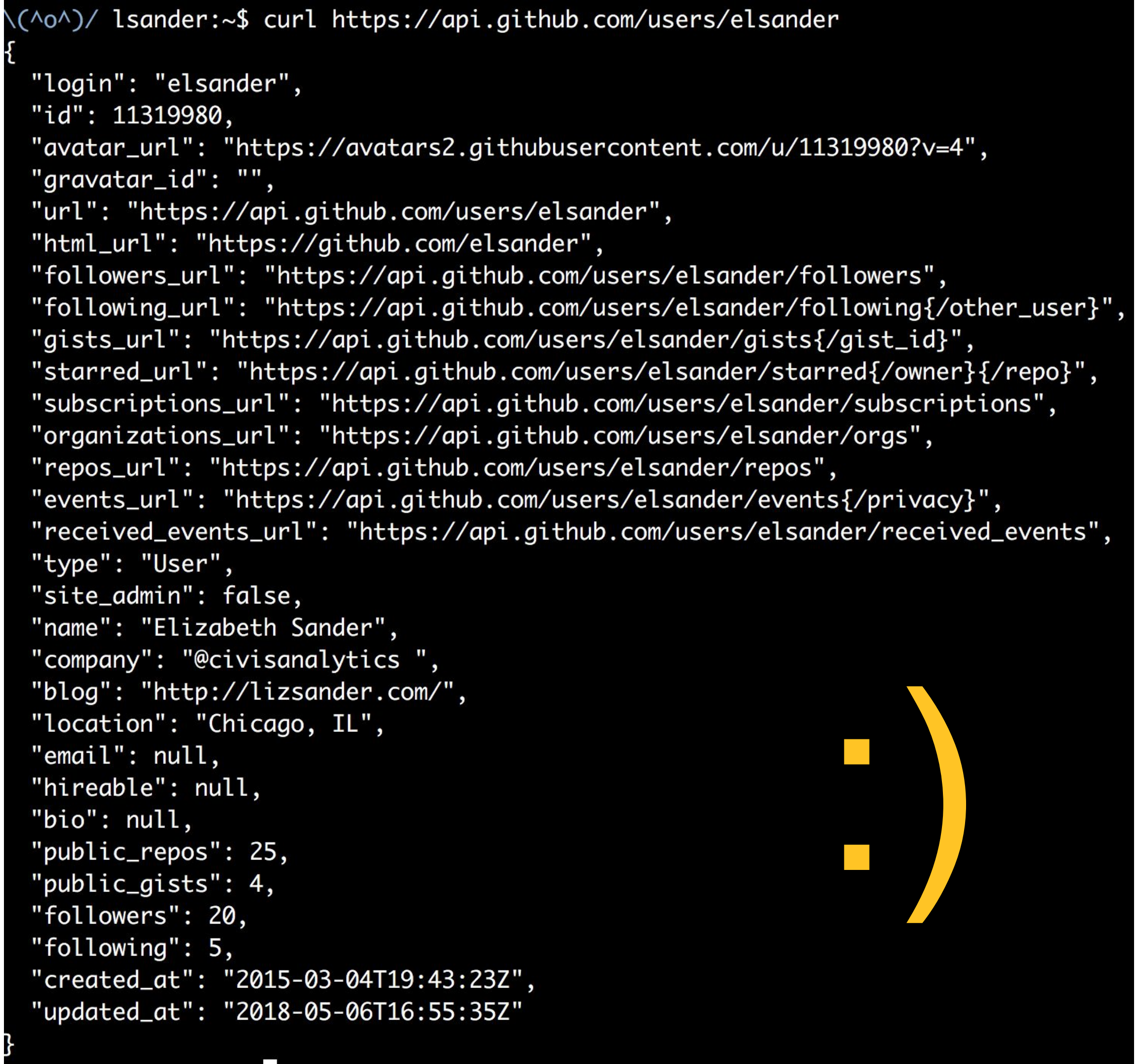


```
\(^o^)/ lsander:~$ head -n 30 github.html


<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8">
  <link rel="dns-prefetch" href="https://assets-cdn.github.com">
  <link rel="dns-prefetch" href="https://avatars0.githubusercontent.com">
  <link rel="dns-prefetch" href="https://avatars1.githubusercontent.com">
  <link rel="dns-prefetch" href="https://avatars2.githubusercontent.com">
  <link rel="dns-prefetch" href="https://avatars3.githubusercontent.com">
  <link rel="dns-prefetch" href="https://github-cloud.s3.amazonaws.com">
  <link rel="dns-prefetch" href="https://user-images.githubusercontent.com/">



  <link crossorigin="anonymous" media="all" integrity="sha512-O8MvHH7UTZWia0+XOTD76ZNDP3IrRQWNXLwf+F4M4av4ahbxG7JN4doDxxpss+XGpdrF8C72Lg8y0
UhxnA==" rel="stylesheet" href="https://assets-cdn.github.com/assets/frameworks-8e75cb55ad06095e497d44ea5c418a39.css" />
  <link crossorigin="anonymous" media="all" integrity="sha512-DyXl1bArsiH1cJi7yX9k1qCph8YUDg/rYX6RTjpjhY8AoRM7AcgwNhjWefhGbHjUW7LbqTtMkOlWt
lNreQ==" rel="stylesheet" href="https://assets-cdn.github.com/assets/github-d26e79a8226bd7891faf32bc2ccb6073.css" />


  <link crossorigin="anonymous" media="all" integrity="sha512-+G4sIYlb3eQxH1jJoAG/Ed2g3dlNc6jvO89e2RBT0+oVtPJQP4AINvlrwG4w48vGz0HVM7frVoaV1
b6/1A==" rel="stylesheet" href="https://assets-cdn.github.com/assets/site-b4158a9f22ebd9e592779d889c0f9aaf.css" />


  <meta name="viewport" content="width=device-width">

  <title>elsander (Elizabeth Sander) · GitHub</title>
    <meta name="description" content="GitHub is where people build software. More than 27 million people use GitHub to discover, fork, and
ribute to over 80 million projects.">
  <link rel="search" type="application/opensearchdescription+xml" href="/opensearch.xml" title="GitHub">
  <link rel="fluid-icon" href="https://github.com/fluidicon.png" title="GitHub">
  <meta property="fb:app_id" content="1401488693436528">
```

:(



```
\(^o^)/ lsander:~$ curl https://api.github.com/users/elsander
{
  "login": "elsander",
  "id": 11319980,
  "avatar_url": "https://avatars2.githubusercontent.com/u/11319980?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/elsander",
  "html_url": "https://github.com/elsander",
  "followers_url": "https://api.github.com/users/elsander/followers",
  "following_url": "https://api.github.com/users/elsander/following{/other_user}",
  "gists_url": "https://api.github.com/users/elsander/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/elsander/starred{/owner}{/repo}",
  "subscriptions_url": "https://api.github.com/users/elsander/subscriptions",
  "organizations_url": "https://api.github.com/users/elsander/orgs",
  "repos_url": "https://api.github.com/users/elsander/repos",
  "events_url": "https://api.github.com/users/elsander/events{/privacy}",
  "received_events_url": "https://api.github.com/users/elsander/received_events",
  "type": "User",
  "site_admin": false,
  "name": "Elizabeth Sander",
  "company": "@civisanalytics ",
  "blog": "http://lizsander.com/",
  "location": "Chicago, IL",
  "email": null,
  "hireable": null,
  "bio": null,
  "public_repos": 25,
  "public_gists": 4,
  "followers": 20,
  "following": 5,
  "created_at": "2015-03-04T19:43:23Z",
  "updated_at": "2018-05-06T16:55:35Z"
}
```

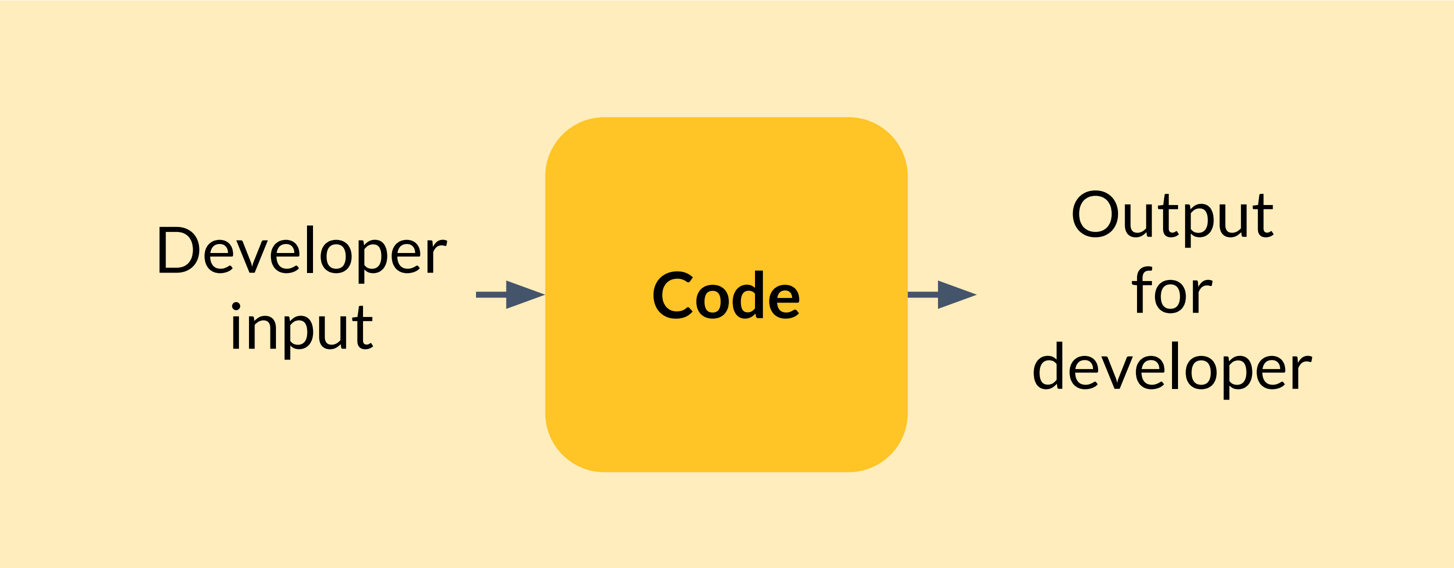


```
\(^o^)/ lsander:~$ curl https://api.github.com/users/elsander
{
  "login": "elsander",
  "id": 11319980,
  "avatar_url": "https://avatars2.githubusercontent.com/u/11319980?v=4",
  "gravatar_id": "",
  "url": "https://api.github.com/users/elsander",
  "html_url": "https://github.com/elsander",
  "followers_url": "https://api.github.com/users/elsander/followers",
  "following_url": "https://api.github.com/users/elsander/following{/other_user}",
  "gists_url": "https://api.github.com/users/elsander/gists{/gist_id}",
  "starred_url": "https://api.github.com/users/elsander/starred{/owner}{/repo}",
  "subscriptions_url": "https://api.github.com/users/elsander/subscriptions",
  "organizations_url": "https://api.github.com/users/elsander/orgs",
  "repos_url": "https://api.github.com/users/elsander/repos",
  "events_url": "https://api.github.com/users/elsander/events{/privacy}",
  "received_events_url": "https://api.github.com/users/elsander/received_events",
  "type": "User",
  "site_admin": false,
  "name": "Elizabeth Sander",
  "company": "@civisanalytics ",
  "blog": "http://lizsander.com/",
  "location": "Chicago, IL",
  "email": null,
  "hireable": null,
  "bio": null,
  "public_repos": 25,
  "public_gists": 4,
  "followers": 20,
  "following": 5,
  "created_at": "2015-03-04T19:43:23Z",
  "updated_at": "2018-05-06T16:55:35Z"
}
```

:)

APIs are for software too!

