Введение в Машинное обучение

Приложения на основе машинного обучения

Machine Learning Applications

Business/ML Problem	Description
Ranking	Помощь пользователям в поиске наиболее релевантной вещи
Recommendation	Предоставление пользователям то, что они могут быть наиболее заинтересованы в
Classification	Выяснение того, что что-то такое
Regression	Прогнозирование численного значения
Clustering	Сгруппировав похожие объекты вместе
Anomaly Detection	Поиск необычных вещей

Example

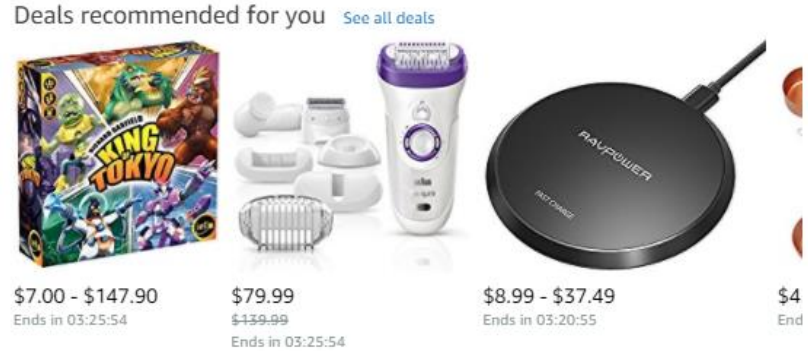
Ranking algorithm within Amazon Search

Machine Learning Applications

Business/ML Problem	Description
Ranking	Помощь пользователям в поиске наиболее релевантной вещи
Recommendation	Предоставление пользователям того, в чем они могут быть наиболее заинтересованы
Classification	Выяснение того, что что-то такое
Regression	Прогнозирование численного значения вещи
Clustering	Сгруппировав похожие объекты вместе
Anomaly Detection	Поиск необычных вещей

Example

Recommendations across the website



Amazon's Choice

Machine Learning Applications

Business/ML Problem	Description
Ranking	Помощь пользователям в поиске наиболее релевантной вещи
Recommendation	Предоставление пользователям того, в чем они могут быть наиболее заинтересованы
Classification	Выяснение того, что что-то такое
Regression	Прогнозирование численного значения
Clustering	Сгруппировав похожие объекты вместе
Anomaly Detection	Поиск необычных вещей

Example

Product classification for our catalog

High-Low Dress

Straight Dress

Striped Skirt

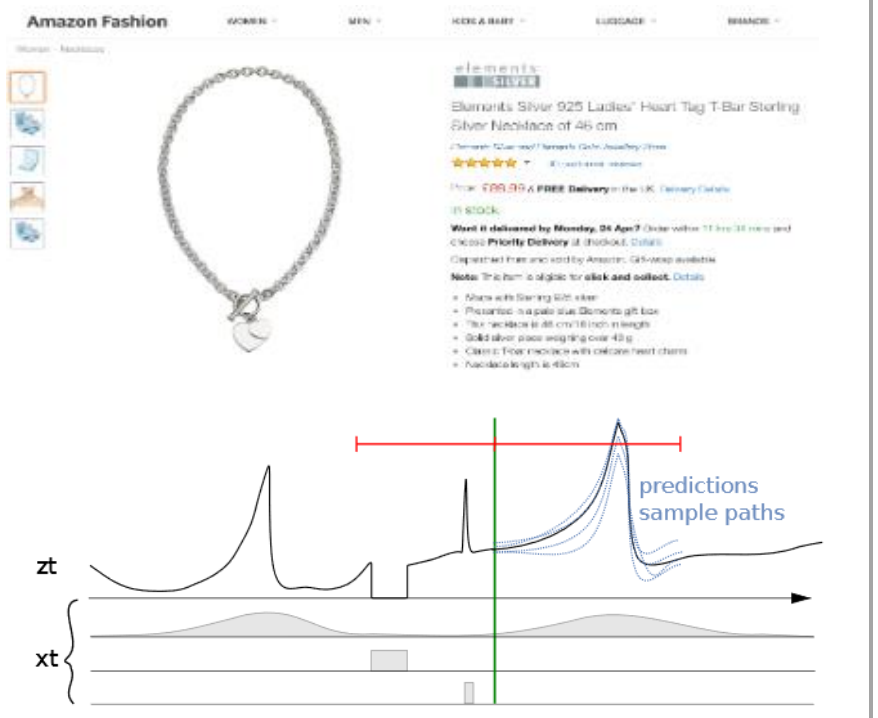
Graphic Shirt

Machine Learning Applications

Business/ML Problem	Description
Ranking	Помощь пользователям в поиске наиболее релевантной вещи
Recommendation	Предоставление пользователям того, в чем они могут быть наиболее заинтересованы
Classification	Выяснение того, что что-то такое
Regression	Прогнозирование численного значения
Clustering	Сгруппировав похожие объекты вместе
Anomaly Detection	Поиск необычных вещей

Example

Predicting sales for specific ASINs



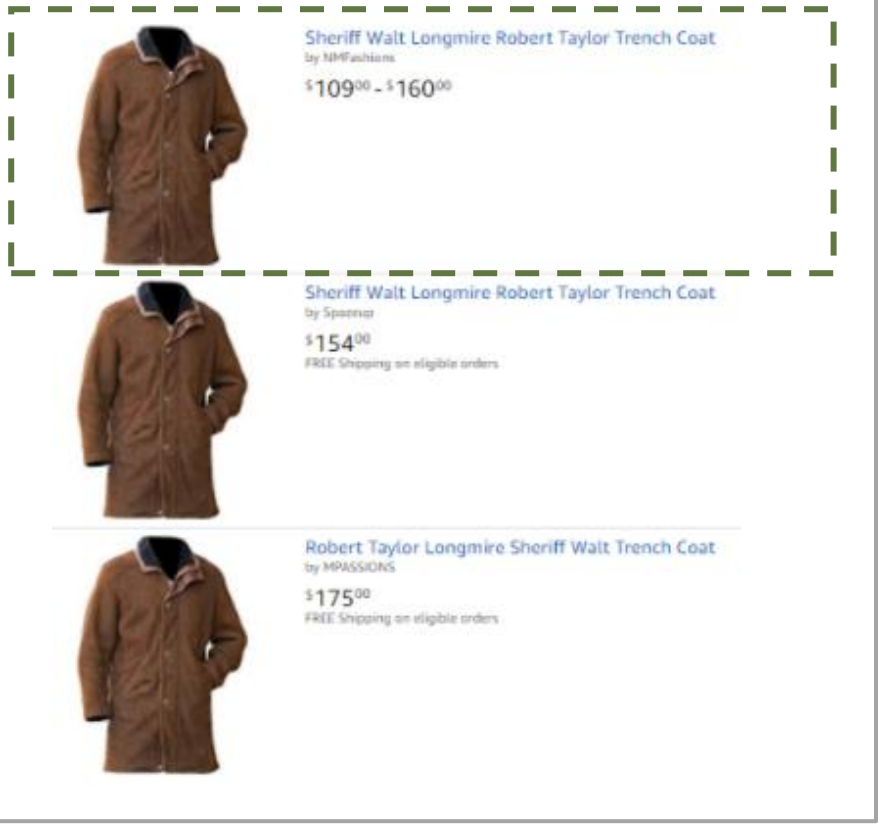
Seasonality | Out of stock | Promotions

Machine Learning Applications

Business/ML Problem	Description
Ranking	Помощь пользователям в поиске наиболее релевантной вещи
Recommendation	Предоставление пользователям того, в чем они могут быть наиболее заинтересованы
Classification	Выяснение того, что что-то такое
Regression	Прогнозирование численного значения
Clustering	Сгруппировав похожие объекты вместе
Anomaly Detection	Поиск необычных вещей

Example

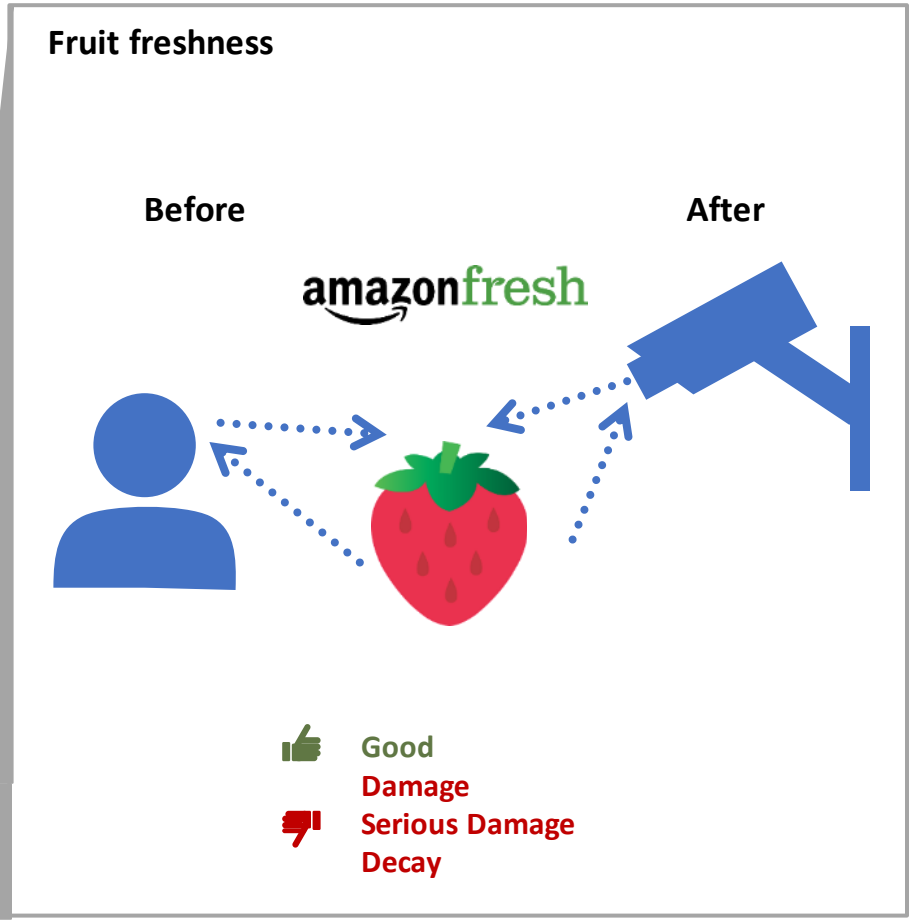
Close-matching for near-duplicates



Machine Learning Applications

Business/ML Problem	Description
Ranking	Помощь пользователям в поиске наиболее релевантной вещи
Recommendation	Предоставление пользователям то, что они могут быть наиболее заинтересованы в
Classification	Выяснение того, что что-то такое
Regression	Прогнозирование численного значения вещи
Clustering	Сгруппировав похожие объекты вместе
Anomaly Detection	Поиск необычных вещей

Example

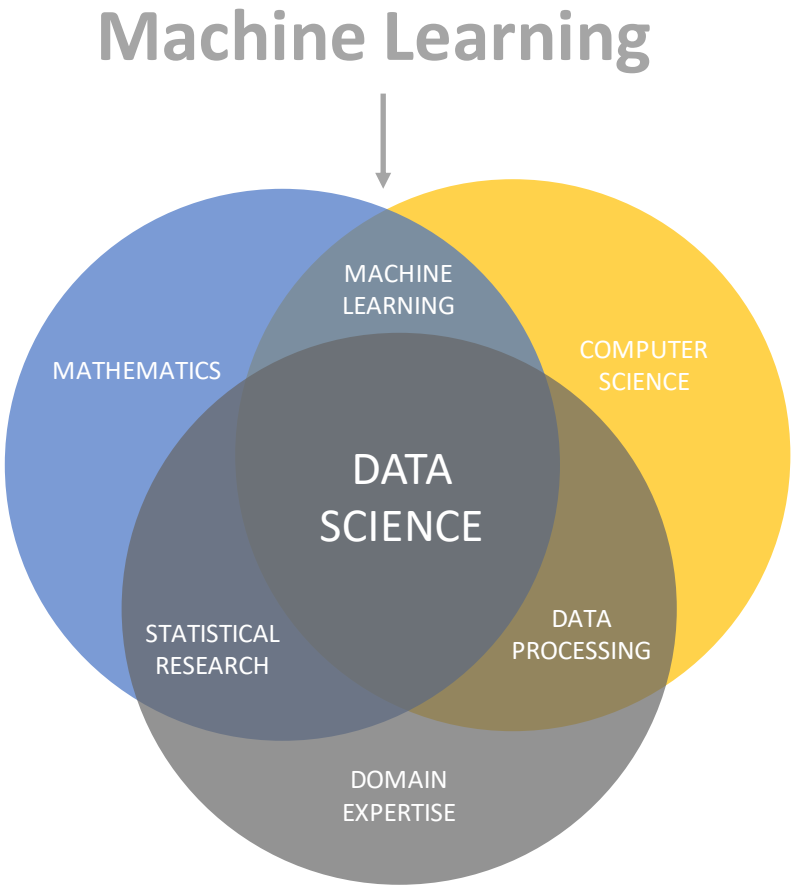


Что такое машинное обучение?

What is Data Science?

Wikipedia describes **Data Science** as:

“междисциплинарная область, которая использует научные методы, процессы, алгоритмы и системы для **извлечения знаний и информации** из структурированных и неструктурированных **данных**.”

