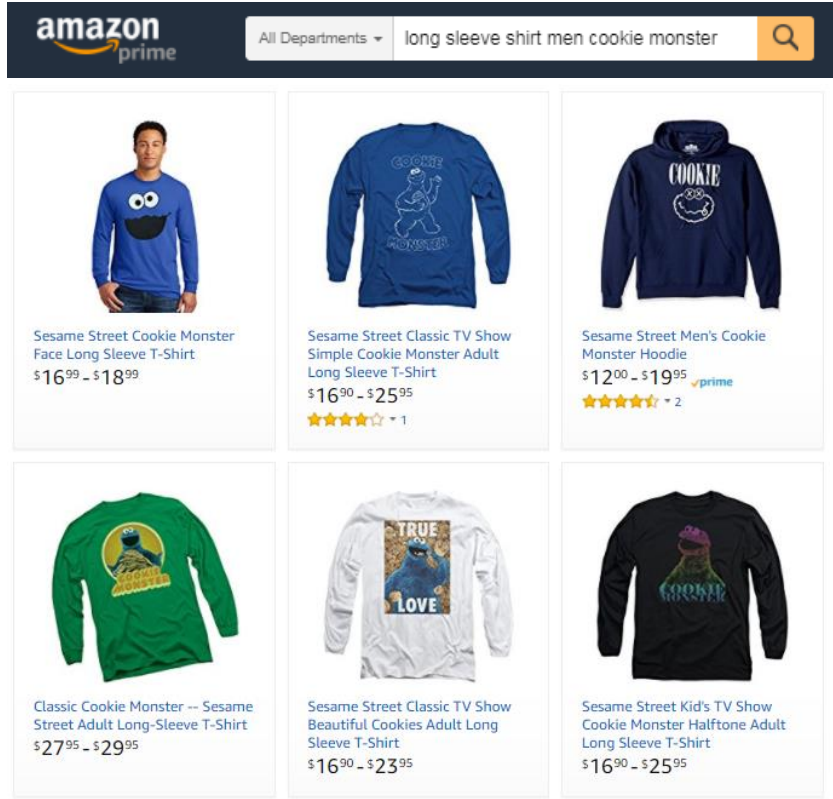




Введение в Машинное обучение

Приложения на основе машинного обучения

Machine Learning Applications

Business/ML Problem	Description	Example
Ranking	Помощь пользователям в поиске наиболее релевантной вещи	<div><h3>Ranking algorithm within Amazon Search</h3><p>The screenshot shows the Amazon Prime search results for 'long sleeve shirt men cookie monster'. The results are ranked by relevance, showing various Cookie Monster merchandise including long-sleeved shirts, hoodies, and t-shirts in different colors and designs. Each item includes a price range and a star rating.</p></div>
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	

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

Deals recommended for you [See all deals](#)



\$7.00 - \$147.90
Ends in 03:25:54



\$79.99
~~\$139.99~~
Ends in 03:25:54



\$8.99 - \$37.49
Ends in 03:20:55

\$4
End

Amazon's Choice

Amazon's Choice



Panasonic RP-HJE120-PPK In-Ear Stereo Earphones
by Panasonic

\$8.18 **prime** | FREE One-Day
Get it by **Tomorrow, Apr 24**
FREE One-Day Shipping on qualifying orders over \$35

More Buying Choices
\$7.99 (37 new offers)
[See newer model of this item *](#)