- Think of an API as the “developer interface” (as opposed to the user interface)



What makes a good API?

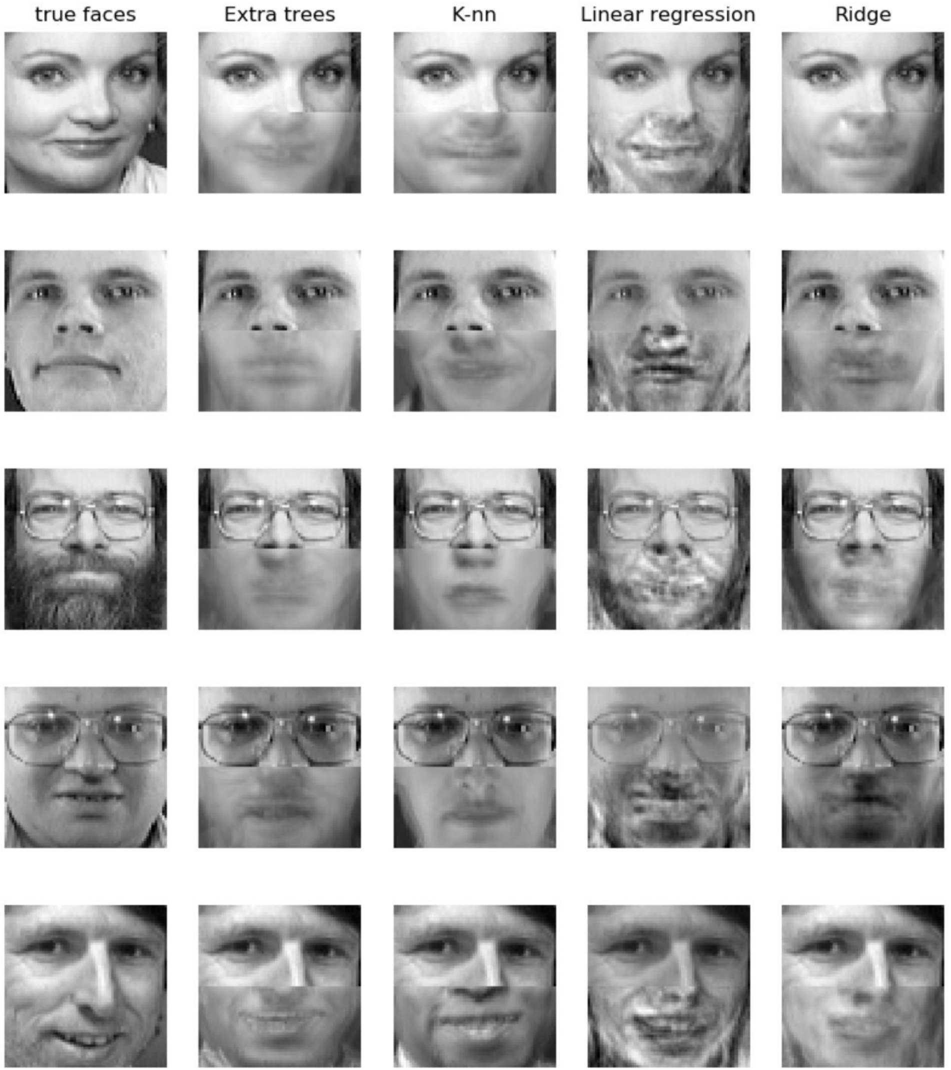
- Stable
- Integrates with existing tools
- Intuitive
- Flexible/extendable

Software libraries have APIs. It’s worth some upfront time to make them useful.

Software libraries have APIs. It’s worth some upfront time to make them useful.

Let’s look at a library that does it well!

Scikit-Learn


true faces
Extra trees
K-nn
Linear regression
Ridge

What makes a good API?

- Stable
- Integrates with existing tools
- Intuitive
- Flexible/extendable

What makes a good API?

- Stable ✓
- Integrates with existing tools
- Intuitive
- Flexible/extendable

What makes a good API?

- Stable ✓
- Integrates with existing tools ✓
- Intuitive
- Flexible/extendable

What makes a good API?

- Stable ✓
- Integrates with existing tools ✓
- **Intuitive**
- **Flexible/extendable**

How do I write a class for logistic regression?

How do I write a class for logistic regression?
… what about a random forest?

How do I write a function/class for logistic regression?

… what about a random forest?

… and a neural network?

~~How do I write a class for logistic regression?~~

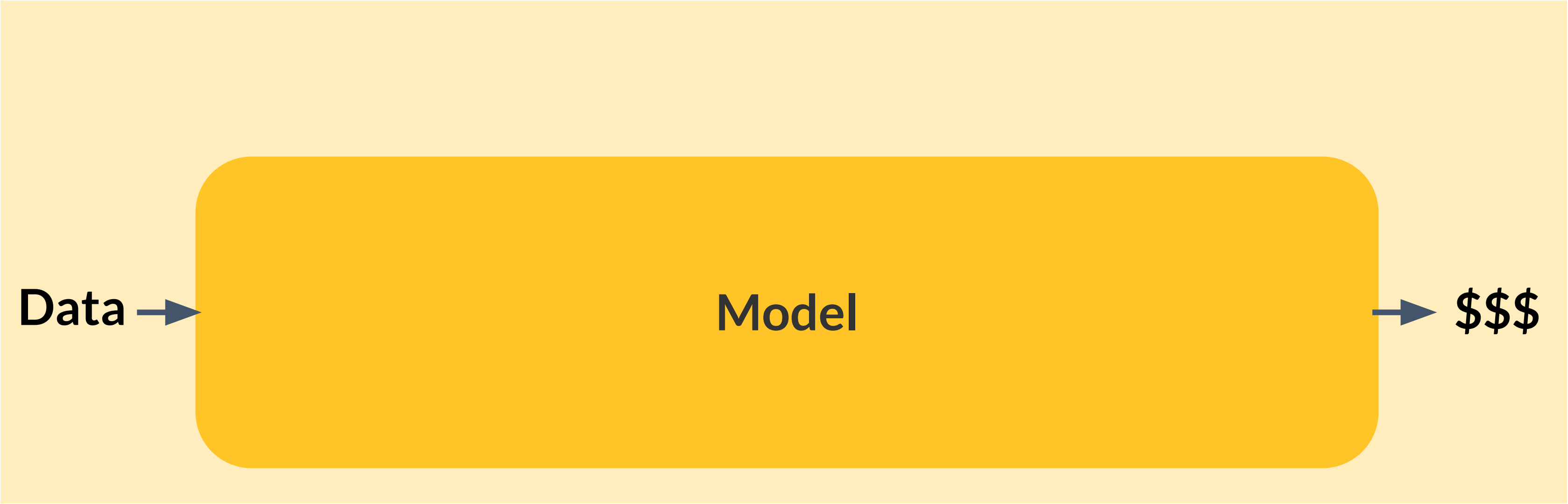
~~… what about a random forest?~~

~~… and a neural network?~~

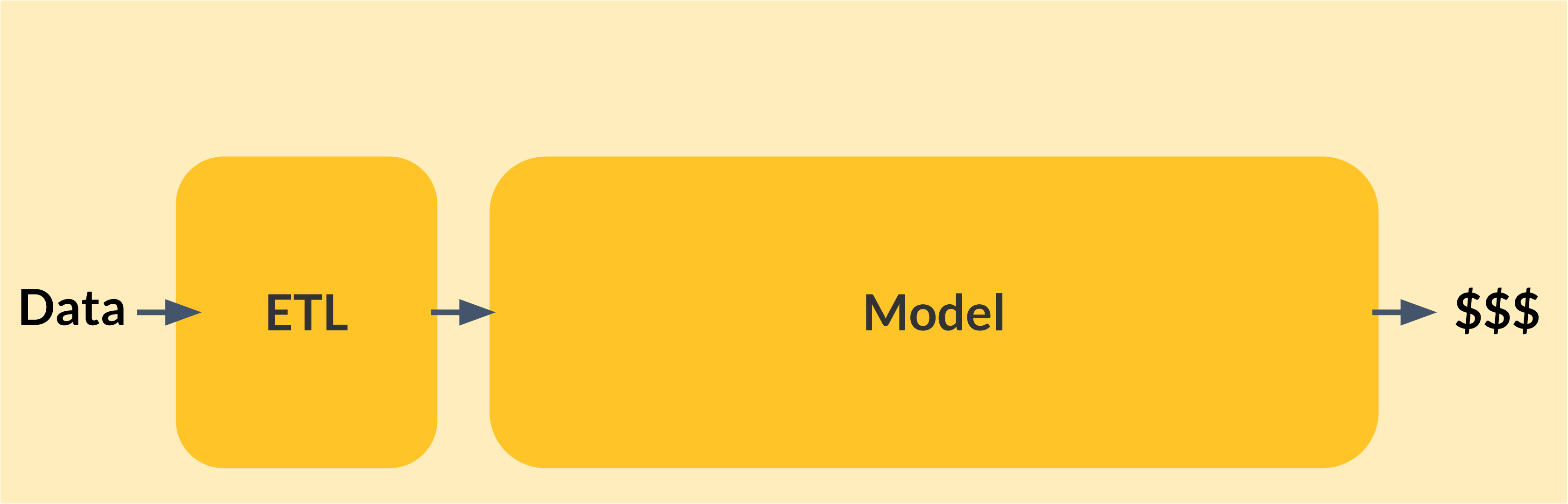
How do I create a general framework for modeling?

What is a model?

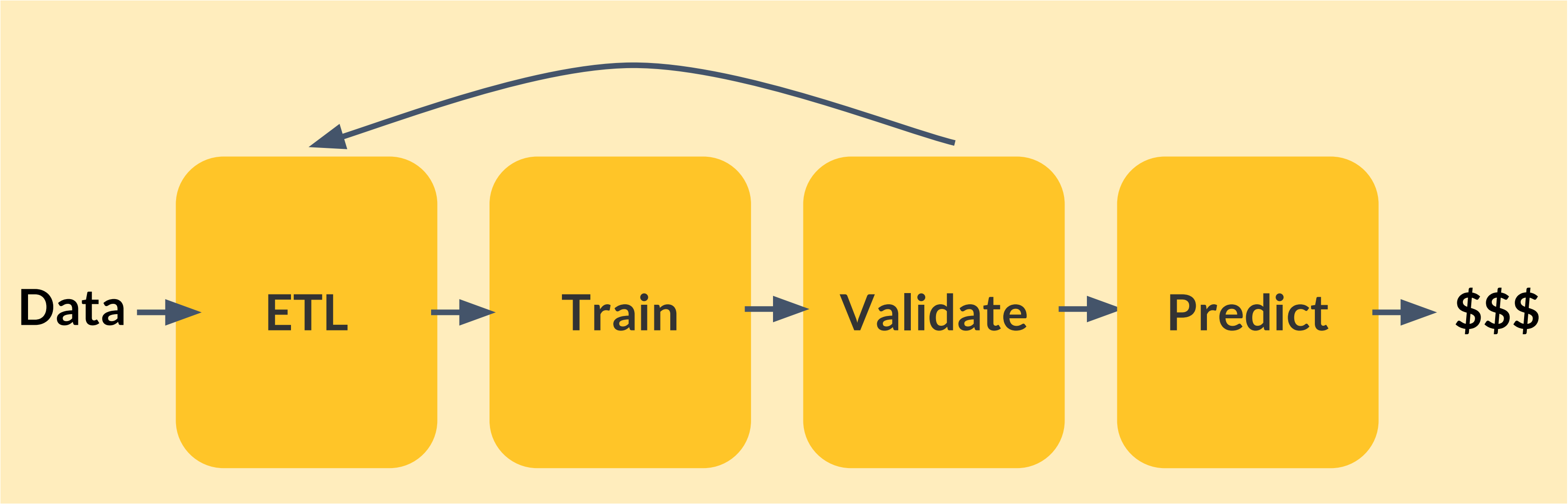
What is a model?