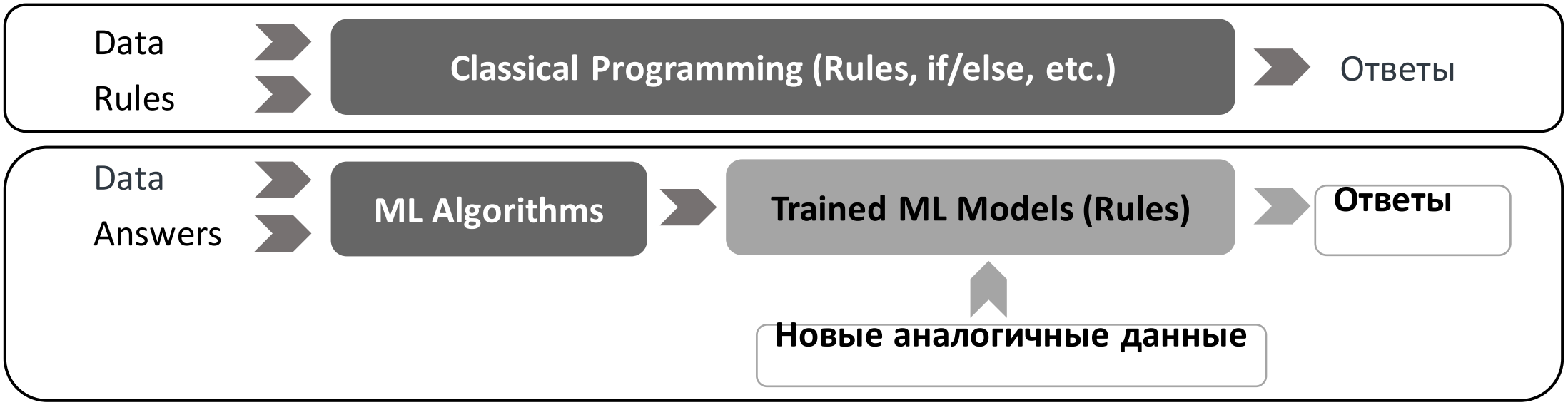


https://en.wikipedia.org/wiki/Data_science

What is Machine Learning?

“Машинное обучение (ML) - это основная ветвь искусственного интеллекта, цель которой - дать компьютерам возможность учиться без явного программирования.”

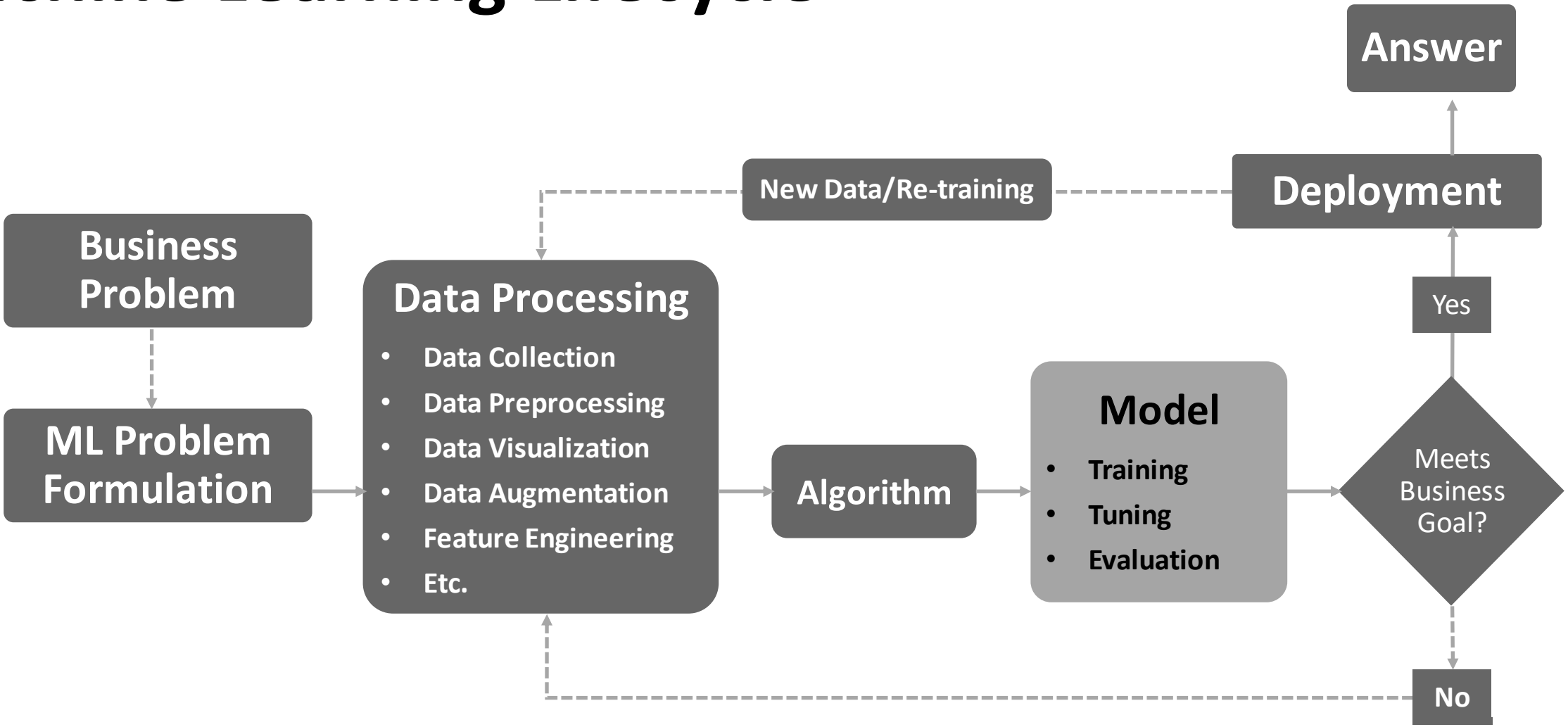
Arthur Samuel (1959) – Computer Scientist



Why ML? Why now?

- **Data**
  - большие объемы данных, легко производить, собирать и хранить
- **Compute**
  - мощные вычислительные единицы, аппаратное ускорение, параллелизация вычислений
- **Algorithms**
  - Рамки ML, библиотеки, улучшенные и более эффективные методы

Machine Learning Lifecycle



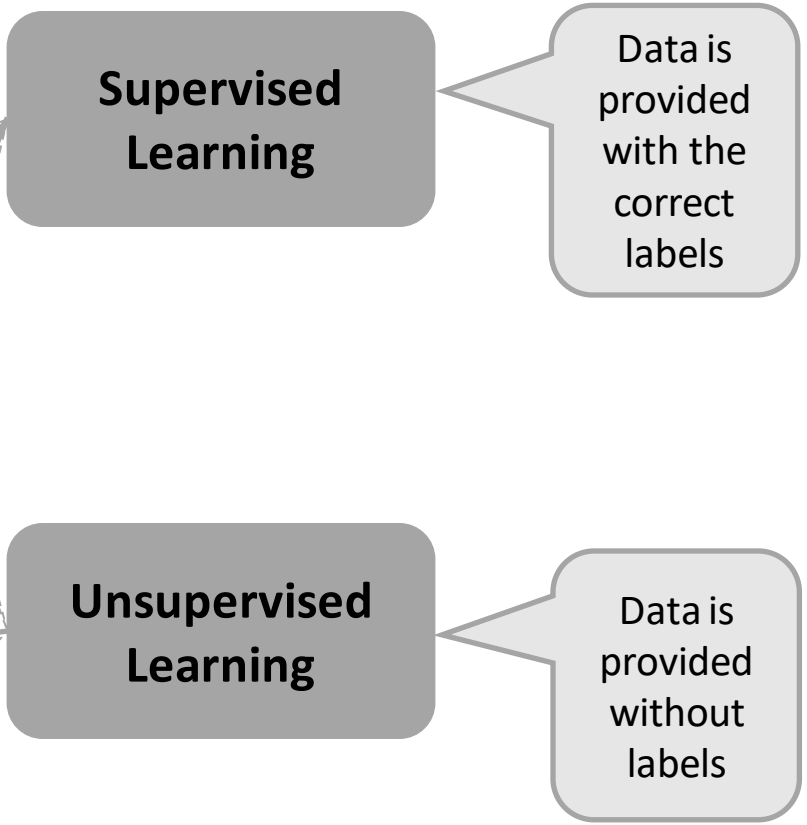
Некоторые важные понятия ML

ML	Statistics/Math/other	Simply Put
Label/Target/y	Dependent/Response/Output Variable	То, что вы пытаетесь предсказать
Feature/x	Independent/Explanatory/Input Variable	Данные, которые помогут вам делать прогнозы
Feature Engineering	Transformation	Изменение данных, чтобы получить больше значения
1d, 2d,… nd	Dimensionality	Количество функций
Model Parameters	Weights	Набор чисел, встроенных в модель, который может предсказывать метки
Model Training	Optimization	Поиск «лучшего» набора параметров модели

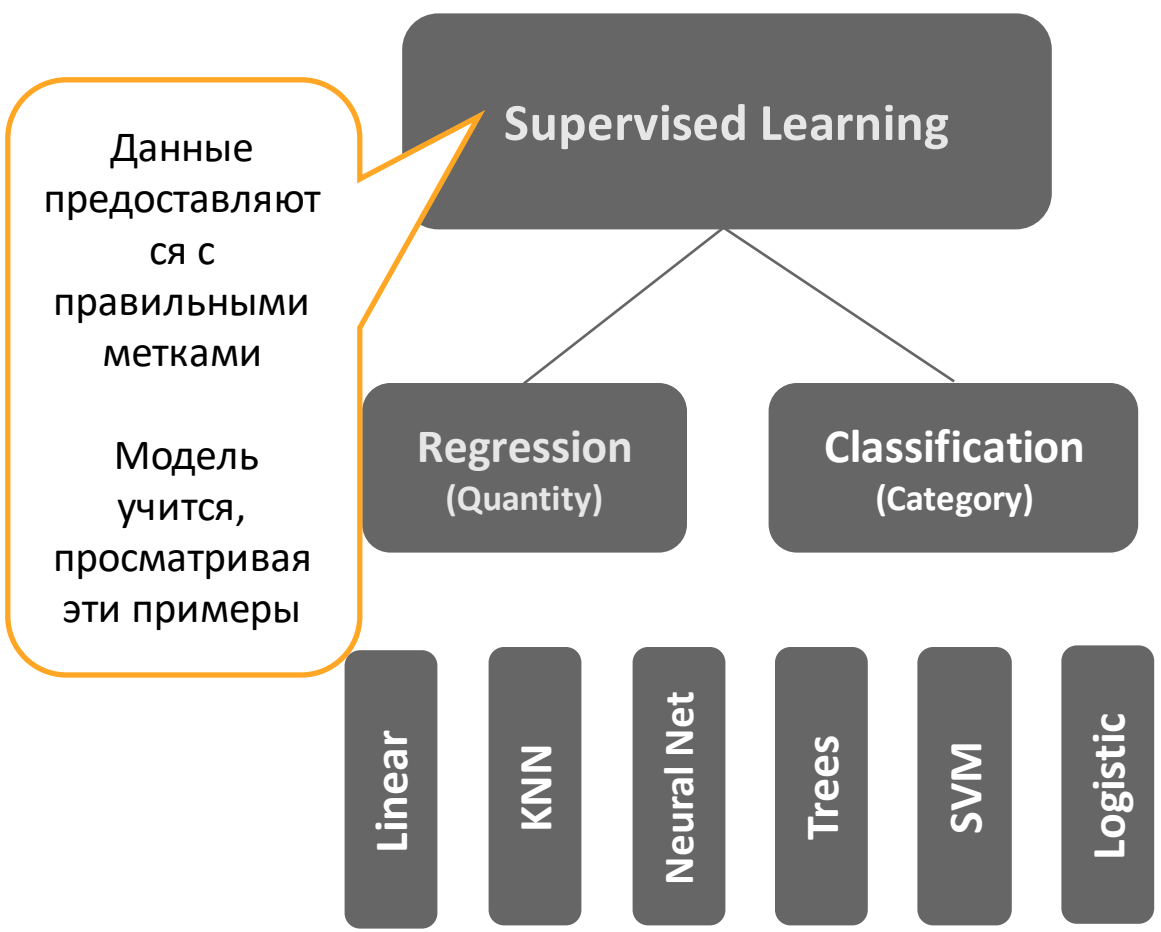
Контролируемое и неконтролируемое обучение

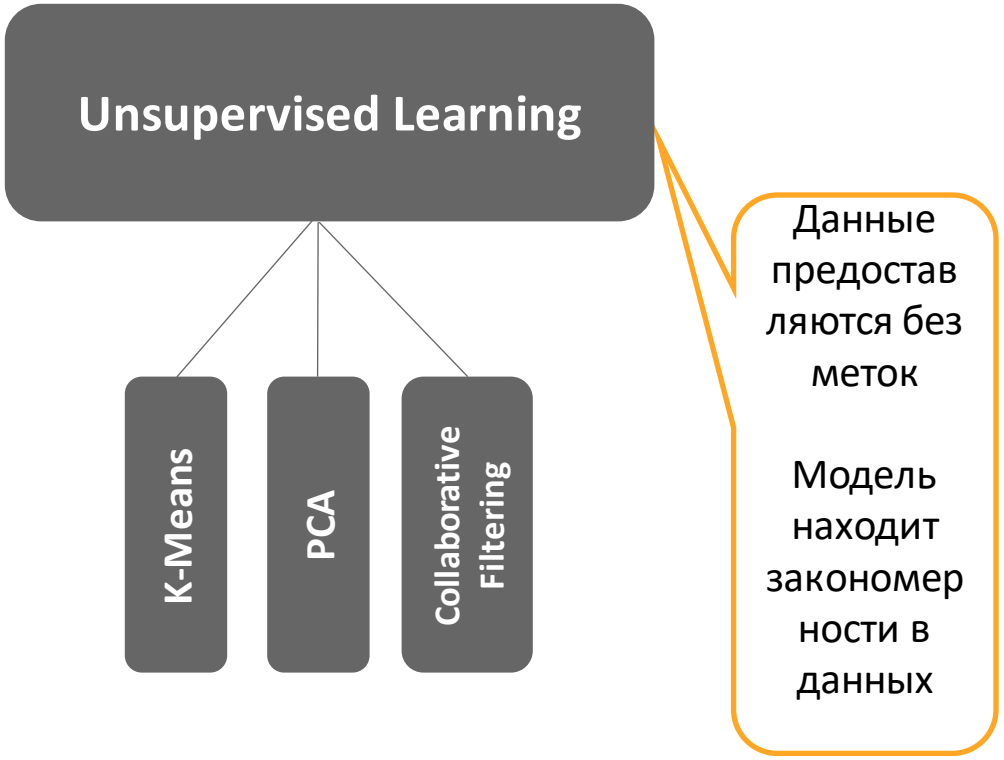
Supervised vs. Unsupervised Learning

Business/ML Problem	Description
Ranking	Помощь пользователям в поиске наиболее релевантной вещи
Recommendation	Предоставление пользователям то, что они могут быть наиболее заинтересованы в
Classification	Выяснение того, что что-то такое
Regression	Прогнозирование численного значения вещи
Clustering	Сложив подобные вещи вместе
Anomaly Detection	Поиск необычных вещей