Machine Learning Applications

Business/ML Problem

Description

Example

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

Product classification for our catalog



High-Low Dress



Straight Dress



Striped Skirt



Graphic Shirt

Machine Learning Applications

Business/ML Problem

Description

Example

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

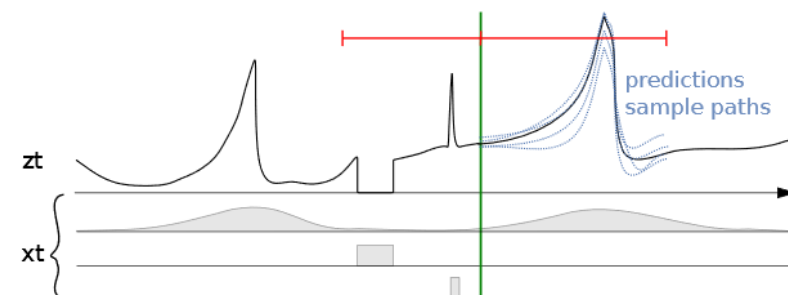
Clustering

Сгруппировать похожие объекты вместе

Anomaly Detection

Поиск необычных вещей

Predicting sales for specific ASINs



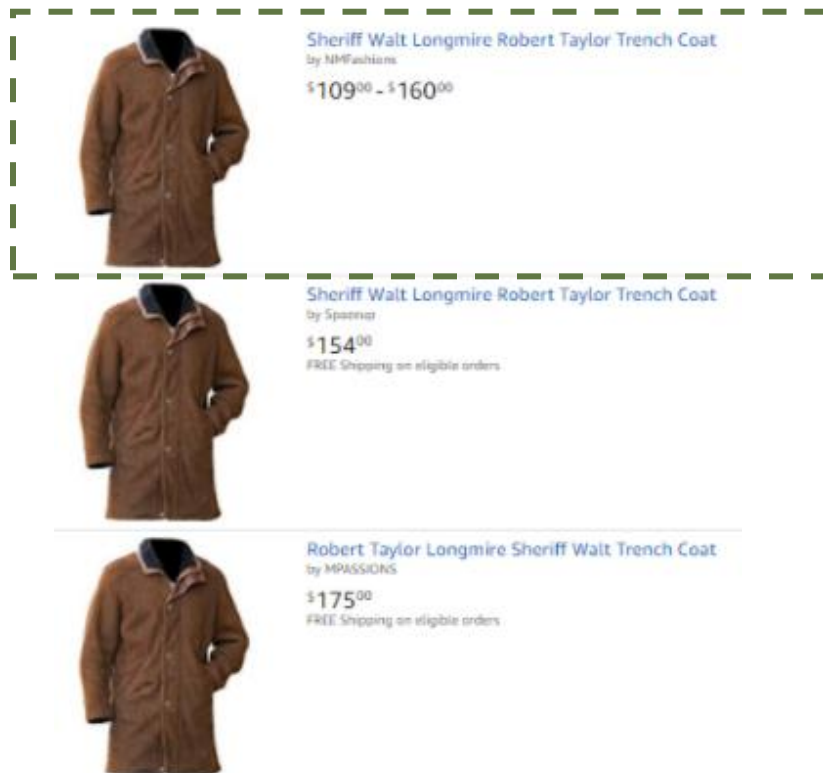
Seasonality | Out of stock | Promotions

Machine Learning Applications

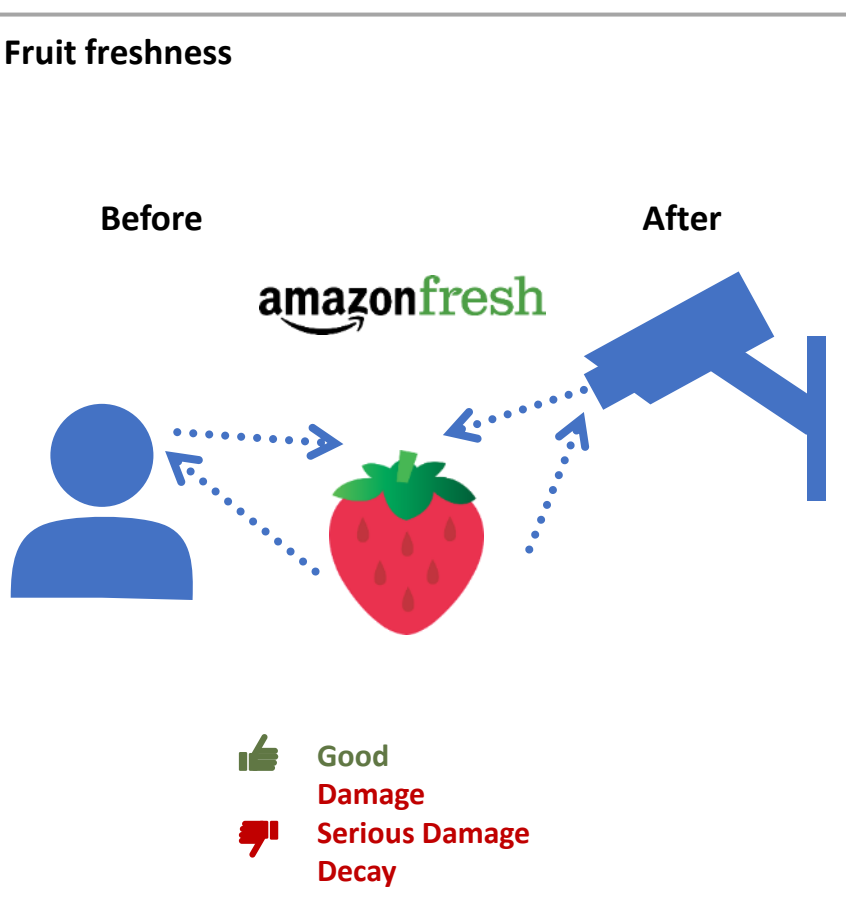
Business/ML Problem	Description
Ranking	Помощь пользователям в поиске наиболее релевантной вещи