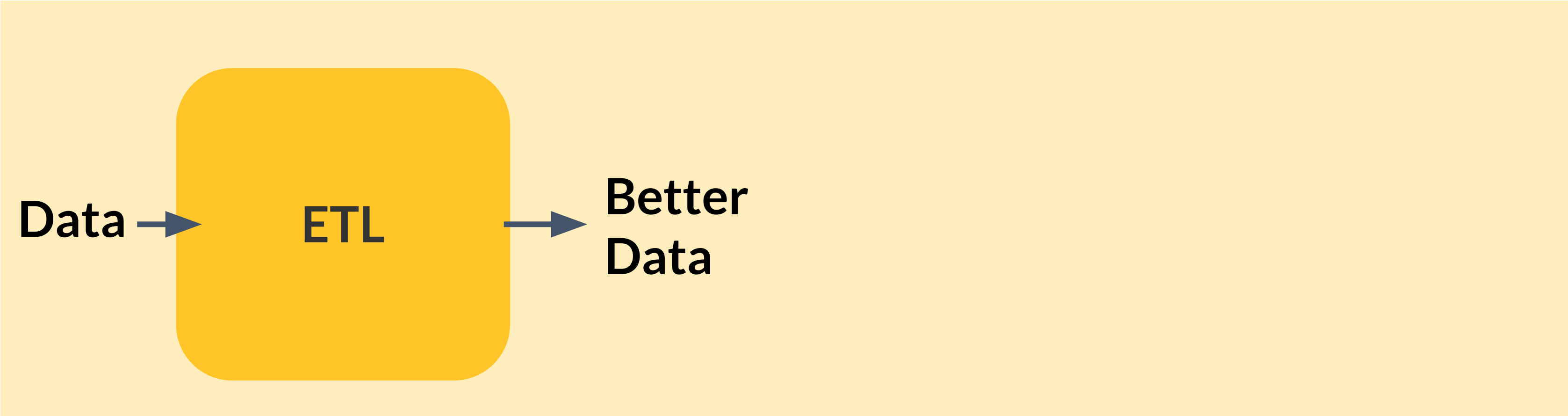
What is a model?



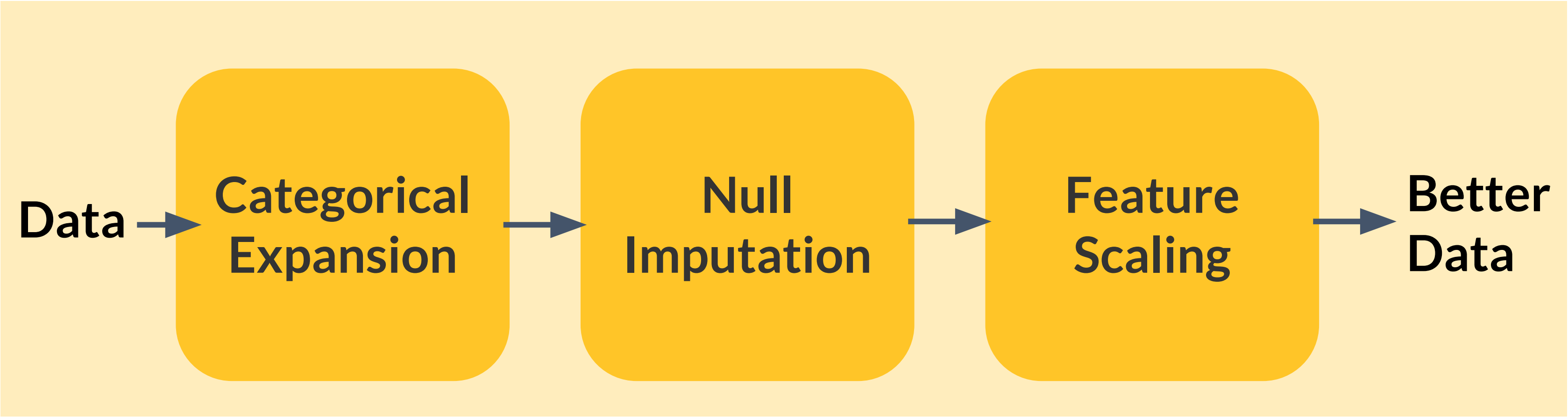
What is a model?



What is ETL?

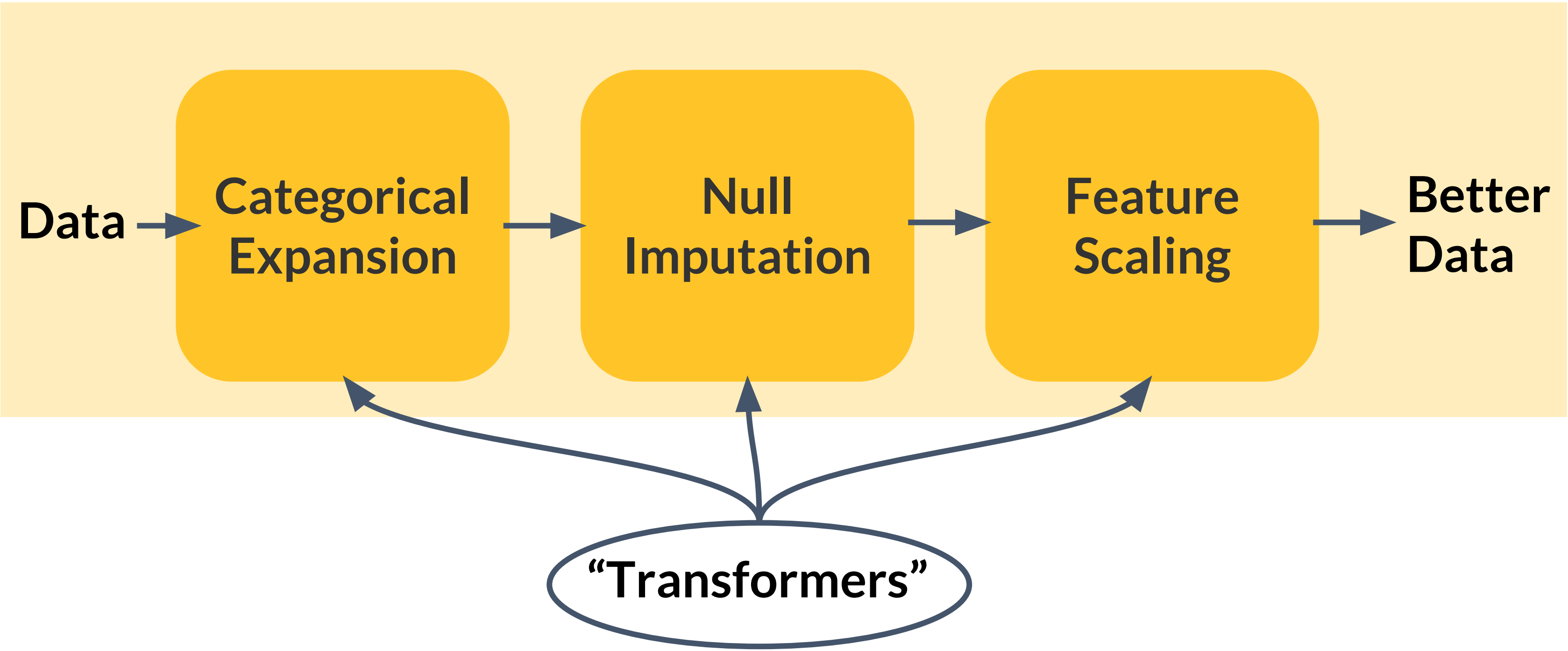


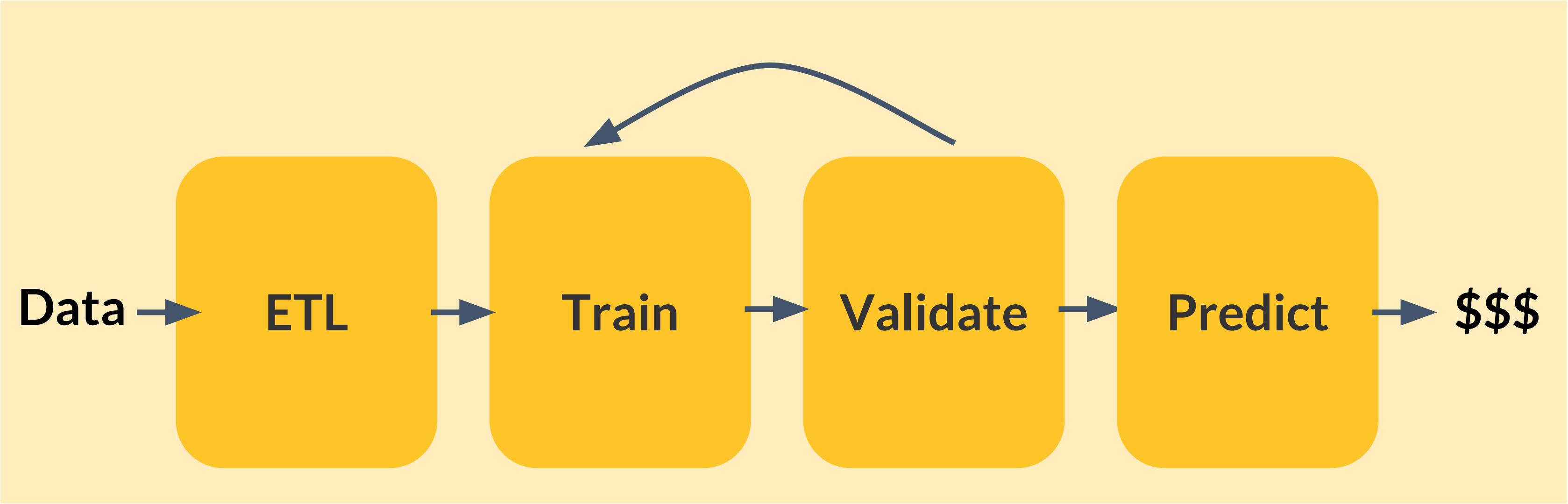
What is ETL?