Supervised vs. Unsupervised Learning

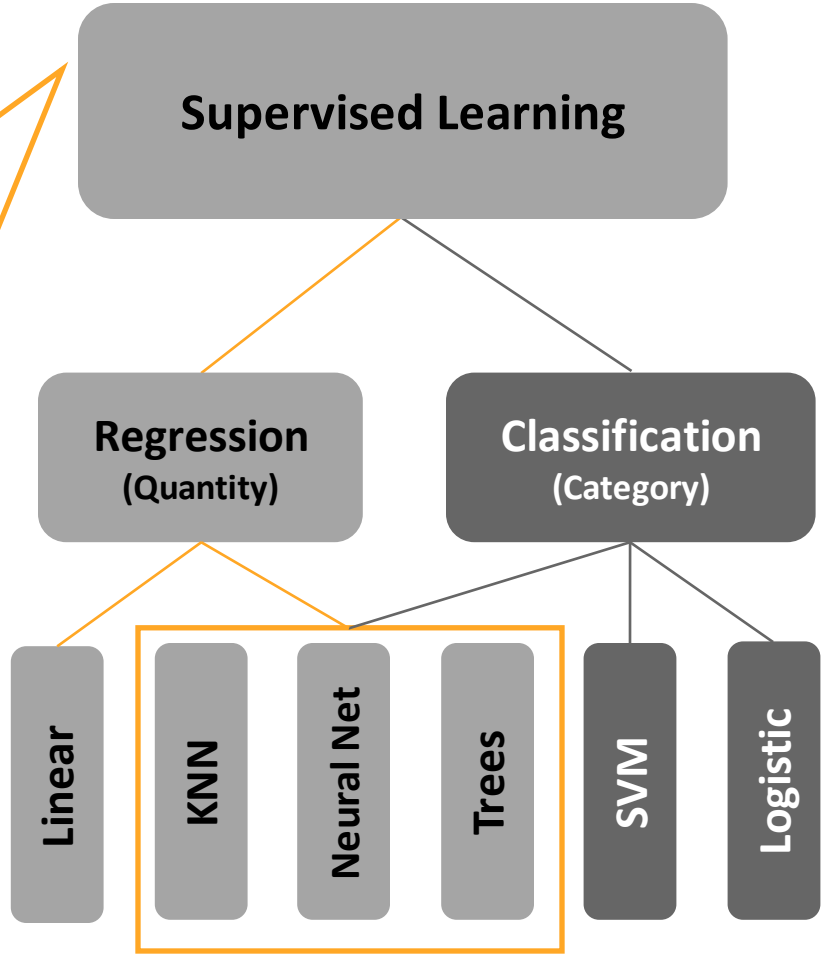


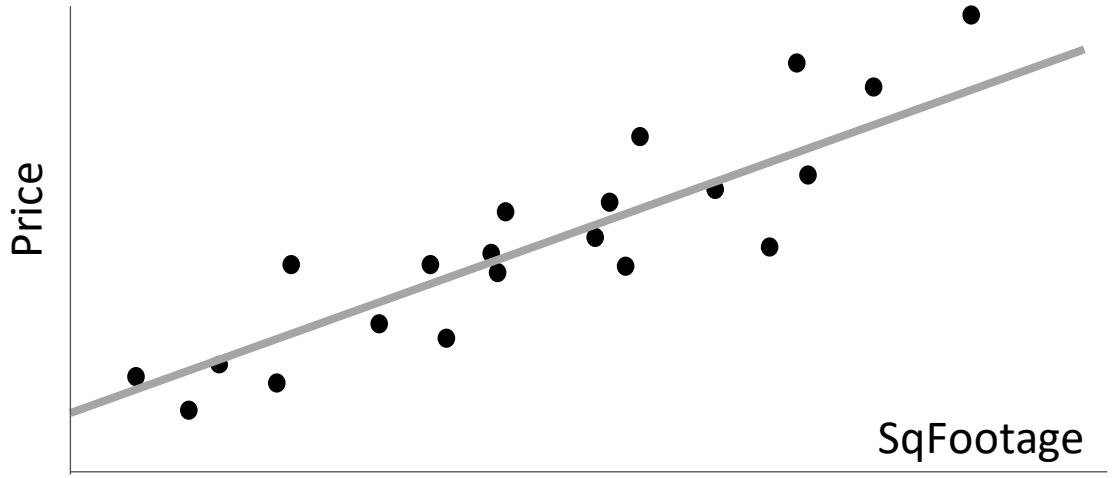


Supervised Learning: Regression

Data is provided with the correct labels

Model learns by looking at these examples





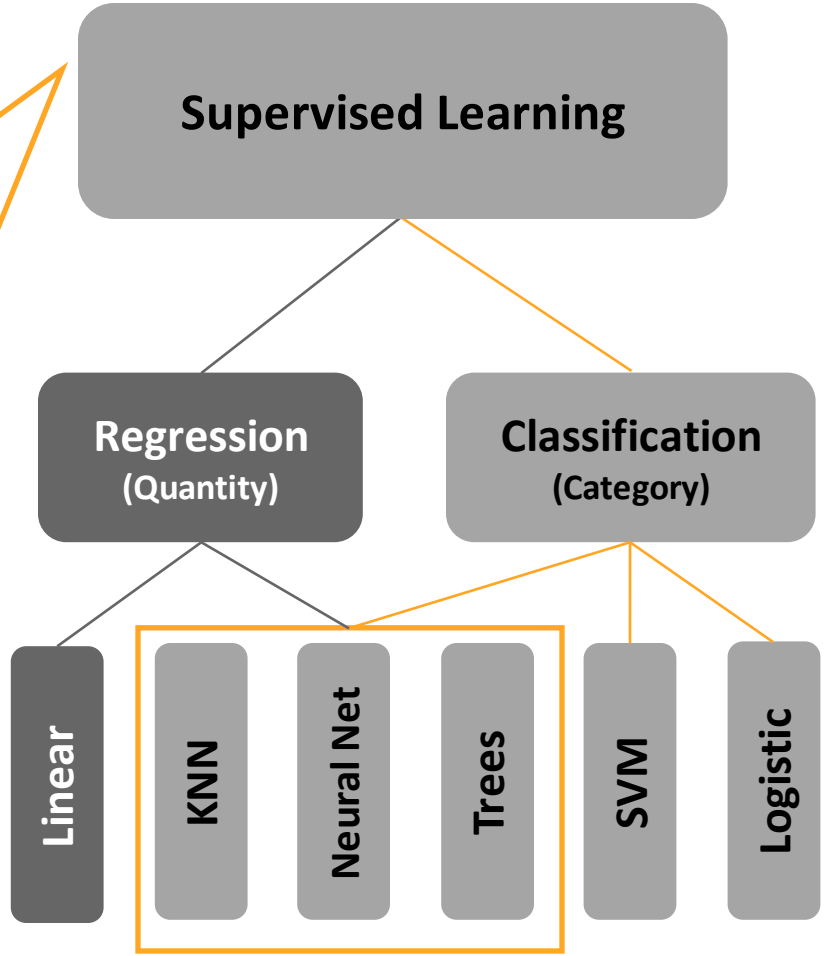
Label | Features

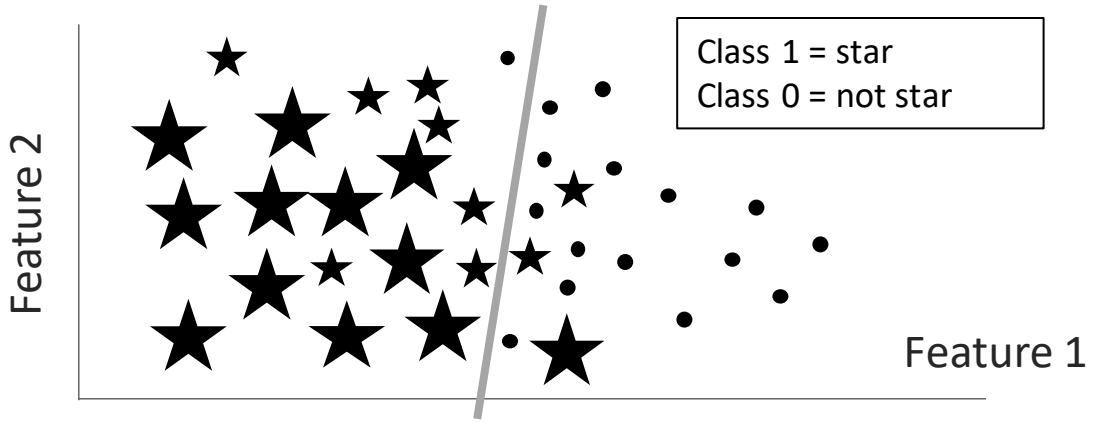
Price	Bedrooms	SqFootage	Age
280.000	3	3292	14
210.030	2	2465	6
…	…	…	…

Supervised Learning: Classification

Data is provided with the correct labels

Model learns by looking at these examples

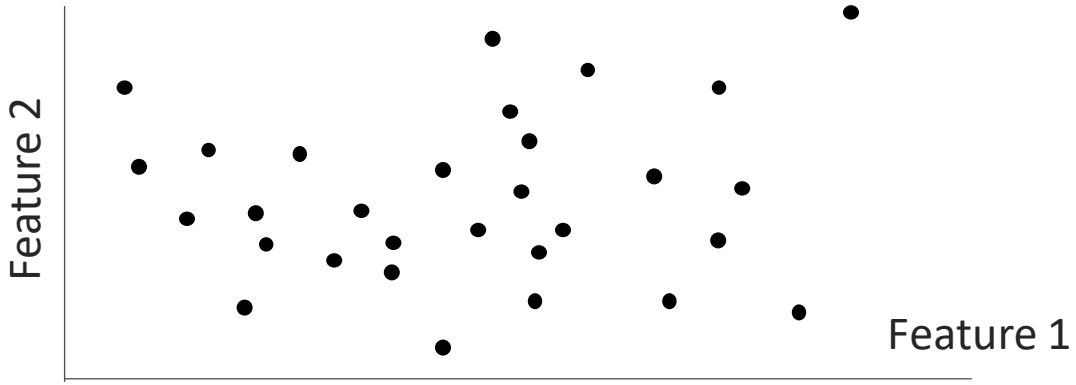




Label | Features

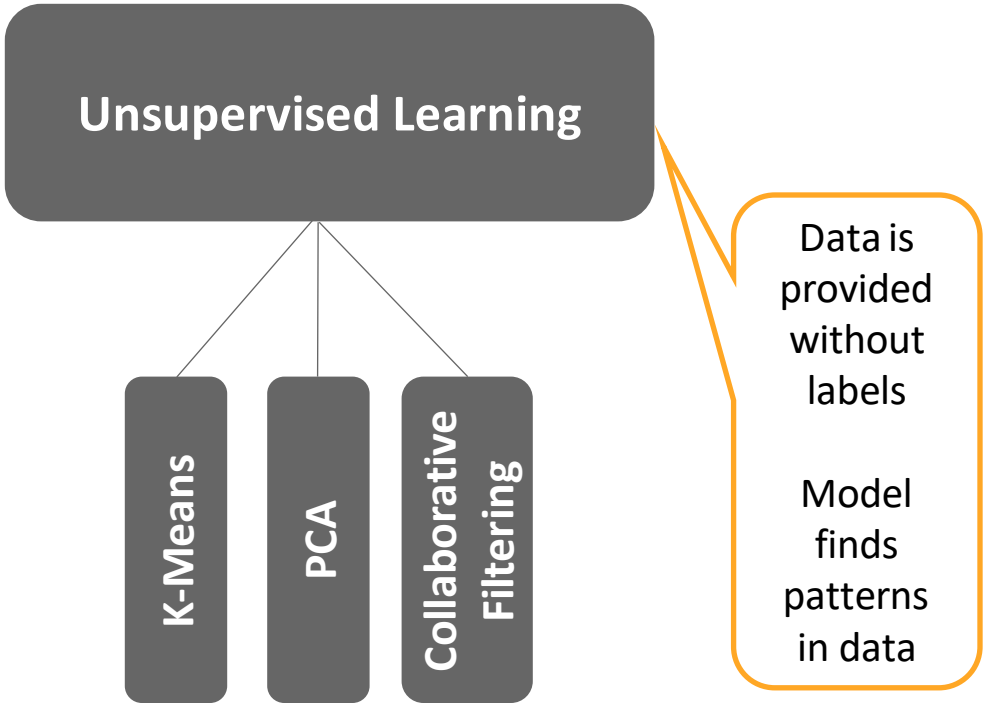
Star	Points	Edges	Size
1	5	10<	750
0	0	>9	150
…	…	…	…