Recommendation	Предоставление пользователям того, в чем они могут быть наиболее заинтересованы
Classification	Выяснение того, что что-то такое
Regression	Прогнозирование численного значения
Clustering	Сгруппировав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

Example

Close-matching for near-duplicates



Machine Learning Applications

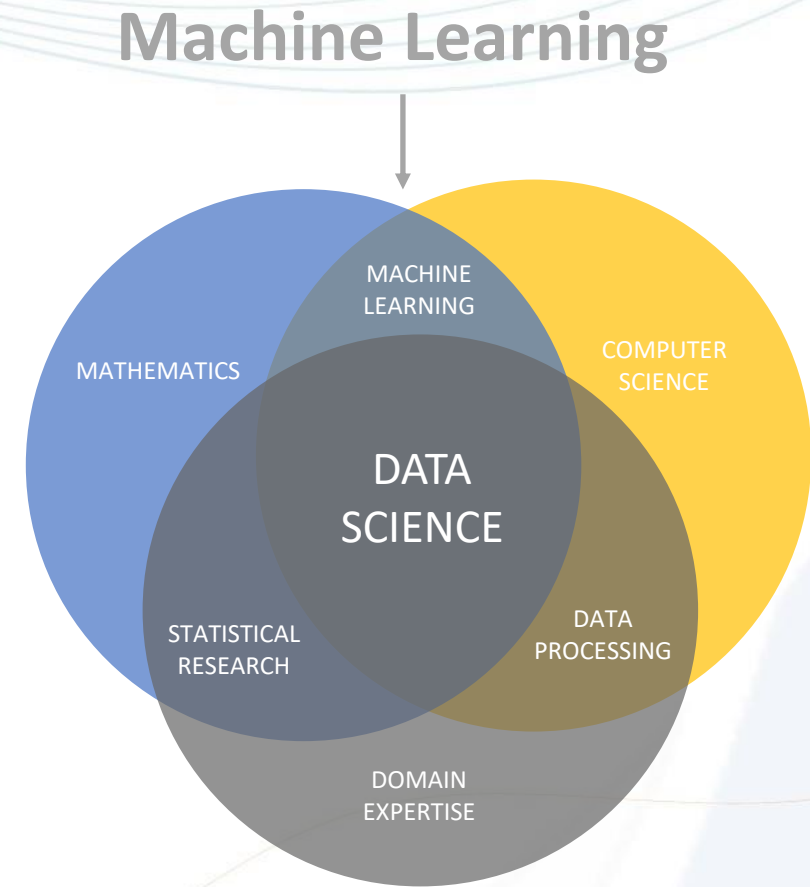
Business/ML Problem	Description	Example
Ranking	Помощь пользователям в поиске наиболее релевантной вещи	<p>Fruit freshness</p>  <p>Before</p> <p>amazonfresh</p> <p>After</p> <p>Good</p> <p>Damage</p> <p>Serious Damage</p> <p>Decay</p>
Recommendation	Предоставление пользователям то, что они могут быть 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	

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**.”