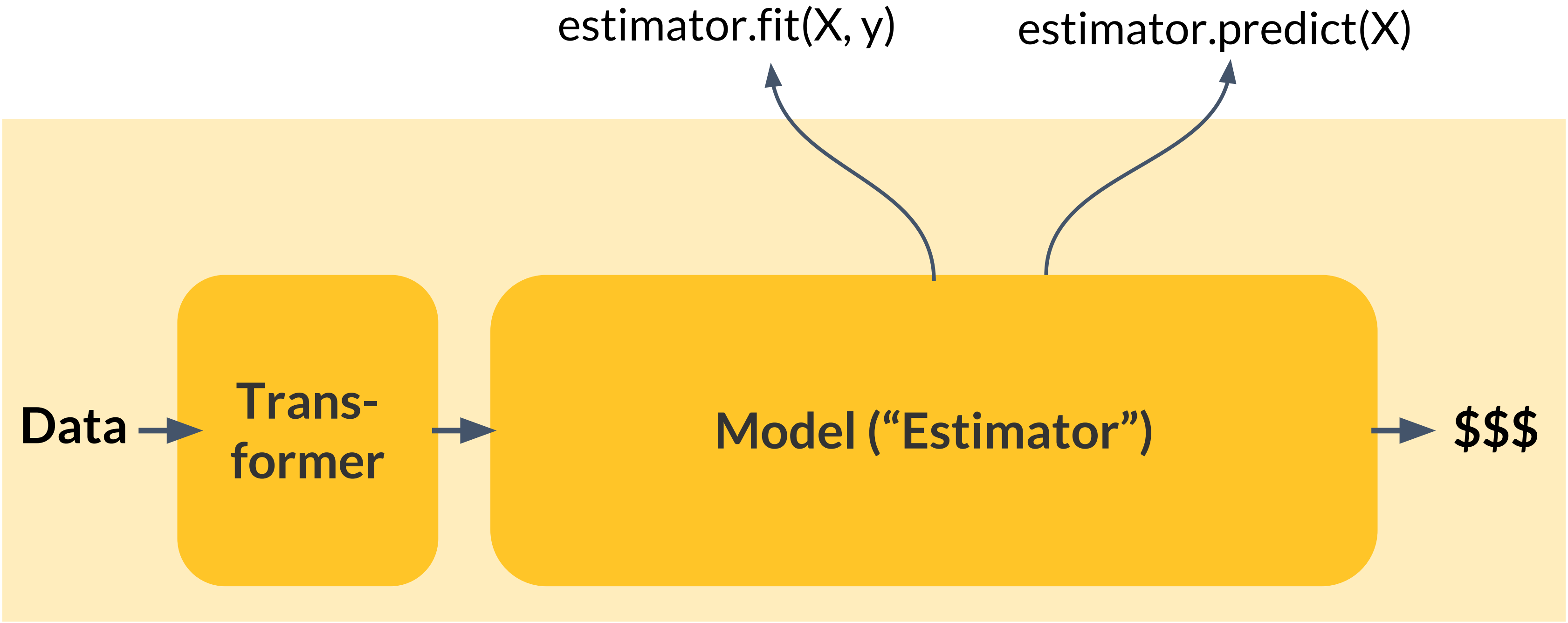


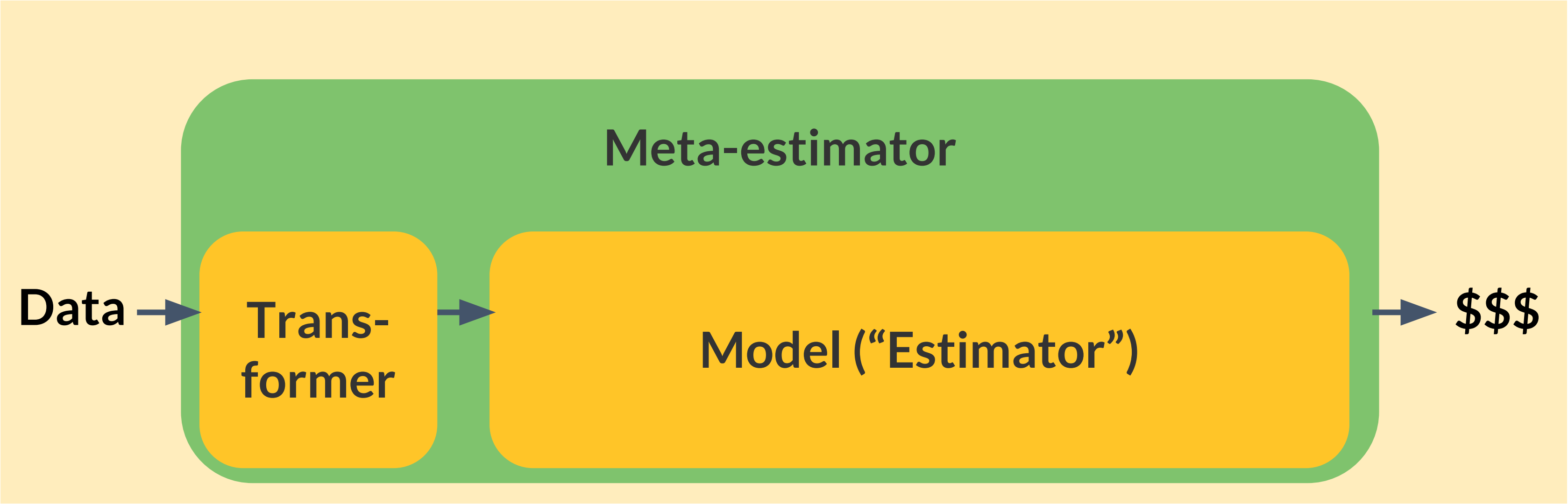
Data → Categorical Expansion → Null Imputation → Feature Scaling → Better Data

What is ETL?

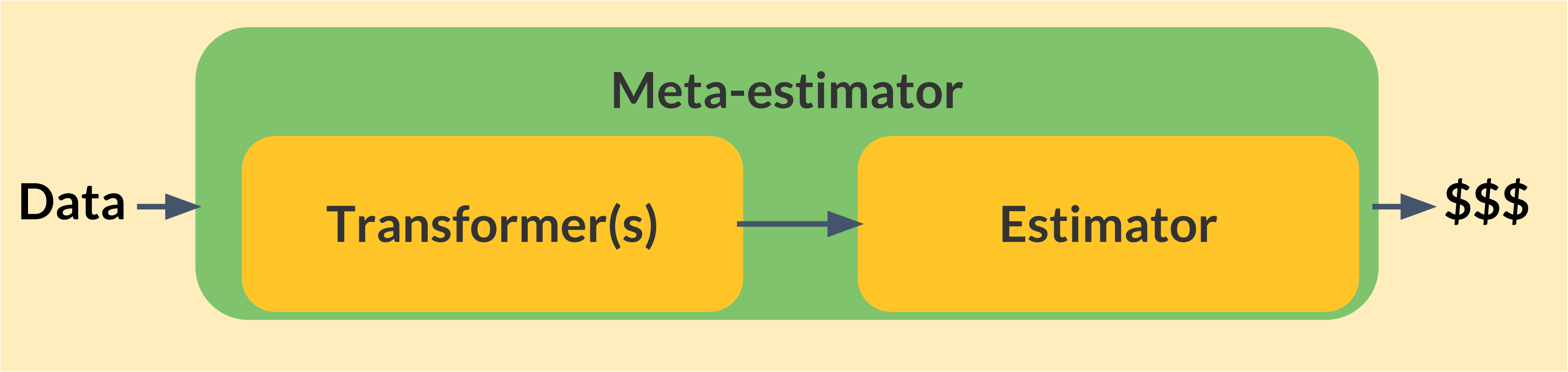

Data
Categorical Expansion
Null Imputation
Feature Scaling
Better Data
“Transformers”

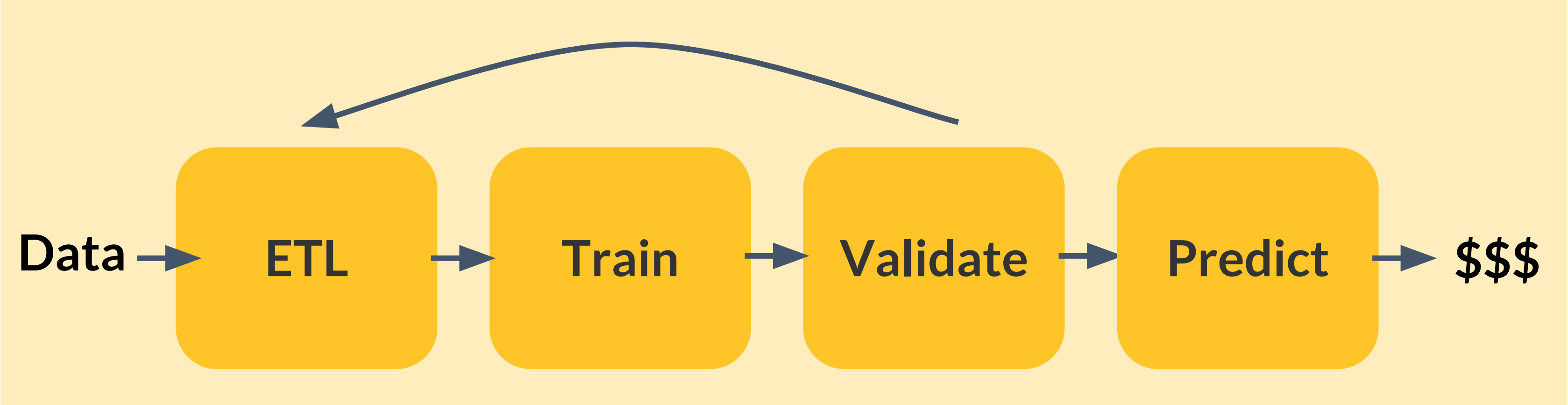

Data
ETL
Train
Validate
Predict
$$$


estimator.fit(X, y)
estimator.predict(X)
Data
Trans-
former
Model ("Estimator")
$$$