Unsupervised Learning: Clustering

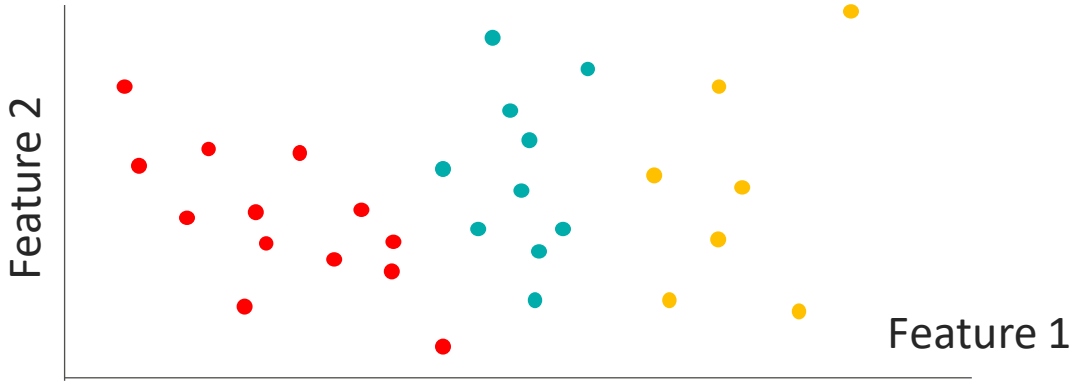


Features

Age	Music	Books
21	Classical	Practical Magic
47	Jazz	The Great Gatsby
…	…	…

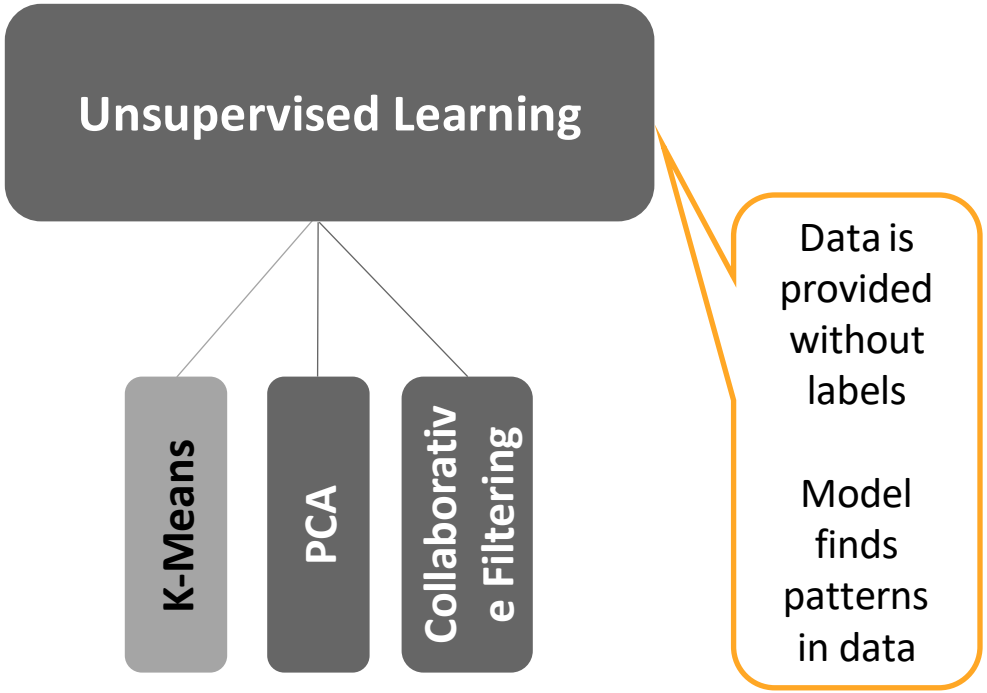


Unsupervised Learning: Clustering



Features

Age	Music	Books
21	Classical	Practical Magic
47	Jazz	The Great Gatsby
…	…	…



Sample ML Problem

Food Delivery Problem

- Джон любит заказывать еду онлайн для дома и работы.
- Он хочет предсказать, будет ли его заказ доставлен вовремя заранее.
- Он зарегистрировал свои предыдущие 45 заказов.

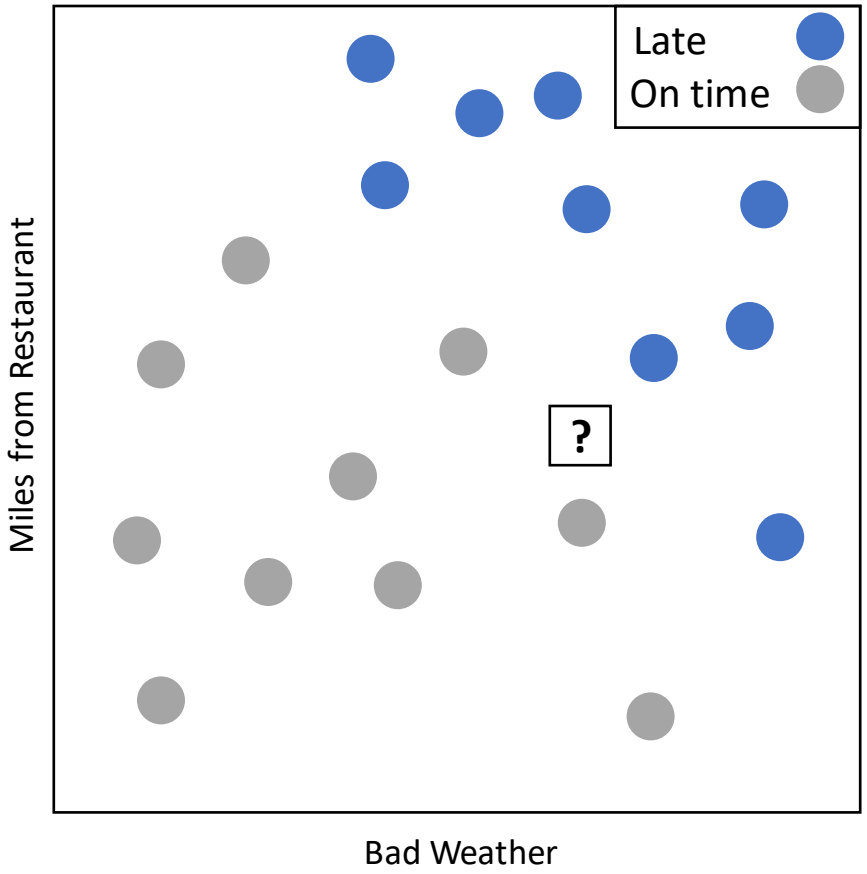
BadWeather Плохая погода	RushHour Час пик	MilesFromRestaurant Расстояние	UrbanAddress Городской адрес	Late
10	1	5	1	0
78	0	7	0	1
14	1	2	1	0
58	1	4.2	1	1
82	0	7.8	0	0
…	…	…	…	…

Two classes: 1/late and 0/on time

Food Delivery Problem

Метод **К ближайших соседей (KNN)** прогнозирует новые точки данных на основе К аналогичных примеров из набора данных (датасета).

К какому классу относятся? ?



Food Delivery Problem

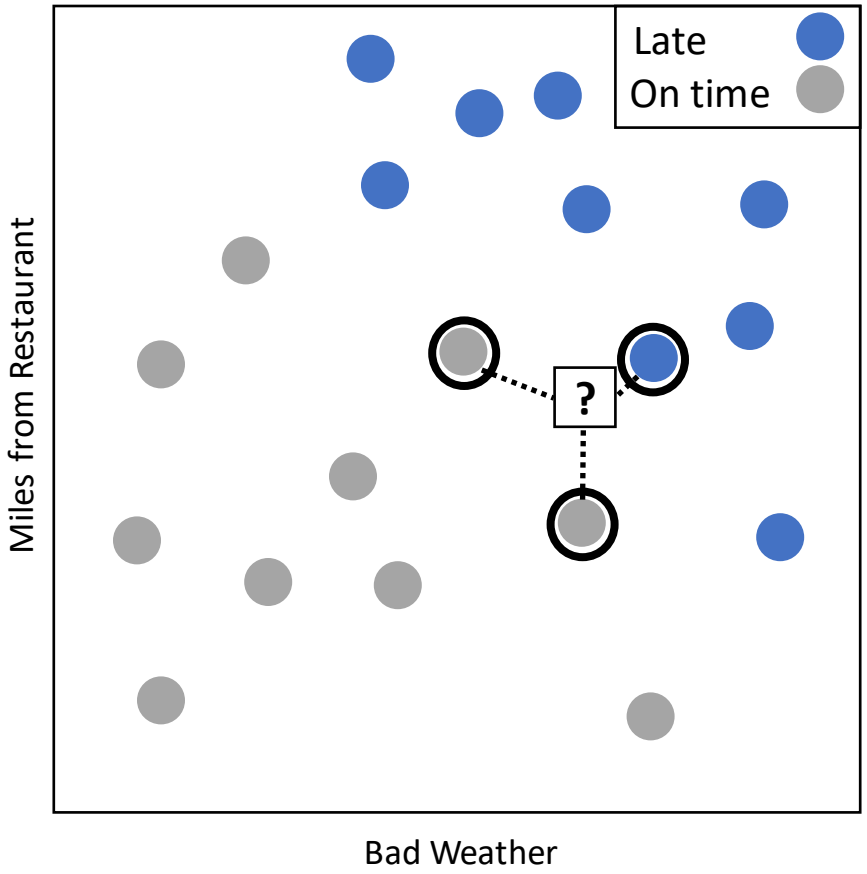
- **K Nearest Neighbors** (KNN) predicts new data points based on K similar records from a dataset.

К какому классу относятся? ?

Посмотрите на ближайшие К точки данных :

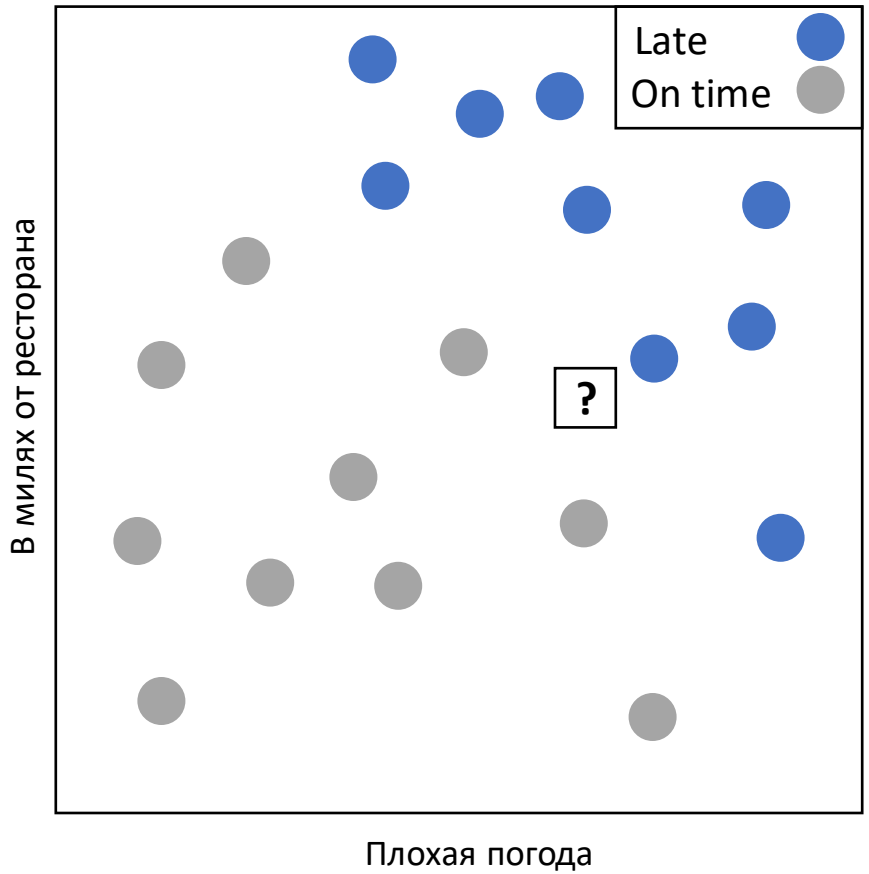
Выбрать К = 3

- Рассчитать расстояния от всех точек данных ?
- Найти ближайших соседей К
- Выберите класс большинства:



Food Delivery Problem Hands-on

- Давайте использовать пример доставки еды Джона и обучить алгоритм К ближайших соседей для прогнозирования новой точки данных.

