

1. Предполагается, что у студентов уже установлен Python. Версия и среда для сдачи заданий значения не имеют.

Если у Вас не установлен Python, можете по своему желанию установить интерпретатор+необходимые пакеты или воспользоваться дистрибутивом [Anaconda](#).

Для установки scikit-learn воспользуйтесь менеджером пакетов pip или conda. Conda поставляется вместе с дистрибутивом Anaconda.

Команды для установки scikit-learn: `pip install scikit-learn/conda install scikit-learn`.

2. Демонстрационные задания созданы с помощью интерактивной среды [Jupyter Notebook](#).

3. Библиотека scikit-learn имеет [прекрасную документацию](#) с примерами, пояснениями относительно алгоритмов и их параметров.

4. Кратко простой случай работы с данными можно описать следующим образом:

Загружаем/создаём выборку	<code>from sklearn.datasets import make_regression</code> <code>X,y=make_regression(n_samples=100, ...)</code>
	<code>from sklearn.datasets import load_boston</code> <code>boston = load_boston()</code>
	<code>pandas.read_csv(...)</code>
Предобработка данных	<code>pandas.DataFrame.fillna(...)</code>
	<code>scaler=sklearn.preprocessing.StandardScaler()</code>
Разделение выборки на обучающую и контрольную	<code>sklearn.cross_validation.train_test_split</code> <code>X_train, X_test, y_train, y_test =</code> <code>train_test_split(...)</code>
Обучение модели	<code>from sklearn.linear_model import LinearRegression</code> <code>regressor=LinearRegression()</code> <code>regressor.fit(X_train,y_train)</code>
Проверка качества	<code>prediction=regressor.predict(X_test)</code> <code>from sklearn.metrics import r2_score</code> <code>print r2_score(y_test, prediction)</code>

Также может понадобиться кросс-валидация

Оценка качества по CV	<code>scores = cross_val_score(regressor, X, y, cv=5)</code>
Подбор параметров	<code>sklearn.model_selection.GridSearchCV</code>
	<code>sklearn.model_selection.LassoCV</code>
	<code>sklearn.model_selection.RidgeCV</code>
Стратегии кросс-валидации	<code>sklearn.cross_validation.KFold</code>
	<code>sklearn.cross_validation.LeaveOneOut</code>
	<code>sklearn.cross_validation.ShuffleSplit</code>

Подробнее в ноутбуках.

5. Для моделей (estimator object) существуют методы:

- **estimator.fit()**
- **estimator.transform()** [или **estimator.fit_transform()**] - для моделей из preprocessing, feature_extraction, feature_selection
- **estimator.predict()** [или **estimator.predict_proba()**] - предсказание вероятностей
- для моделей классификации и регрессии

6. Работа с категориальными признаками и спрямляющими пространствами (создании полиномиальных признаков, логарифмических и других преобразованиях) рассматривается в ноутбуках.