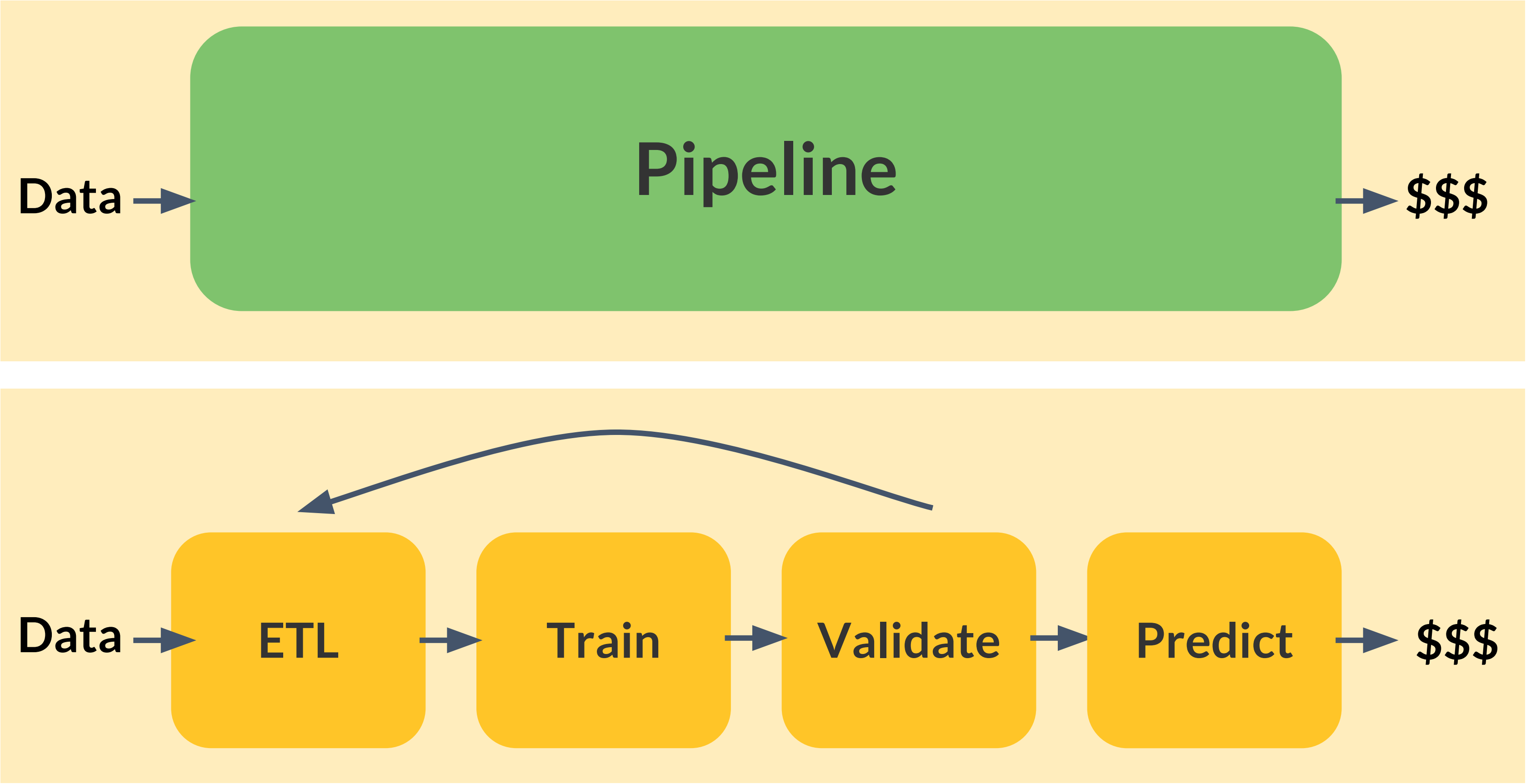

Meta-estimator
Data
Trans-
former
Model ("Estimator")
$$$

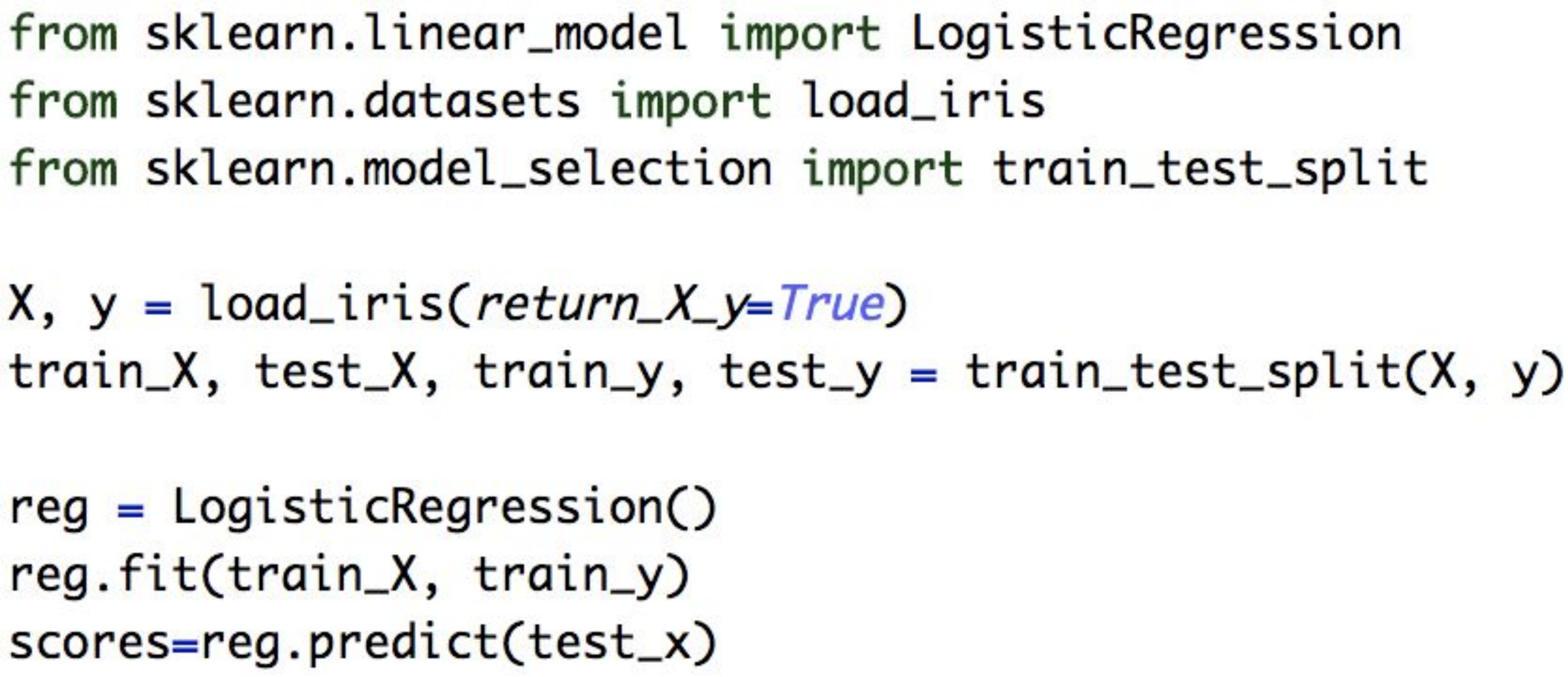
The Scikit-learn API



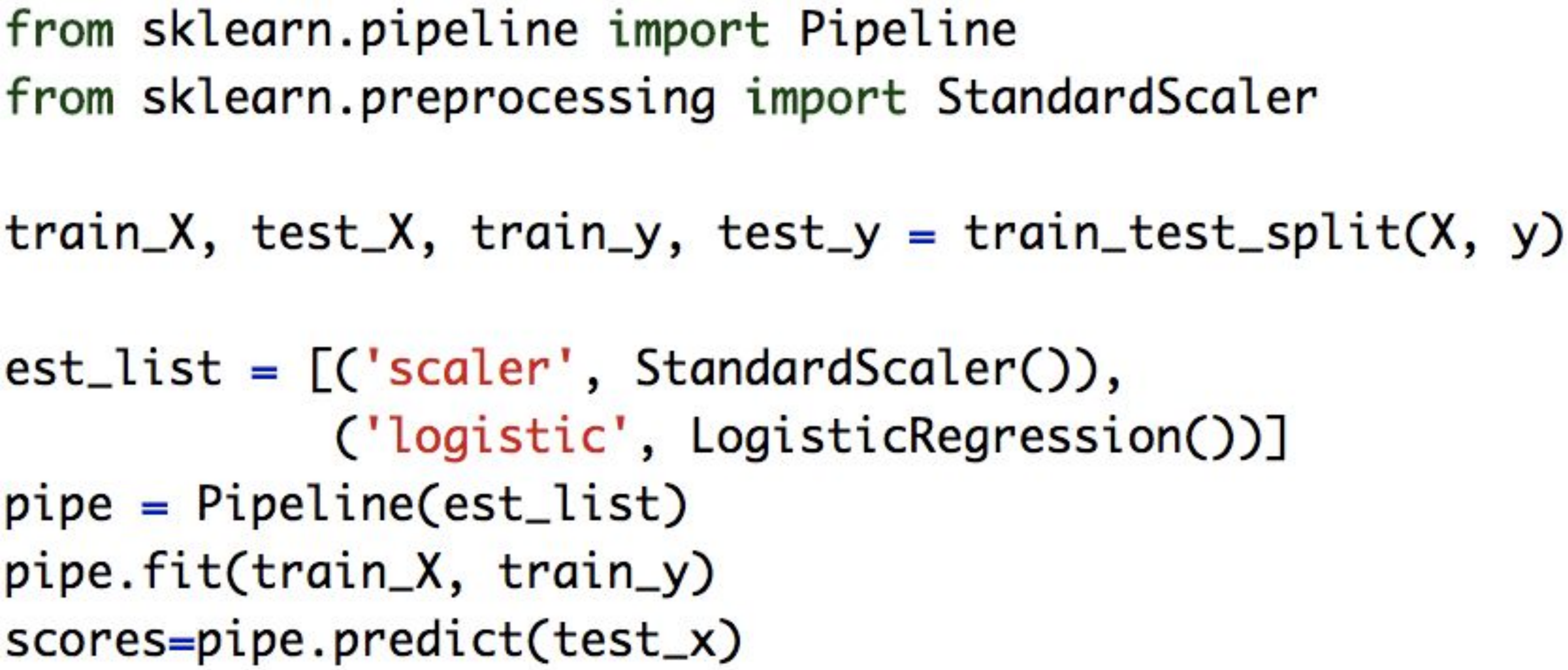


The Scikit-learn API

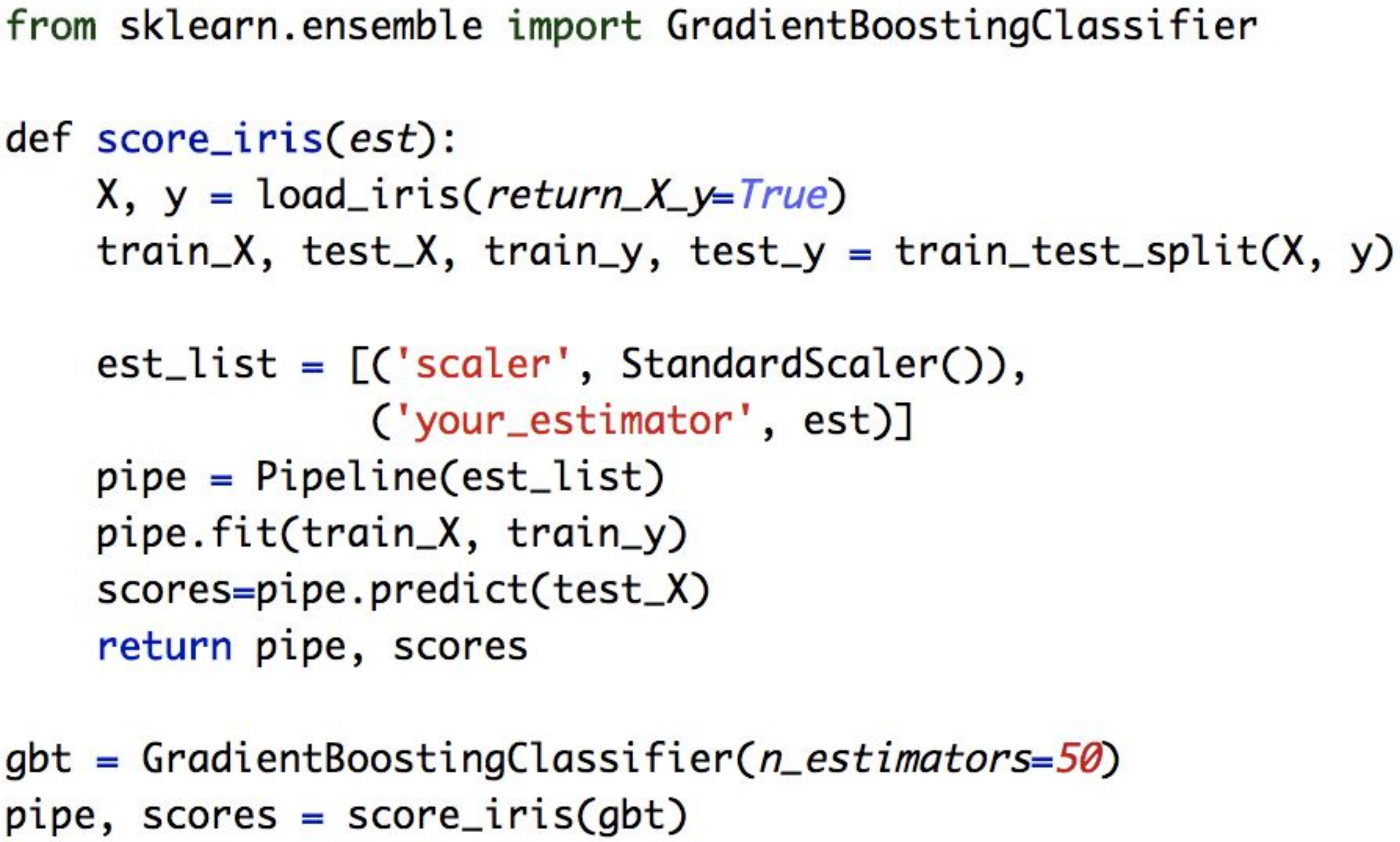




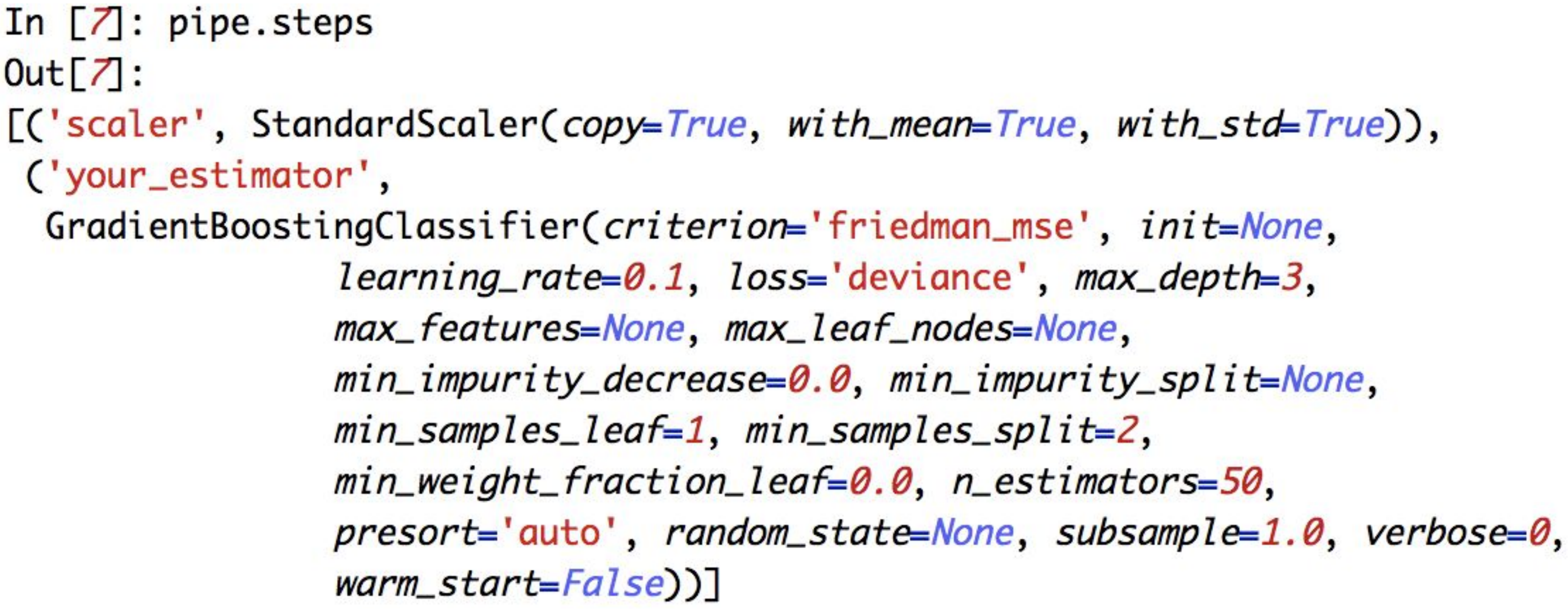
```
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

X, y = load_iris(return_X_y=True)
train_X, test_X, train_y, test_y = train_test_split(X, y)

reg = LogisticRegression()
reg.fit(train_X, train_y)
scores=reg.predict(test_x)
```



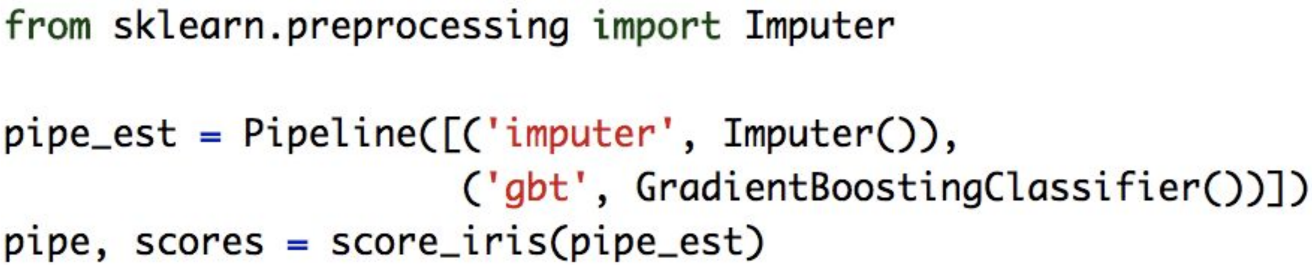
```python
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler

train_X, test_X, train_y, test_y = train_test_split(X, y)

est_list = [('scaler', StandardScaler()),
            ('logistic', LogisticRegression())]
pipe = Pipeline(est_list)
pipe.fit(train_X, train_y)
scores=pipe.predict(test_x)
```



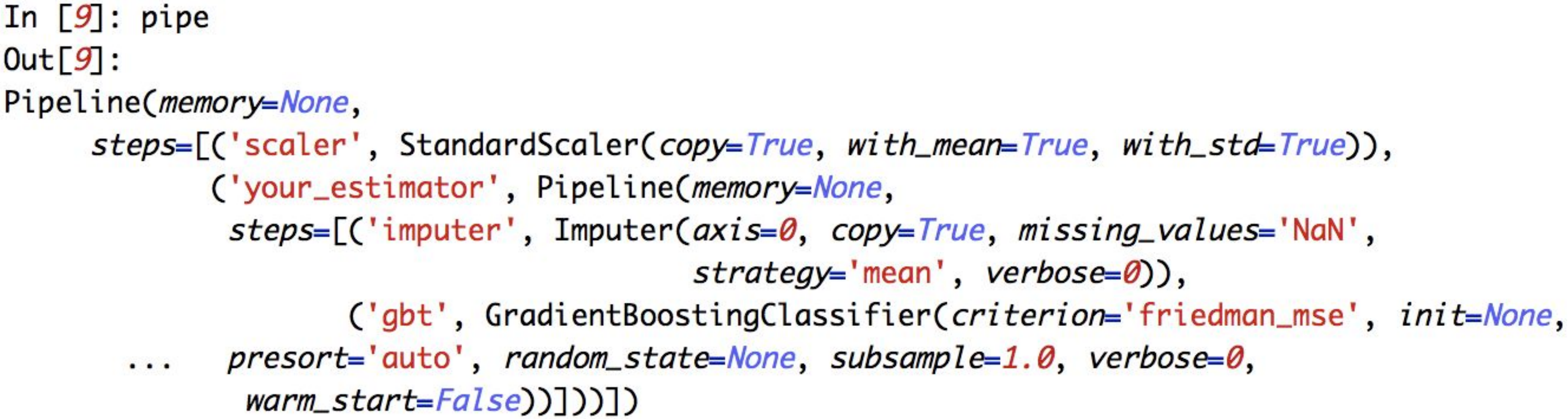
```python
from sklearn.ensemble import GradientBoostingClassifier

def score_iris(est):
    X, y = load_iris(return_X_y=True)
    train_X, test_X, train_y, test_y = train_test_split(X, y)

    est_list = [('scaler', StandardScaler()),
                ('your_estimator', est)]
    pipe = Pipeline(est_list)
    pipe.fit(train_X, train_y)
    scores=pipe.predict(test_X)
    return pipe, scores

gbt = GradientBoostingClassifier(n_estimators=50)
pipe, scores = score_iris(gbt)
```