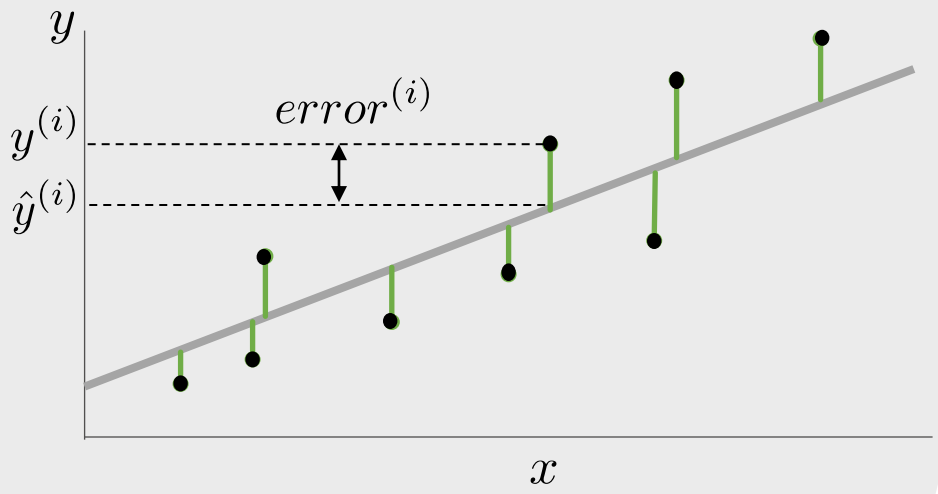


Model Evaluation

Regression Metrics

Metrics	Equations
Mean Squared Error (MSE)	$MSE = \frac{1}{n}\sum_{i=0}^{n}(y^{(i)} - \hat{y}^{(i)})^2$
Root Mean Squared Error (RMSE)	$RMS = \sqrt{\frac{1}{n}\sum_{i=0}^{n}(y^{(i)} - \hat{y}^{(i)})^2}$
Mean Absolute Error (MAE)	$MAE = \frac{1}{n}\sum_{i=0}^{n}\lvert y^{(i)} - \hat{y}^{(i)}\rvert$
R Squared (R^2)	$R^2 = 1 - \frac{\sum_{i=0}^{n}(y^{(i)} - \hat{y}^{(i)})^2}{\sum_{i=0}^{n}(y^{(i)} - \bar{y})^2}$

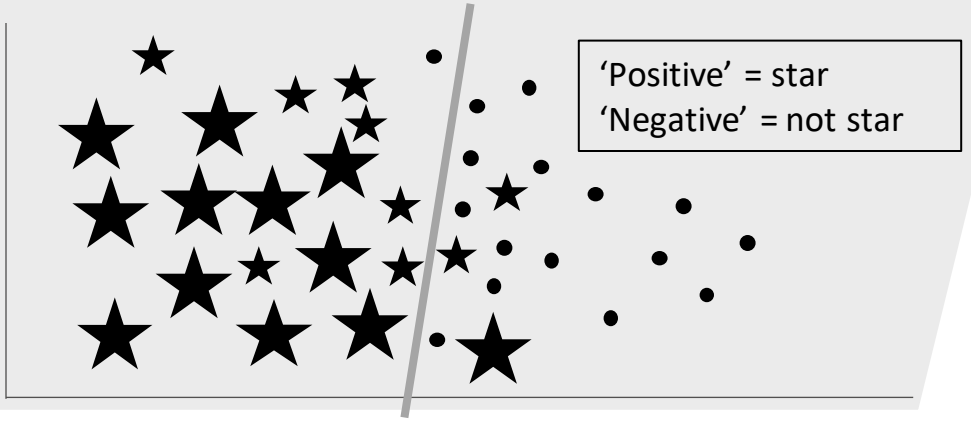
$y^{(i)}$: Data values
$\hat{y}^{(i)}$: Predicted values
$\bar{y}$: Mean value of data values, $\bar{y} = \frac{1}{n}\sum_{i=0}^{n} y^{(i)}$
n : Number of data records



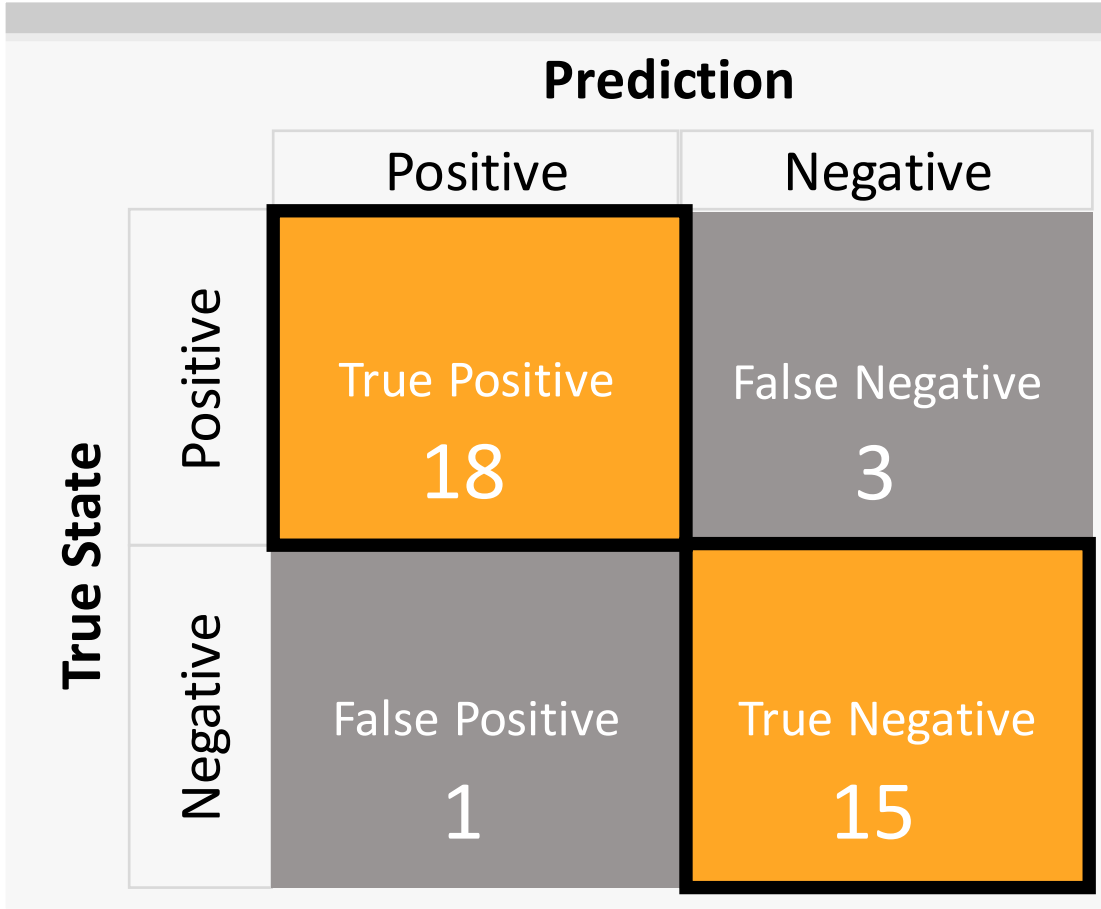
Classification Metrics

True State \ Prediction	Positive	Negative
Positive	True Positive 18	False Negative 3
Negative	False Positive 1	True Negative 15

True Positive: Predicted 'Positive' when the actual is 'Positive'
False Positive: Predicted 'Positive' when the actual is 'Negative'
False Negative: Predicted 'Negative' when the actual is 'Positive'
True Negative: Predicted 'Negative' when the actual is 'Negative'



Classification Metrics: Accuracy



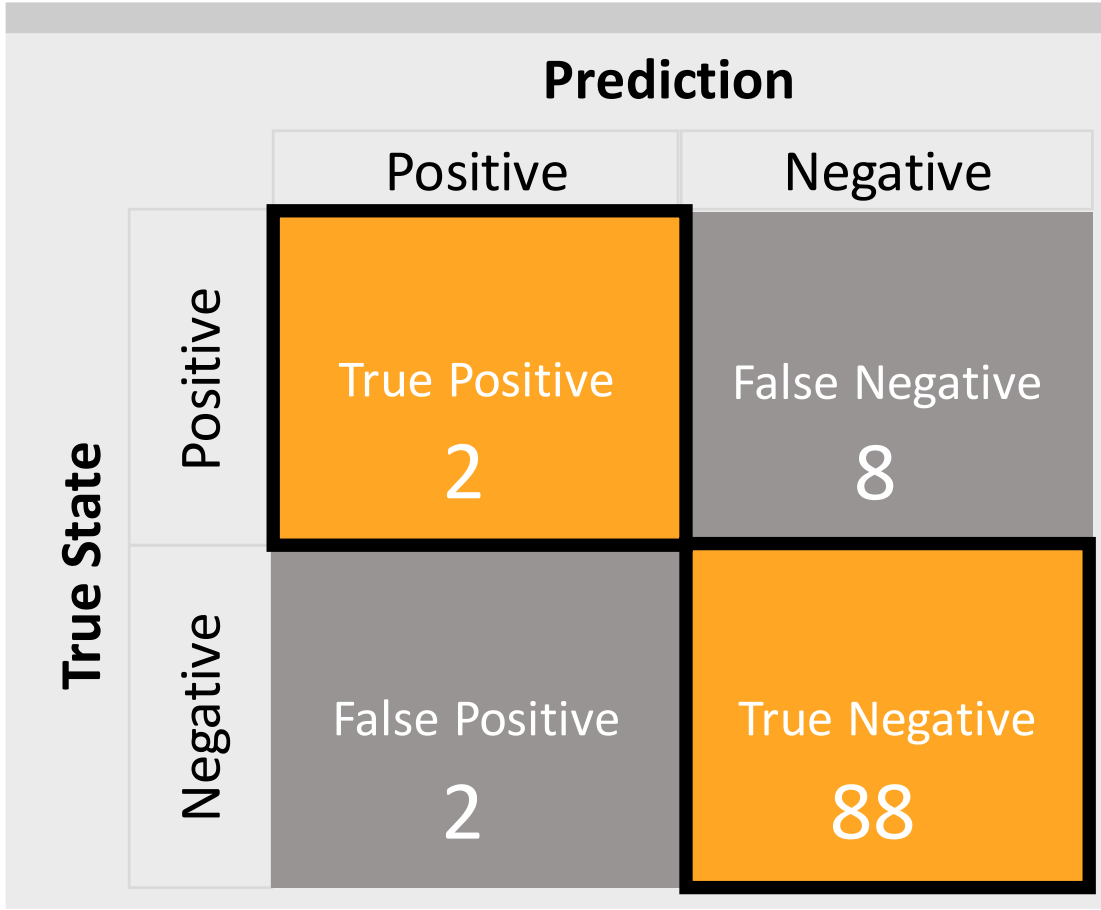
Accuracy*: Процент (коэффициент) случаев, классифицированных правильно

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

$$Accuracy = \frac{18 + 15}{18 + 1 + 3 + 15} = 0.89$$

$^{*}(bad)\ 0 \leq Accuracy \leq 1\ (good)$

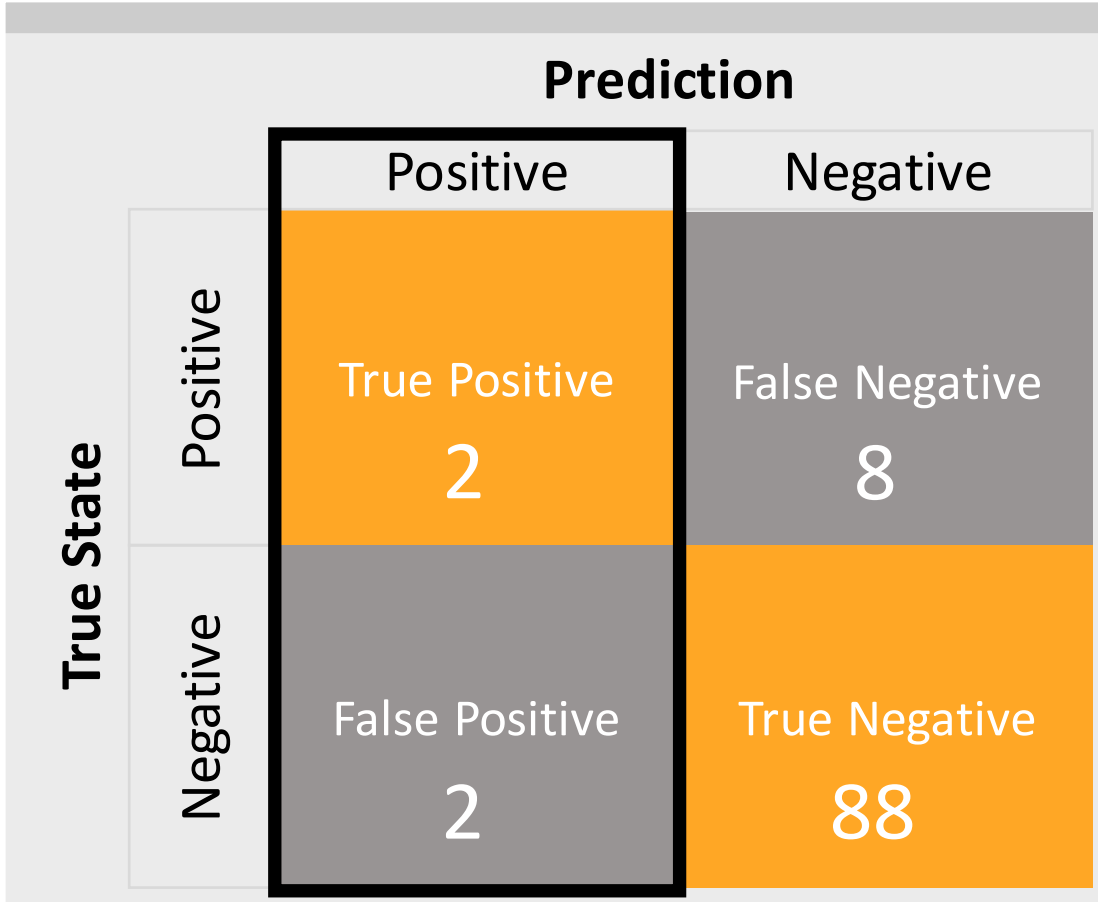
Classification Metrics: Accuracy



Парадокс высокой точности: Точность вводит в заблуждение при работе с несбалансированными набор данных - несколько True Positives, "редкий" класс, и многие True Negatives, "доминирующий" класс. **Высокая точность даже тогда, когда мало истинных срабатываний.**