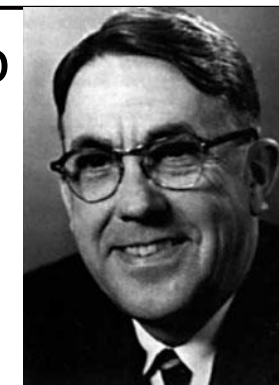


https://en.wikipedia.org/wiki/Data_science

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



Data
Rules



Classical Programming (Rules, if/else, etc.)



Ответы

Data
Answers



ML Algorithms



Trained ML Models (Rules)



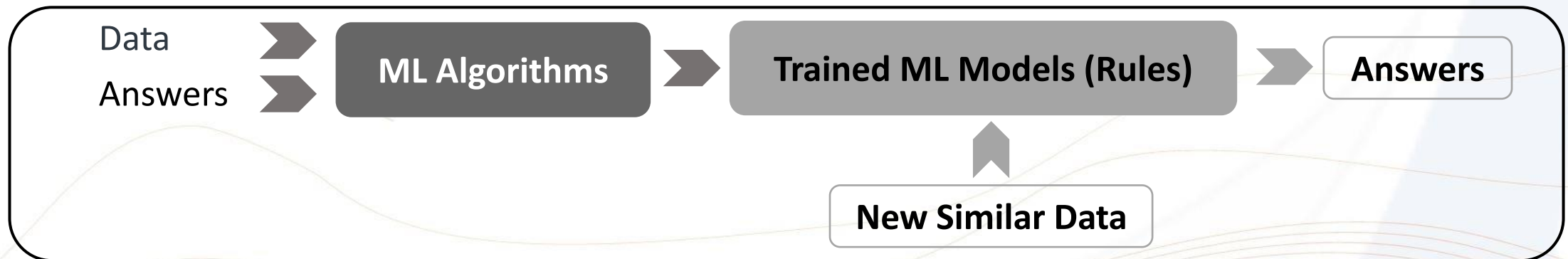
Ответы

Новые аналогичные данные

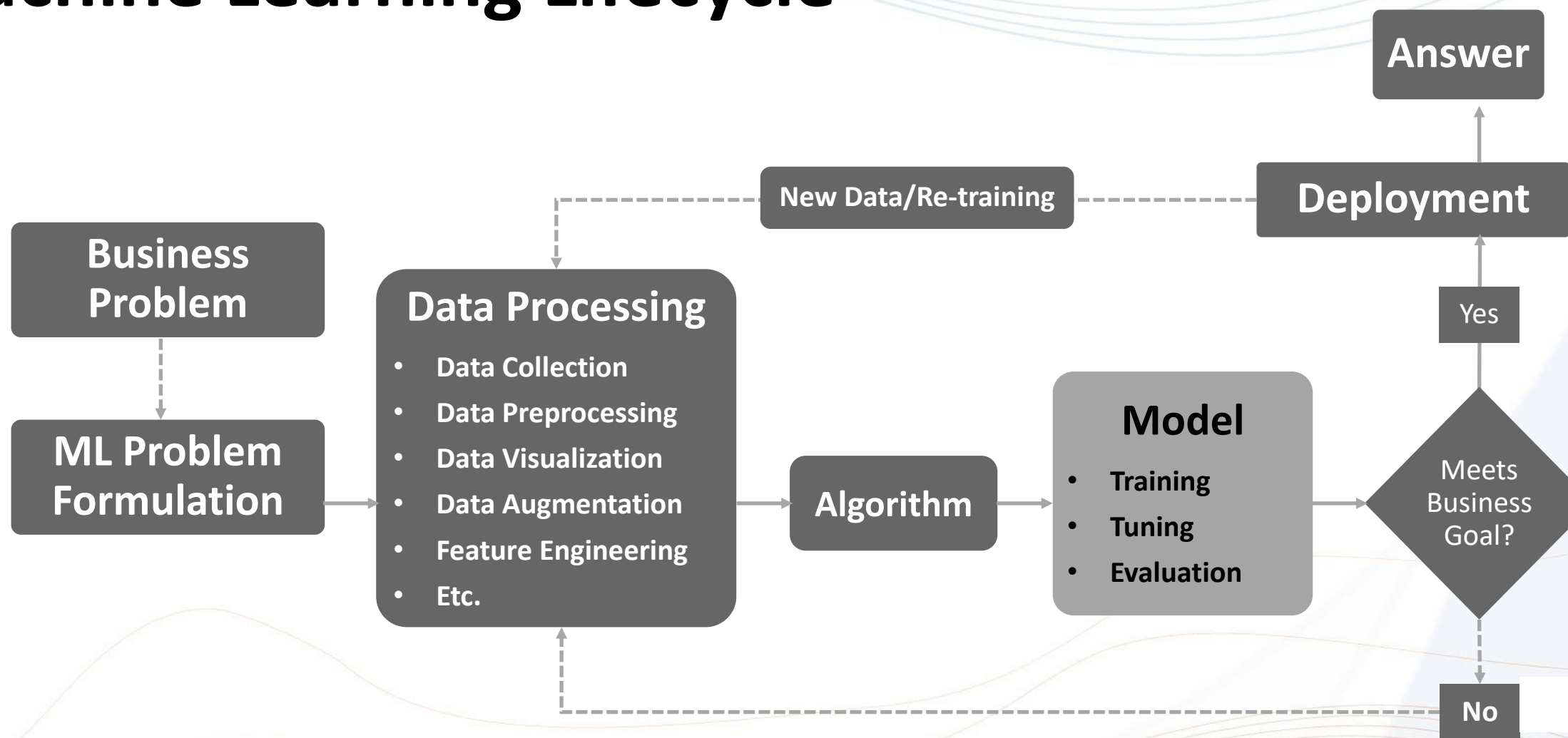


Why ML? Why now?