```
In [7]: pipe.steps
Out[7]:
[('scaler', StandardScaler(copy=True, with_mean=True, with_std=True)),
 ('your_estimator',
  GradientBoostingClassifier(criterion='friedman_mse', init=None,
              learning_rate=0.1, loss='deviance', max_depth=3,
              max_features=None, max_leaf_nodes=None,
              min_impurity_decrease=0.0, min_impurity_split=None,
              min_samples_leaf=1, min_samples_split=2,
              min_weight_fraction_leaf=0.0, n_estimators=50,
              presort='auto', random_state=None, subsample=1.0, verbose=0,
              warm_start=False))]
```



```python
from sklearn.preprocessing import Imputer

pipe_est = Pipeline([('imputer', Imputer()),
                     ('gbt', GradientBoostingClassifier())])
pipe, scores = score_iris(pipe_est)
```



```
In [9]: pipe
Out[9]:
Pipeline(memory=None,
     steps=[('scaler', StandardScaler(copy=True, with_mean=True, with_std=True)),
          ('your_estimator', Pipeline(memory=None,
           steps=[('imputer', Imputer(axis=0, copy=True, missing_values='NaN',
                                  strategy='mean', verbose=0)),
                ('gbt', GradientBoostingClassifier(criterion='friedman_mse', init=None,
      ...  presort='auto', random_state=None, subsample=1.0, verbose=0,
            warm_start=False))]))])
```

The right abstraction makes a library easier to use and reason about.

What makes a good API?