$$Accuracy = \frac{2 + 88}{2 + 2 + 8 + 88} = 0.90$$

Classification Metrics: Precision



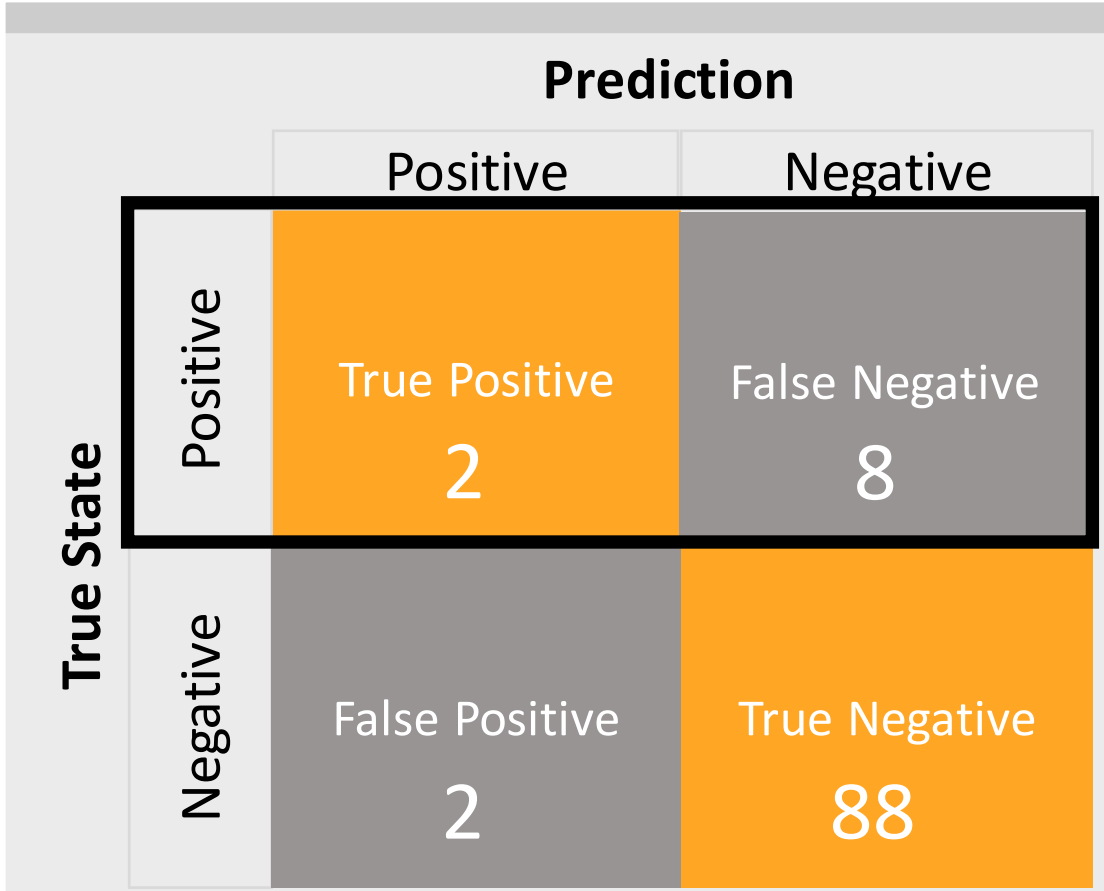
Precision*: Точность прогнозируемого положительного результата

$$Precision = \frac{TP}{TP + FP}$$

$$Precision = \frac{2}{2 + 2} = 0.50$$

$^{*}(bad)\ 0 \leq Precision \leq 1\ (good)$

Classification Metrics: Recall



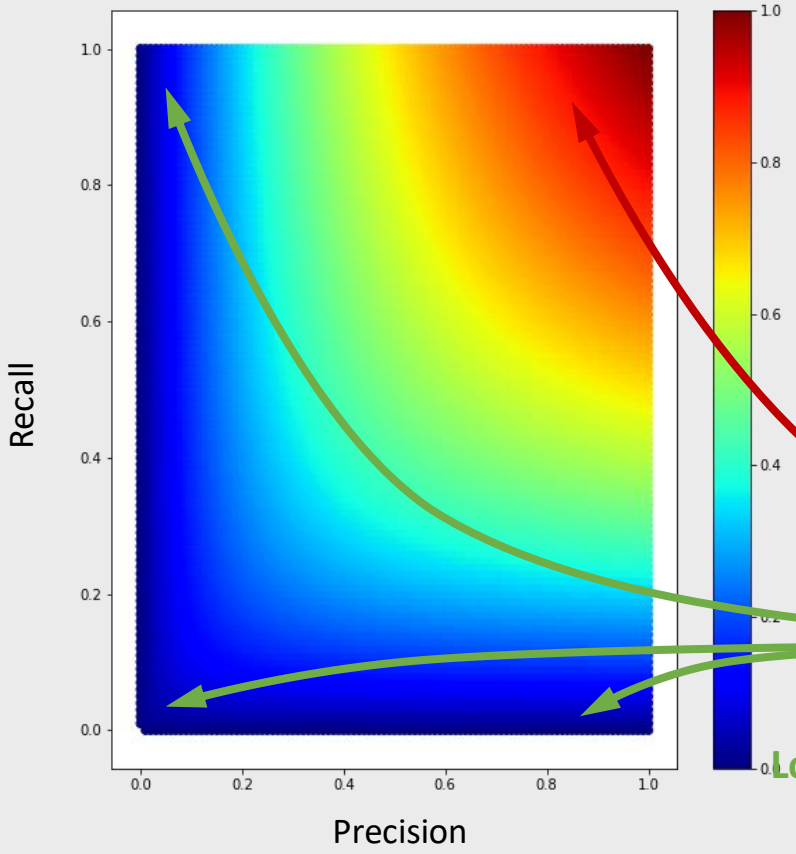
Recall*: Измеряет способность модели прогнозировать положительный результат

$$Recall = \frac{TP}{TP + FN}$$

$$Recall = \frac{2}{2 + 8} = 0.20$$

$^{*}(bad)\ 0 \leq Recall \leq 1\ (good)$

Classification Metrics: F1 Score



F1 Score*: A combined metric, the harmonic mean of Precision and Recall.

$$F1\,Score = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

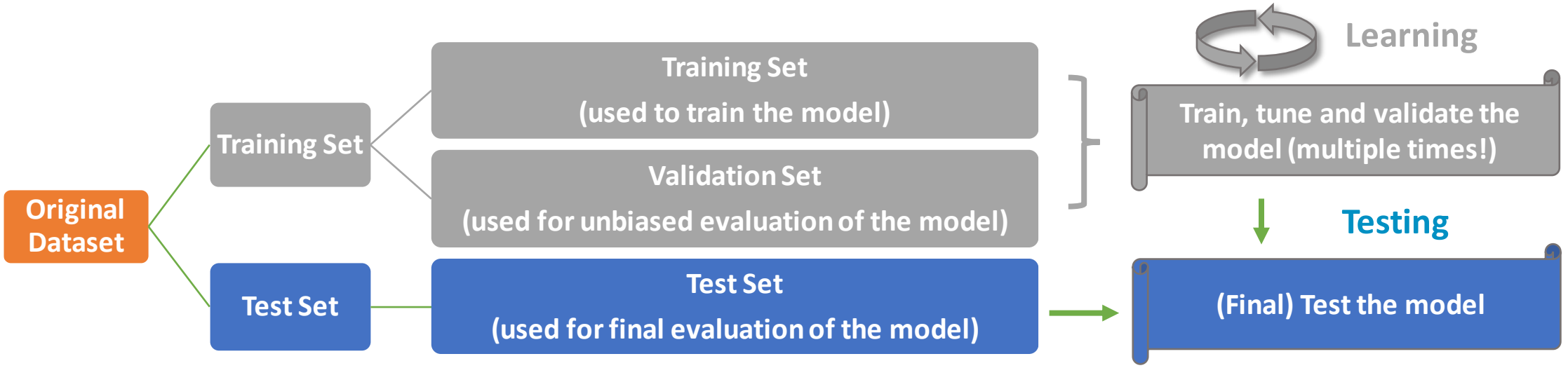
$$^*(bad)\ 0 \leq F1\,Score \leq 1\ (good)$$

Low when one or both of the Precision and Recall are low

High when both Precision and Recall are high

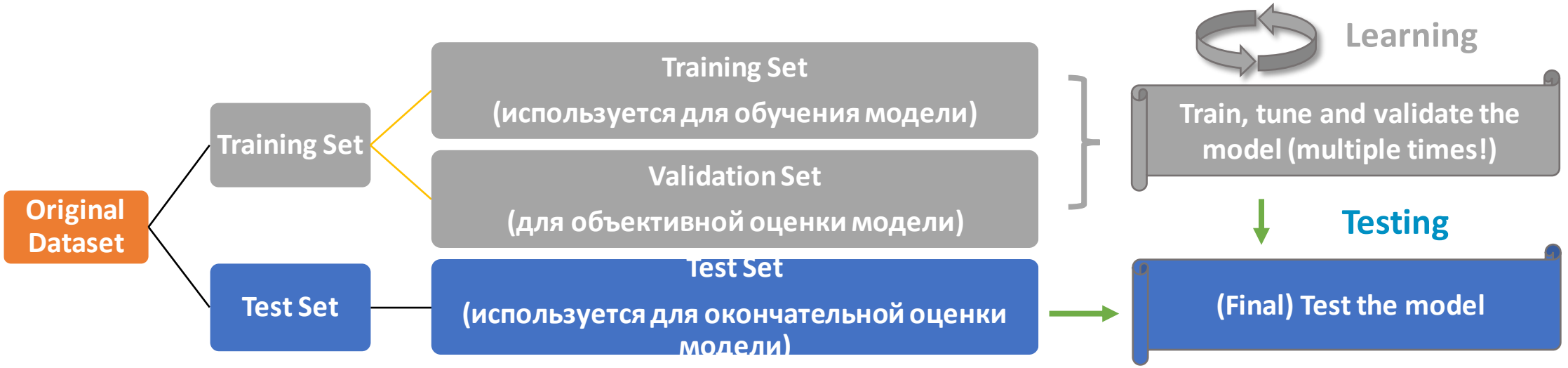
Train – Validation – Test Datasets

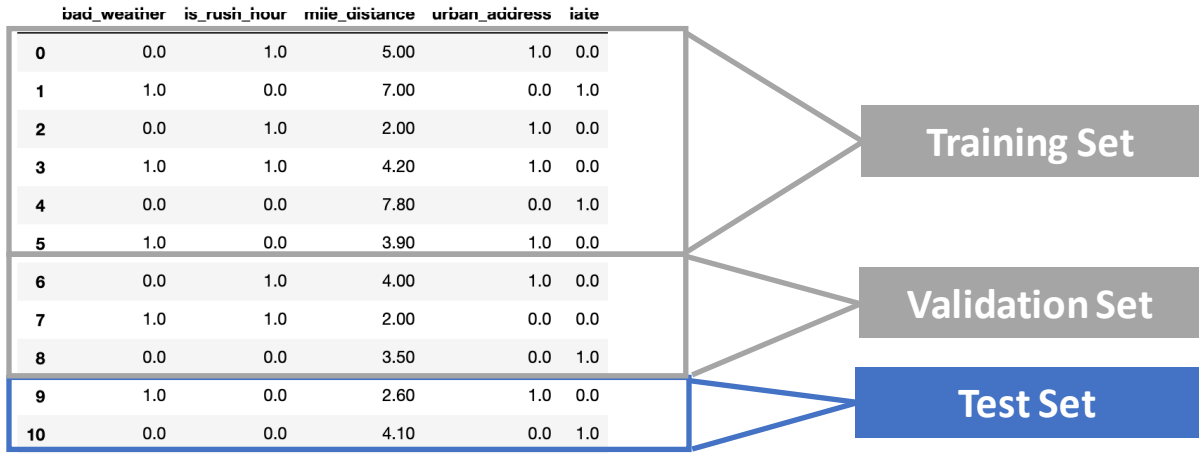
Training – Validation – Test Sets



Тестовый набор недоступен модели при обучении, он используется только обобщения (прогноза) модели на новых данных (данных, на которых она не обучалась)

Training – Validation – Test Sets

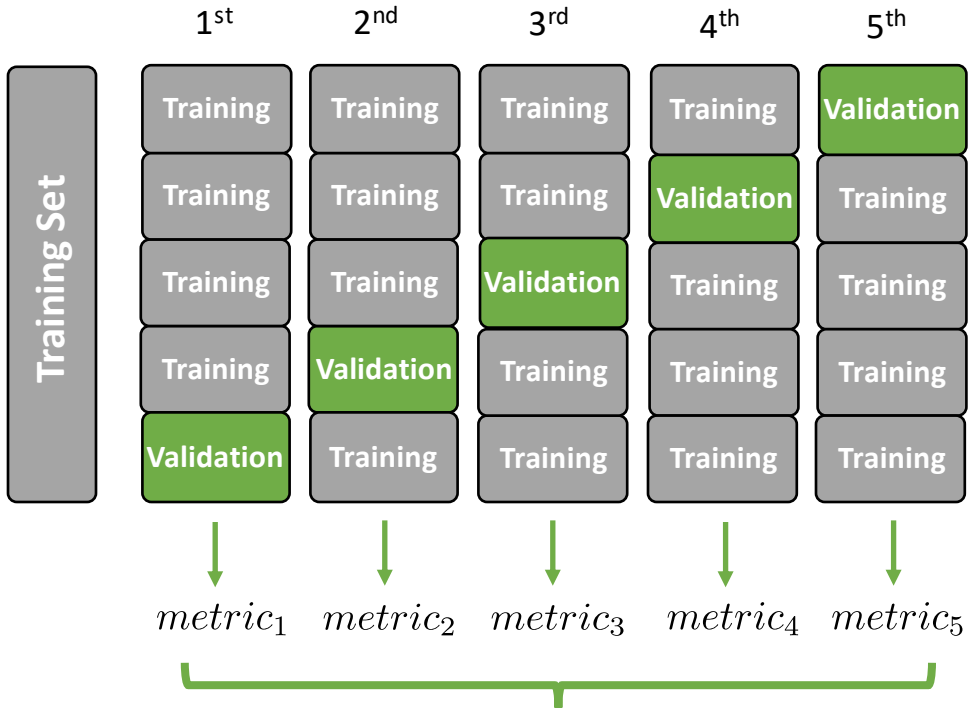




	bad_weather	is_rush_hour	mile_distance	urban_address	late
0	0.0	1.0	5.00	1.0	0.0
1	1.0	0.0	7.00	0.0	1.0
2	0.0	1.0	2.00	1.0	0.0
3	1.0	1.0	4.20	1.0	0.0
4	0.0	0.0	7.80	0.0	1.0
5	1.0	0.0	3.90	1.0	0.0
6	0.0	1.0	4.00	1.0	0.0
7	1.0	1.0	2.00	0.0	0.0
8	0.0	0.0	3.50	0.0	1.0
9	1.0	0.0	2.60	1.0	0.0
10	0.0	0.0	4.10	0.0	1.0

Это хорошая практика, чтобы перетасовать набор данных до разделения, чтобы избежать смещения в результате наборов.