- **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

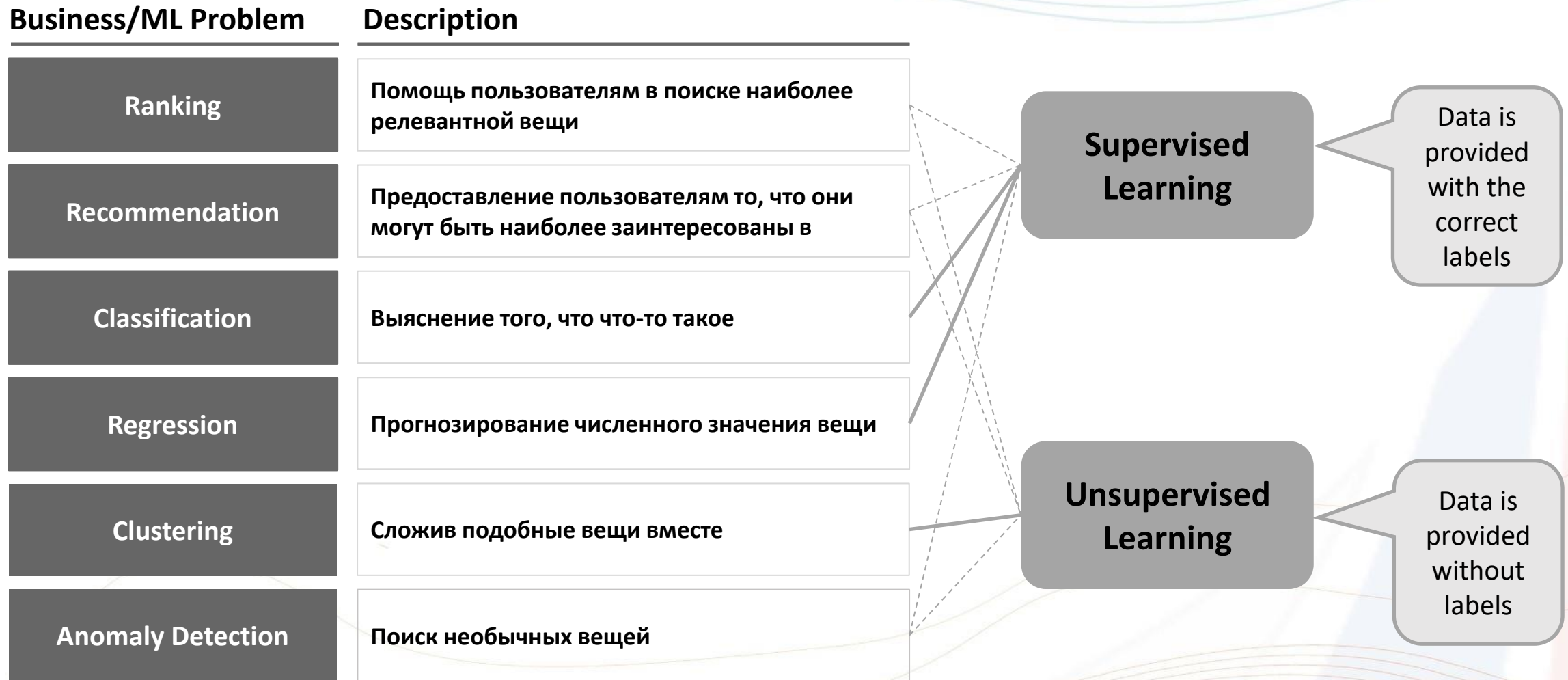


Некоторые важные понятия ML

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



Supervised vs. Unsupervised Learning

Supervised Learning

Regression
(Quantity)

Classification
(Category)