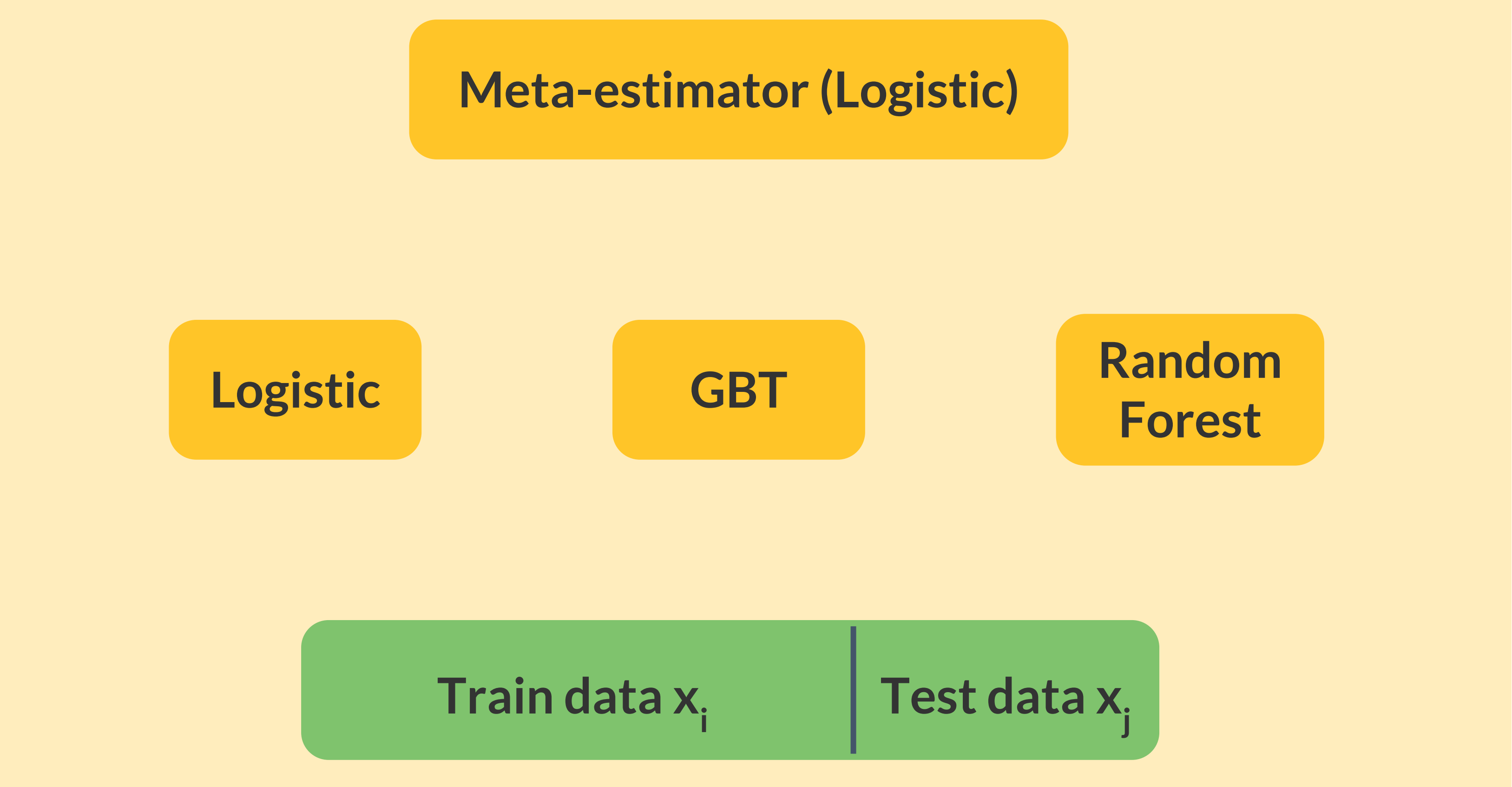
- Stable ✓
- Integrates with existing tools ✓
- Intuitive ✓
- **Flexible/extendable**

Scikit-learn extensions

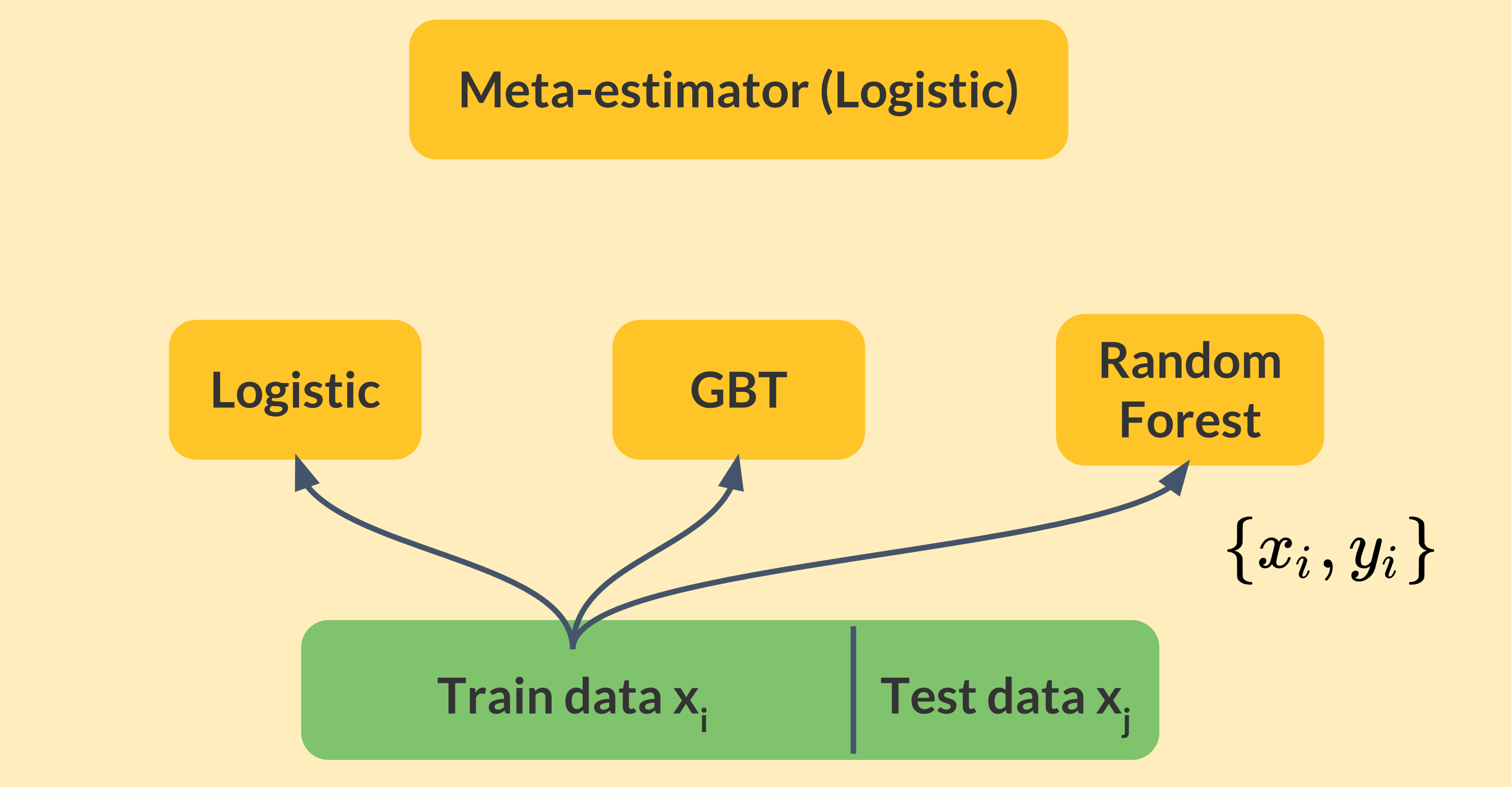
- xgboost, keras, lightning
- Civis-maintained
  - python-glmnet (R wrapper)
  - civisml-extensions
  - muffnn
- Scikit-learn maintains a list of many others

Stacking

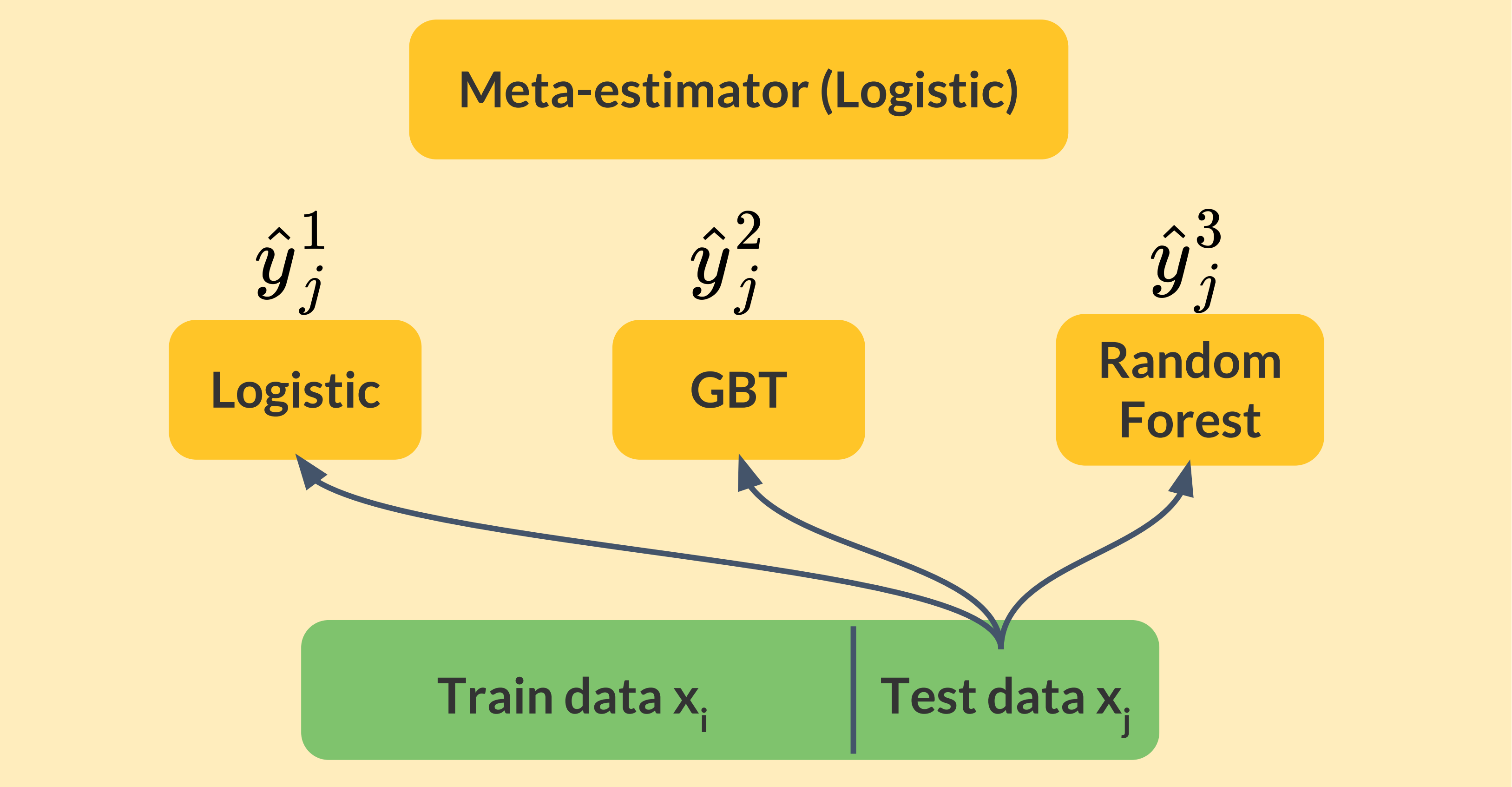
Stacking


Meta-estimator (Logistic)
Logistic
GBT
Random Forest
Train data x_i
Test data x_j