K-fold Cross Validation (K = 5)



K-fold Cross-Validation (CV) это метод проверки, чтобы увидеть, насколько хорошо обученная модель обобщается для независимого набора проверки.

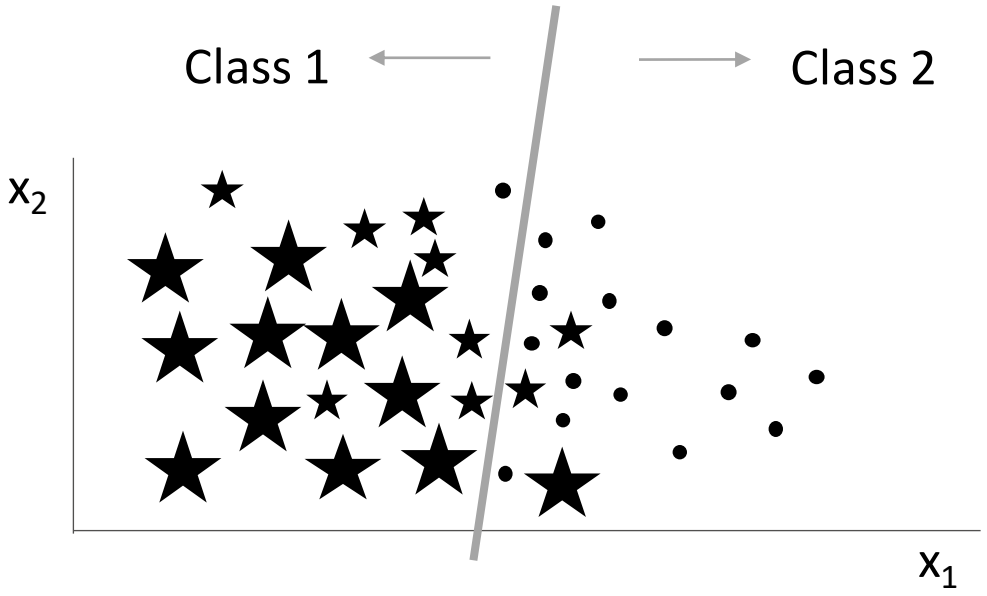
Используйте K различные набора (фолда) для проверки модели, каждый раз обучая остальные примеры:

- Разделение датасета на независимые поднаборы (фолды) K.
- Повторите следующие действия K:
  - Зафиксируйте фолд K данных - **test set**.
  - Тренируйте модель на других фолдах (поднаборах) - **train set**.
  - Проверьте модель на **validation set**.
- Усредните (объедините) метрики качества модели.

Underfitting & Overfitting

Model Evaluation: Underfitting

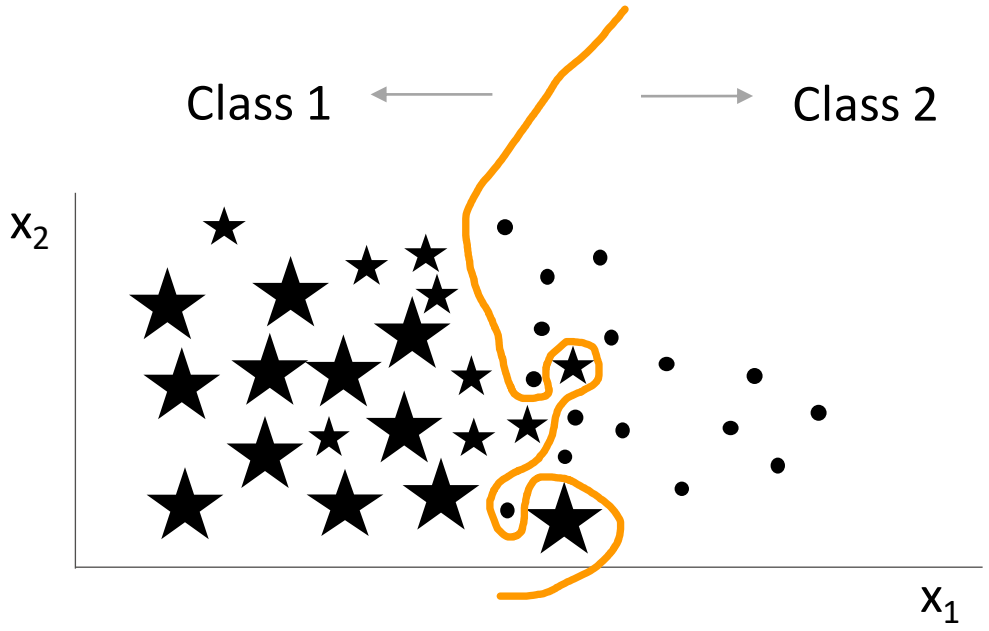
Underfitting: Модель недостаточно хороша для описания взаимосвязи между входным данными (x_1, x_2) и выходом y: : {Class 1, Class 2}.



- Модель **слишком проста** для определения важных закономерностях в обучающих данных.
- Модель **будет плохо работать** на и на обучающих и на тестовых (новых) данных

Model Evaluation: Overfitting

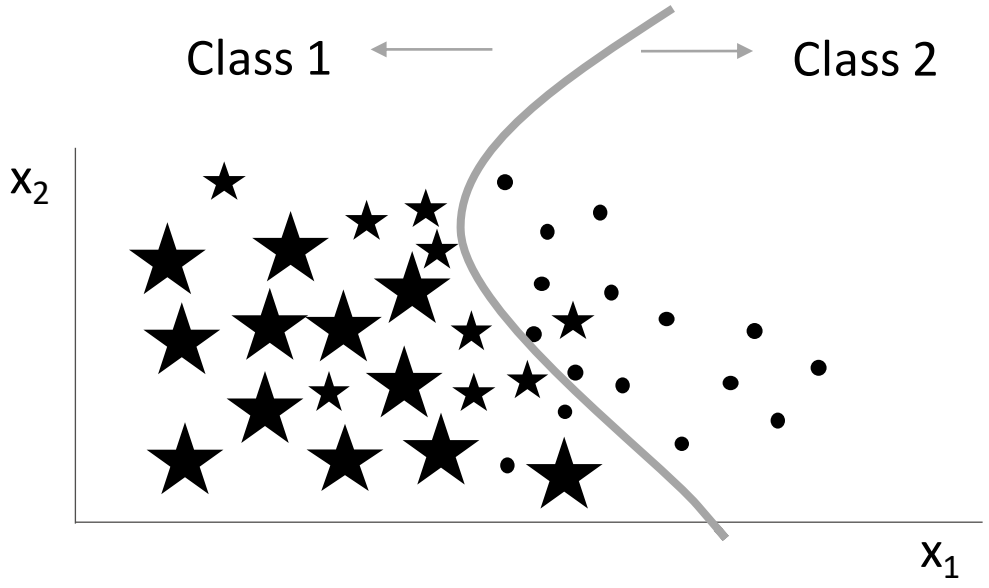
Overfitting: Модель запоминает или имитирует учебные данные и не может хорошо обобщить новые «невидимые» данные (тестовые данные).



- Модель **слишком сложна**.
- Модель запоминает шум (выбросы и аномалии) вместо лежащих в основе отношений (закономерностей).
- Модель будет хорошо работать на обучающих данных, и плохо на тестовых.

Model Evaluation: Good Fit

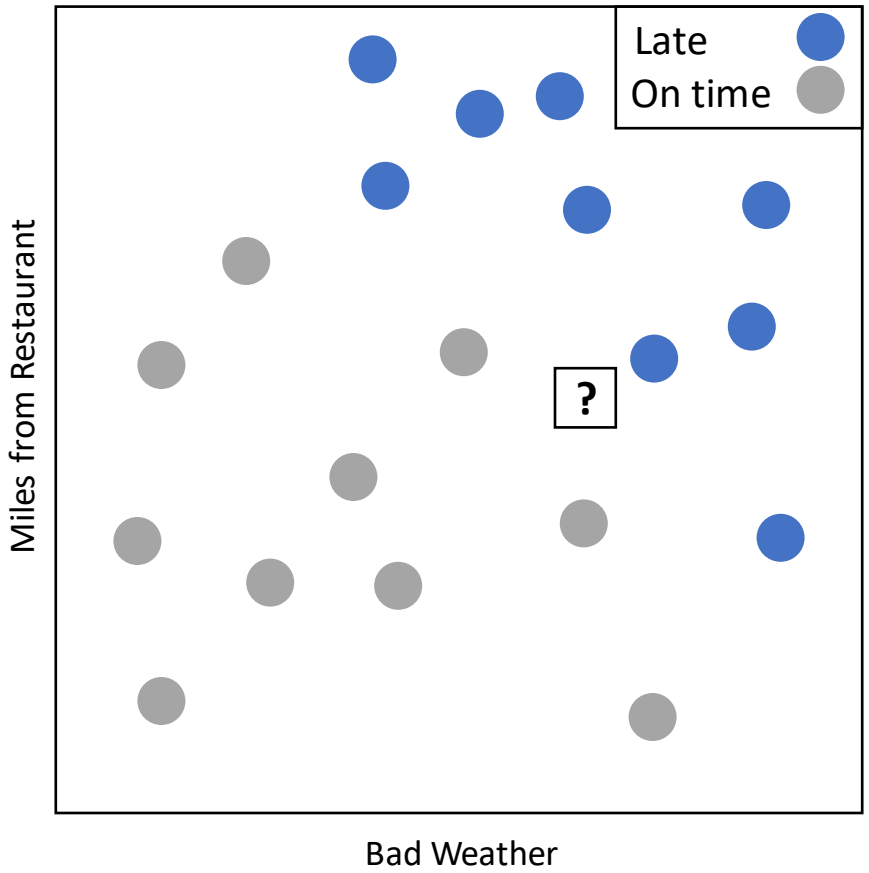
Appropriate fitting: Модель фиксирует общую взаимосвязь между входным данными (x_1, x_2) и выходом y: {Class 1, Class 2}.



- Модель не слишком простая, не слишком сложная.
- Модель находит основные закономерности в данных, а не шум.
- Модель будет достаточно хорошо работать и на обучающих и тестовых данных.

Overfitting Hands-on

- Давайте еще раз возьмем пример доставки еды Джона.
- Мы обучаем модель К ближайших соседей и анализируем переобучение.



Exploratory Data Analysis (EDA)

Exploratory Data Analysis

- **Exploratory Data Analysis (EDA)** это подход к анализу набора данных и определении основных его характеристик.

- **Сбор (Collect)** или агрегирование данных
- **Выполните первоначальные исследования, чтобы обнаружить закономерности, точечные аномалии, проверить гипотезу и проверить предположения**
  - Краткая статистика
  - Graphical/visual представления (histograms, plots)
- **Процесс данных** для получения значимой информации

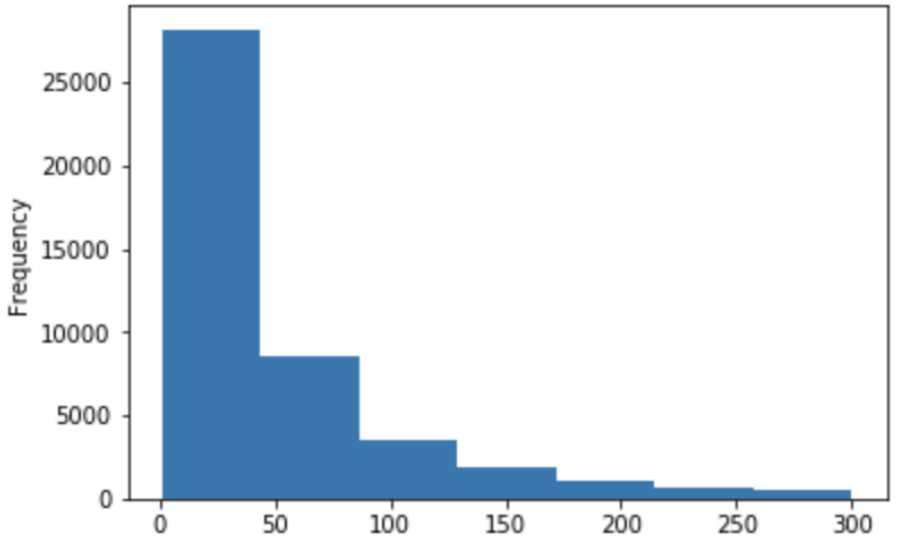
Descriptive Statistics

- **Overall statist** df.head(), df.shape, df.info()
  - Количество примеров (i.e. number of rows)
  - Количество функций (i.e. number of columns)
- **Univariate statistics** (single feature)
  - Статистика по численным характеристикам (mean, variance, histogram) -
  - Статистика по категоричным признакам (histograms, mode, most/least frequent values, percentage, number of unique values) df.describe(), hist(df[feature])
    - Histogram of value df[feature].value_counts() or seaborn's distplot()
  - Target statistics
    - Class distributio df[target].value_counts() or np.bincount(y)
- **Multivariate statistics** (more than one feature)
  - Correlatio df.plot.scatter(feature1, feature2), df[[feature1, feature2]].corr()