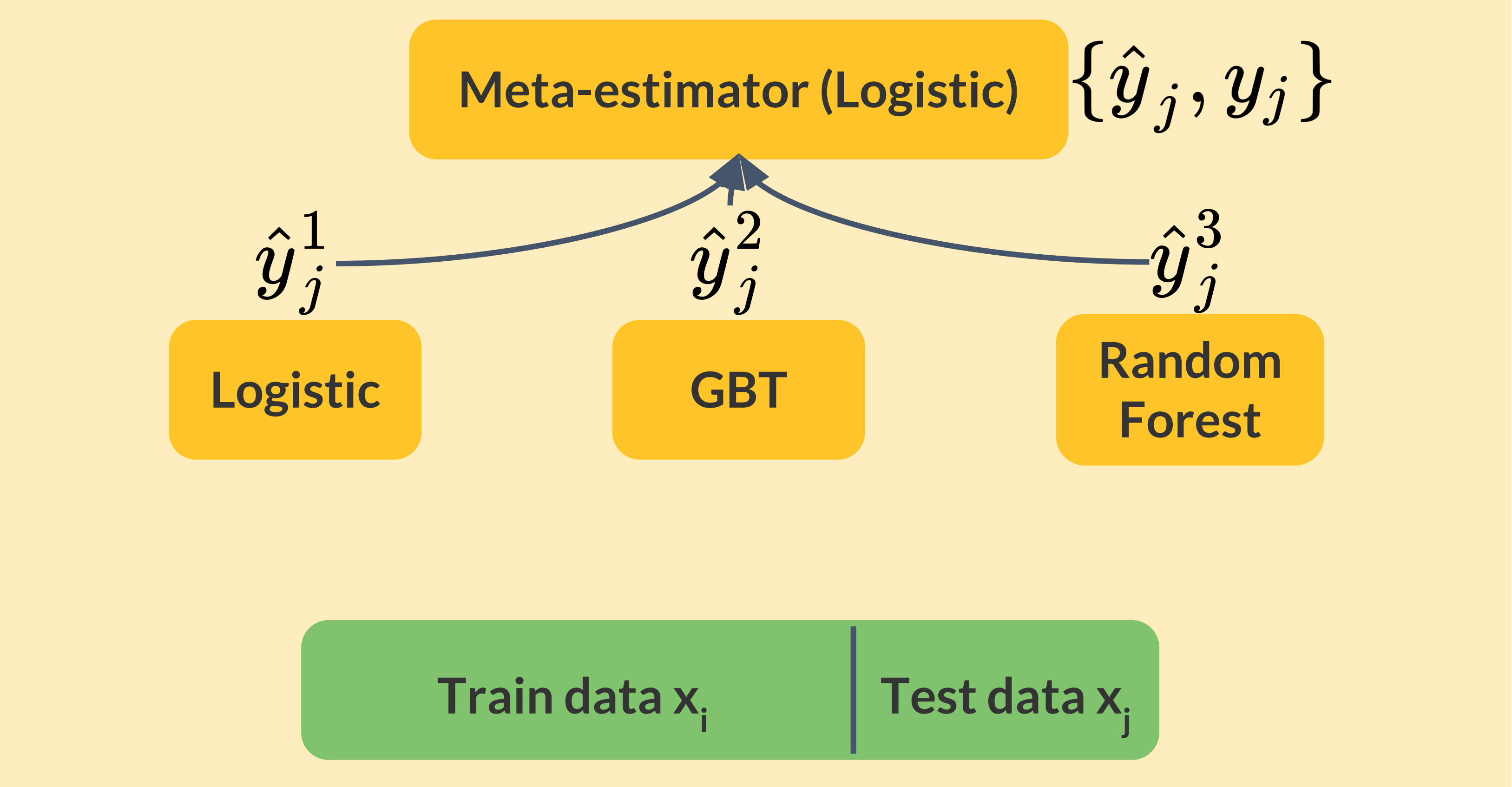
Train base estimators using {x_i, y_i}



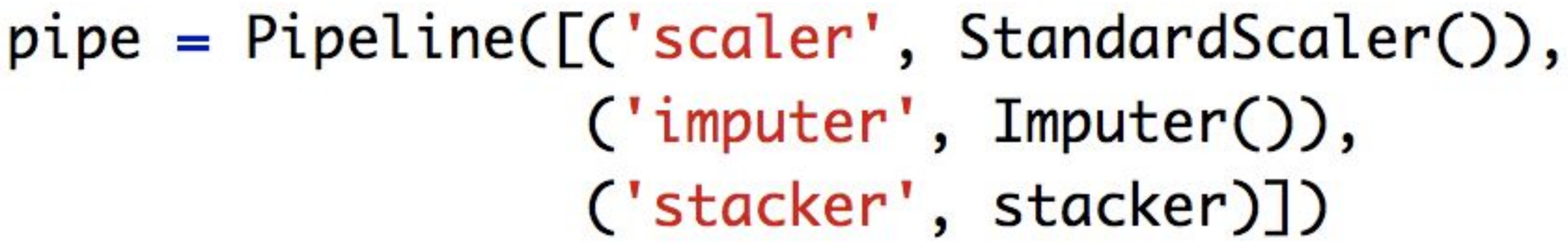
Predict base estimators on $\{x_j\}$



Use predictions as features to train meta-estimator

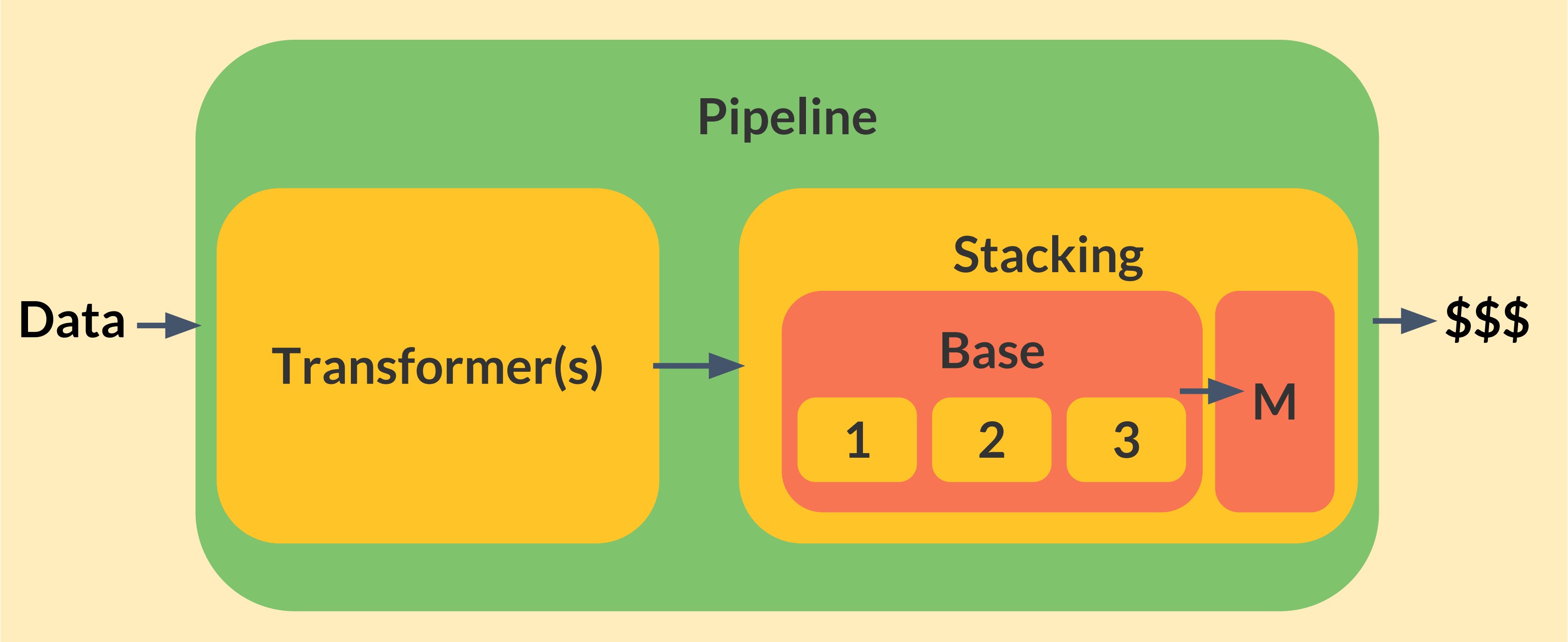


```python
from civismlext.stacking import StackedClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier
from sklearn.datasets import load_iris

iris_x, iris_y = load_iris(return_X_y=True)
est_list = [('logistic', LogisticRegression()),
            ('rf', RandomForestClassifier()),
            ('gbt', GradientBoostingClassifier()),
            ('meta', LogisticRegression())]

stacker = StackedClassifier(est_list)
stacker.fit(iris_x, iris_y)
scores = stacker.predict(iris_x)
```

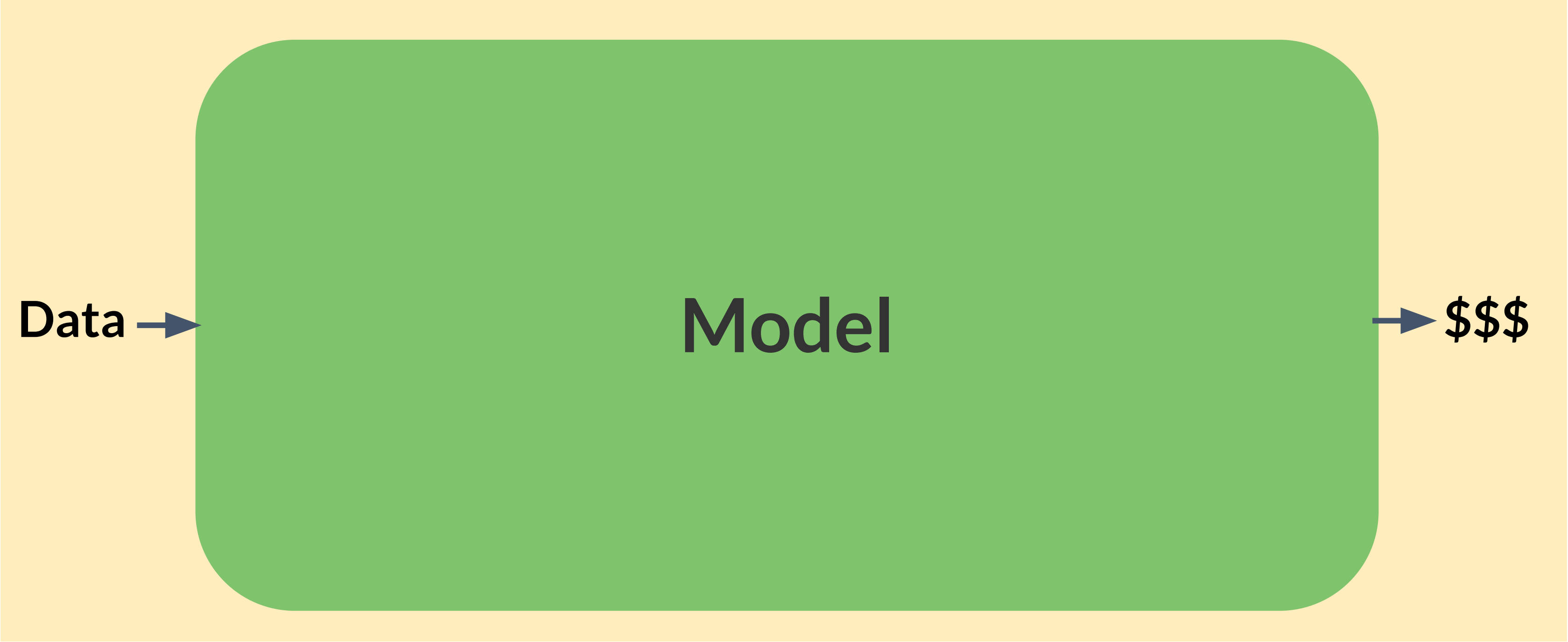


```
pipe = Pipeline([('scaler', StandardScaler()),
                 ('imputer', Imputer()),
                 ('stacker', stacker)])
```

Stacking



Stacking



A robust API can give your library life beyond your own ideas.

Conclusion

- Make your API clear and consistent
- Find an abstraction that mirrors your mental model
- Think about developers as *users*

Resources

https://github.com/civisanalytics/civisml-extensions

http://scikit-learn.org/stable/documentation.html

Questions?

THANK YOU!

Liz Sander
lsander@civisanalytics.com