Univariate Statistics: Histograms

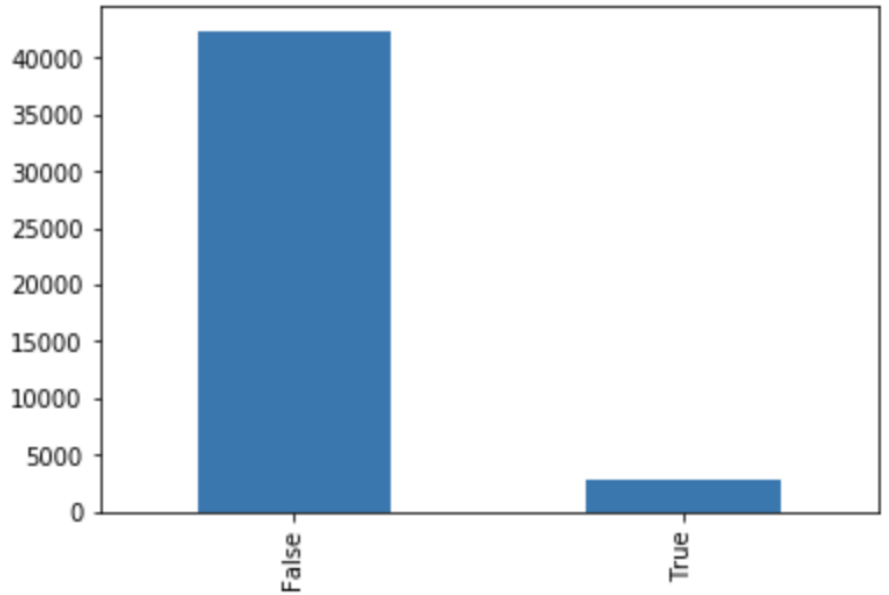
Numerical features:

```
import matplotlib.pyplot as plt

df[num_feature].plot.hist(bins = 7)
plt.show()
```



Categorical features:

```
import matplotlib.pyplot as plt

df[cat_feature].value_counts().plot.bar()
plt.show()
```

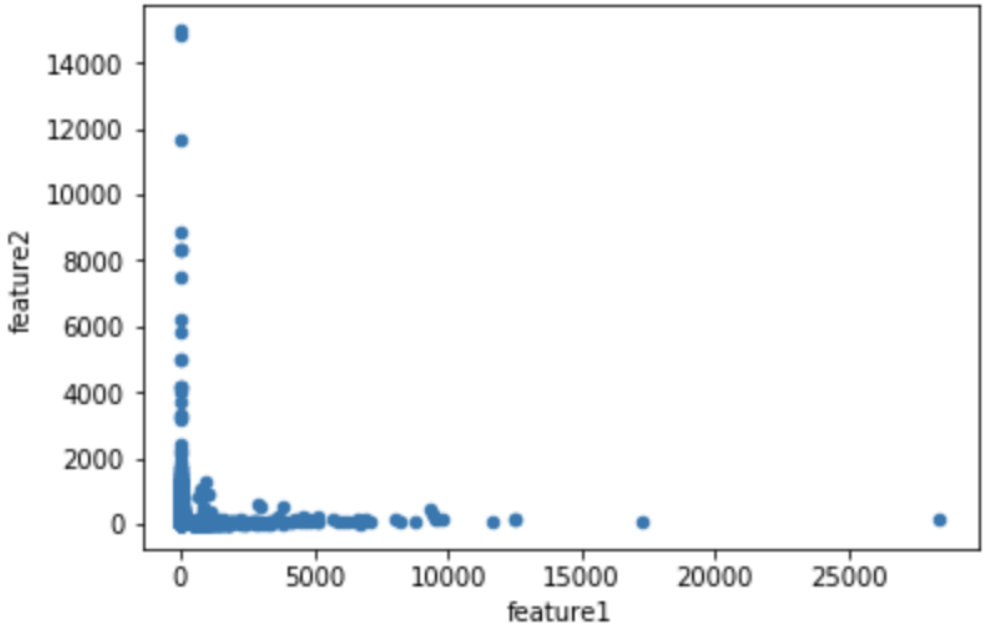


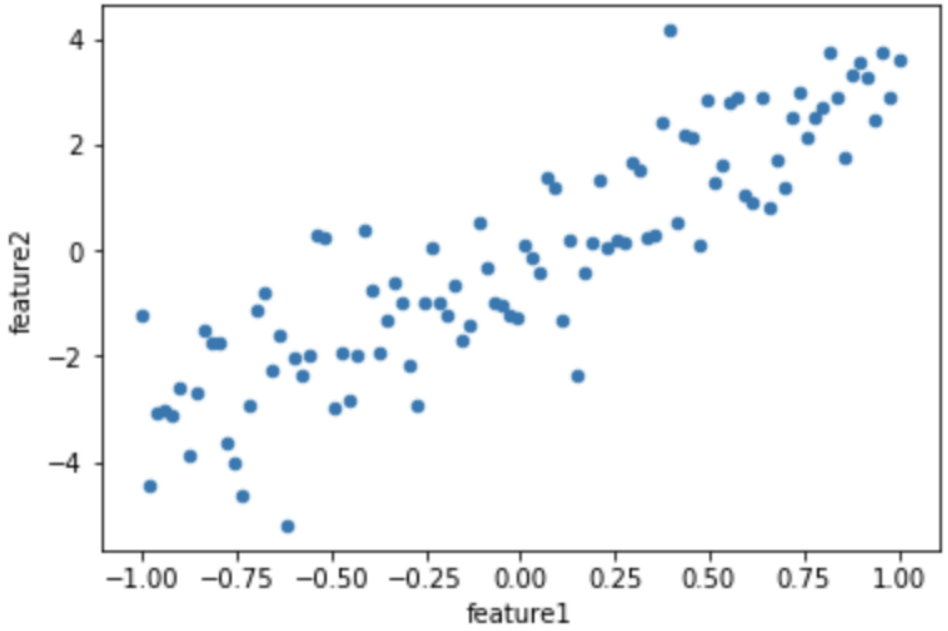
Correlations: Scatterplot

- **Correlations**: Насколько сильно связаны пары функций.

```
df.plot.scatter(feature1,feature2)
plt.show()
```

Матрицы точечной диаграммы визуализируют взаимосвязь между признаками.





Correlations: Correlation Matrix

- **Correlations**: How strongly pairs of features are related.

```
cols = [feature1, feature2]
df[cols].corr()
```

Матрицы корреляции измеряют **линейную** зависимость между признаками; легче читать; могут использовать тепловые карты.

	feature1	feature2
feature1	1	0.0128493
feature2	0.0128493	1

	feature1	feature2
feature1	1	0.882106
feature2	0.882106	1

Значения **корреляции** между -1 и 1: -1 означает идеальную отрицательную корреляцию, 1 означает идеальную положительную корреляцию, а 0 означает отсутствие связи между двумя переменными.

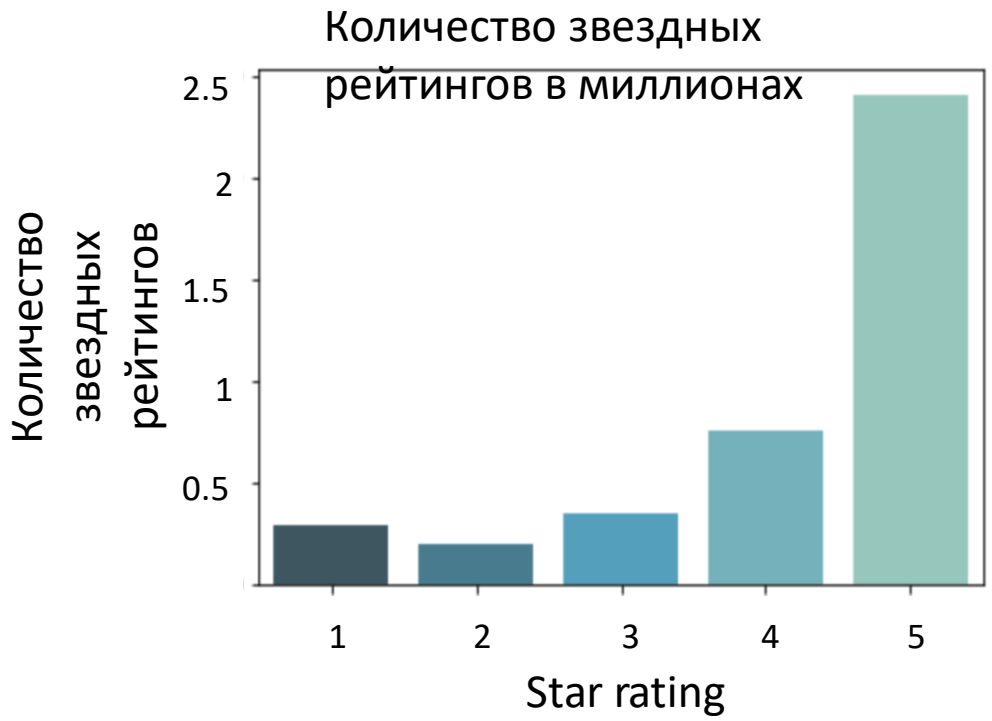
Correlations

- **Высоко коррелированные** (положительные или отрицательные) функции могут ухудшить производительность некоторых моделей ML, таких как линейные и логистические модели регрессии.

  - Выберите одну из коррелированных функций и отбросьте другую (ие).
  - Другие модели ML, такие как деревья решений, в основном невосприимчивы к этой проблеме.
- В то время как высоко **target-correlated** (положительные или отрицательные) функции могут повысить производительность линейных и логистических регрессионных моделей.
-

Imbalanced Datasets

Классовый дисбаланс



- Количество образцов в классе распределяется неравномерно.
- Модель ML может плохо работать для редких классов.
- **Examples**:
  - Обнаружение мошенничества
  - Обнаружение аномалий
  - Медицинская диагностика

Amazon review dataset: Количество 5 звездных отзывов почти равно сумме 4 других типов звездных отзывов вместе взятых.

Class Imbalance

- Как решить проблемы **классового дисбаланса**?

Down-sampling	Up-sampling	Data generation	Sample weights
Уменьшите размер доминирующего или частого класса	Увеличьте размер редкого или малого класса	Создавайте новые записи (примеры), похожие, но не идентичные.	Для модели, в которой используется функция стоимости (Loss), присвойте более высокие веса редким классам и более низкие веса доминирующим классам.

Missing Data

Обработка Missing Data

- Удалить (Drop) строки и / или столбцы с пропущенными значениями: удалите эти строки и / или столбцы из набора данных.
  - Меньшее количество примеров обучающих данных и / или меньшее количество функций может привести к переобучению / недообучению
  - Внести (заполнить) недостающие значения:
  - **Среднее значение** отсутствующих числовых значений: замените средним значением в столбце - `df[‘col'].fillna((df['col'].mean()))` `df['col'].fillna((df['col'].mode()))`
  - Расчет по общей точке для отсутствующих категориальных значений: замените наиболее распространенным значением для этого **столбца**
  - **Placeholder**: назначьте общее значение для местоположения отсутствующих данных
  - **Advanced imputation**: Прогнозируйте недостающие значения из полных выборок с помощью методов машинного обучения. Например, AWS Datawig использует нейронные сети для прогнозирования отсутствующих значений табличных данных https://github.com/awslabs/datawig

SimpleImputer in sklearn

- SimpleImputer: **в sklearn для заполнения пропущенных значений-**

.fit(), .transform()

- **SimpleImputer***(missing_values=nan, strategy='mean', fill_value=None)*
  - **numerical data**:
    - Strategy = “mean”, заменить отсутствующие значения, используя среднее значение по каждому столбцу
    - Strategy = “median”, заменить отсутствующие значения с помощью медианы по каждому столбцу
  - **numerical or categorical data**:
    - Strategy = “most_frequent”, заменить отсутствующее, используя наиболее частое значение в каждом столбце
    - Strategy = “constant”, заменить отсутствующие значения на fill_value

Feature Scaling

Feature Scaling

- **Motivation**: Многие алгоритмы чувствительны к функциям, находящимся в разных масштабах, например, алгоритмы на основе метрик (KNN, K Means) и алгоритмы на основе градиентного спуска (регрессия, нейронные сети).
- Примечание: древовидные алгоритмы (деревья решений, случайные леса) не имеют этой проблемы.
- **Solution**: Привести функции к одному масштабу
  - Общие варианты (оба линейные):
    - Mean/variance стандартизация
    - MinMax масштабирование

Standardization in sklearn

- **StandardScaler**: sklearn масштабирование, значения масштабирования должны быть сосредоточены вокруг среднего 0 со стандартным отклонением 1

Transform: $x_{scaled} = \frac{x - x_{mean}}{x_{std}}$

.fit(), .transform()

```
from sklearn.preprocessing import StandardScaler
stdsc = StandardScaler()

raw_data = np.array([[-3.4], [4.5], [50], [24], [3.4], [1.6]])
scaled_data = stdsc.fit_transform(raw_data)
print(scaled_data.reshape(1,-1))
```

```
[[-0.90560498 -0.47848383  1.98151777  0.57580257 -0.53795639 -0.63527514]]
```

MinMax Scaling in sklearn

- **MinMaxScaler**: sklearn масштабирование, значения масштабирования должны быть сосредоточены вокруг среднего 0 со стандартным отклонением 1 -

.fit(), .transform()

Transform: $$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

```
from sklearn.preprocessing import MinMaxScaler
minmaxsc = MinMaxScaler()

raw_data = np.array([[-3.4], [4.5], [50], [24], [3.4], [1.6]])
scaled_data = minmaxsc.fit_transform(raw_data)
print(scaled_data.reshape(1,-1))
```

```
[[0.         0.14794007 1.         0.51310861 0.12734082 0.09363296]]
```

Pipeline (sklearn)

Pipeline in sklearn

- **Pipeline**: sklearn последовательные преобразования данных с окончательной оценкой (предотвращает утечку данных) --

Pipeline(steps, verbose=False)

```
pipeline = Pipeline([
    ('imputer', SimpleImputer(strategy='mean')),
    ('scaler', MinMaxScaler()),
    ('clf', KNeighborsClassifier(n_neighbors = 3))
    ])

pipeline.fit(X_train, y_train)
predictions = pipeline.predict(X_test)
```