Linear

KNN

Neural Net

Trees

SVM

Logistic

Данные
предоставляют
ся с
правильными
метками

Модель
учится,
просматривая
эти примеры

Unsupervised Learning

K-Means

PCA

Collaborative
Filtering

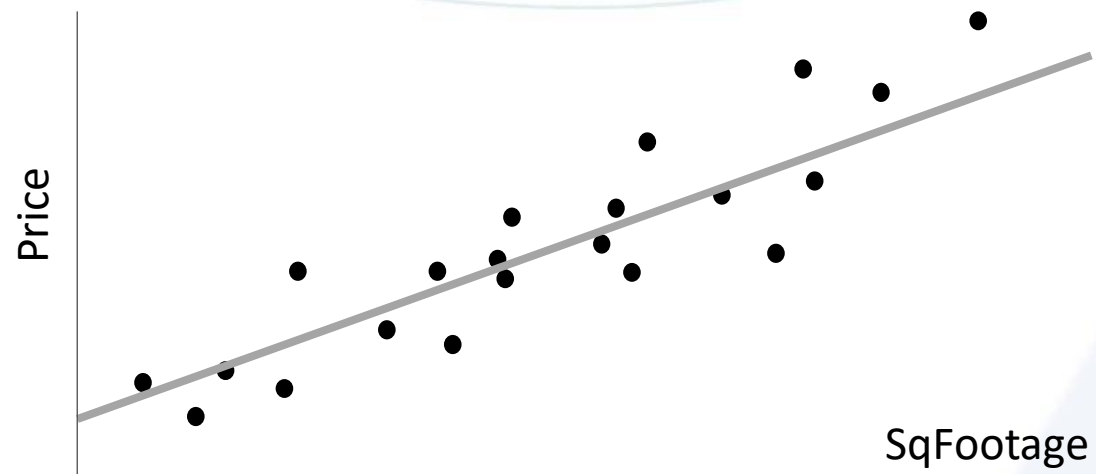
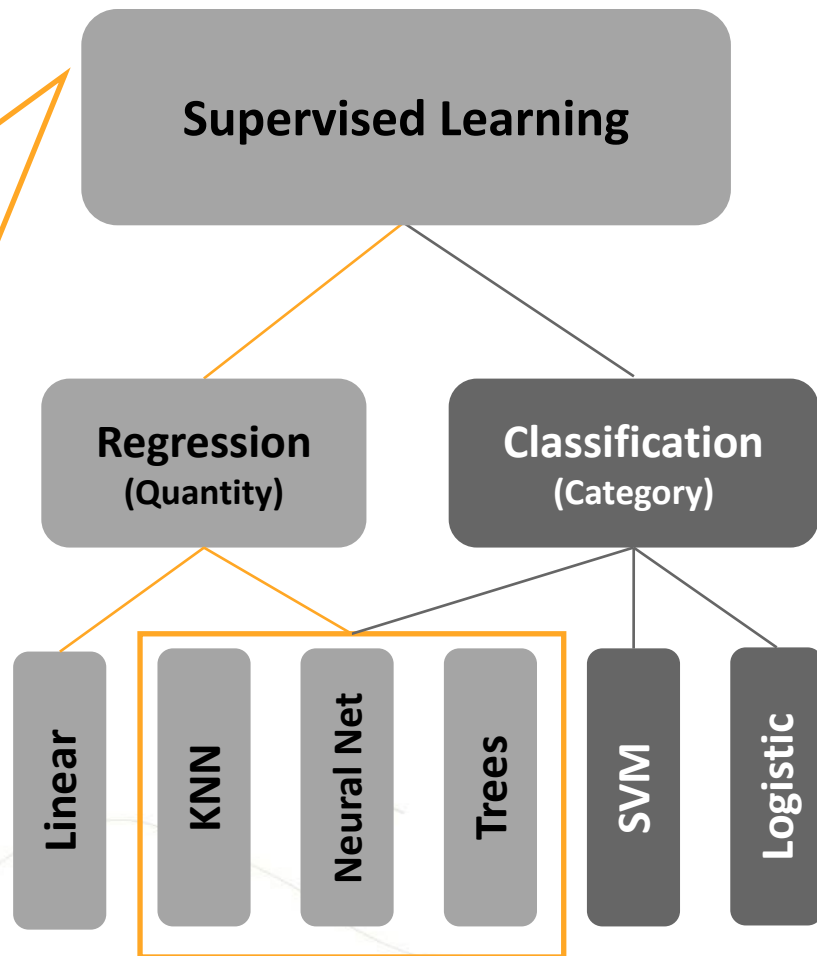
Данные
предостав
ляются без
меток

Модель
находит
закономер
ности в
данных

Supervised Learning: Regression

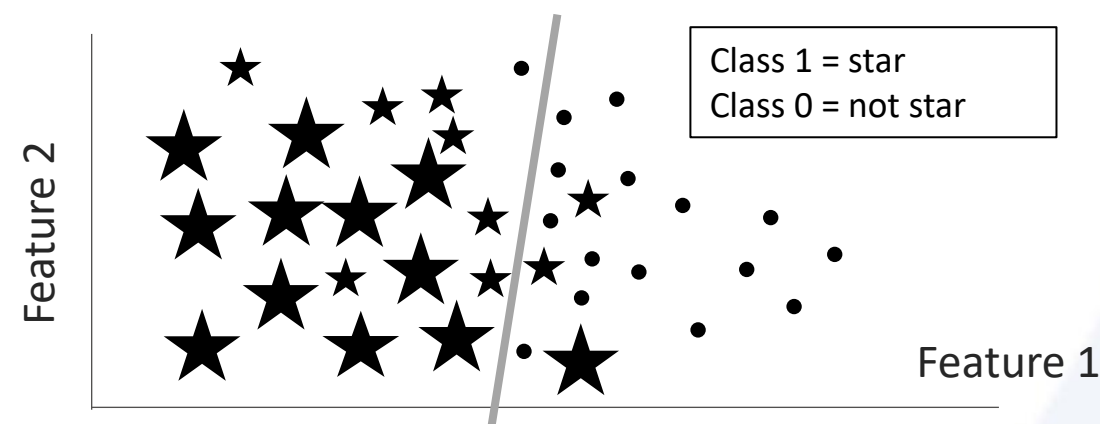
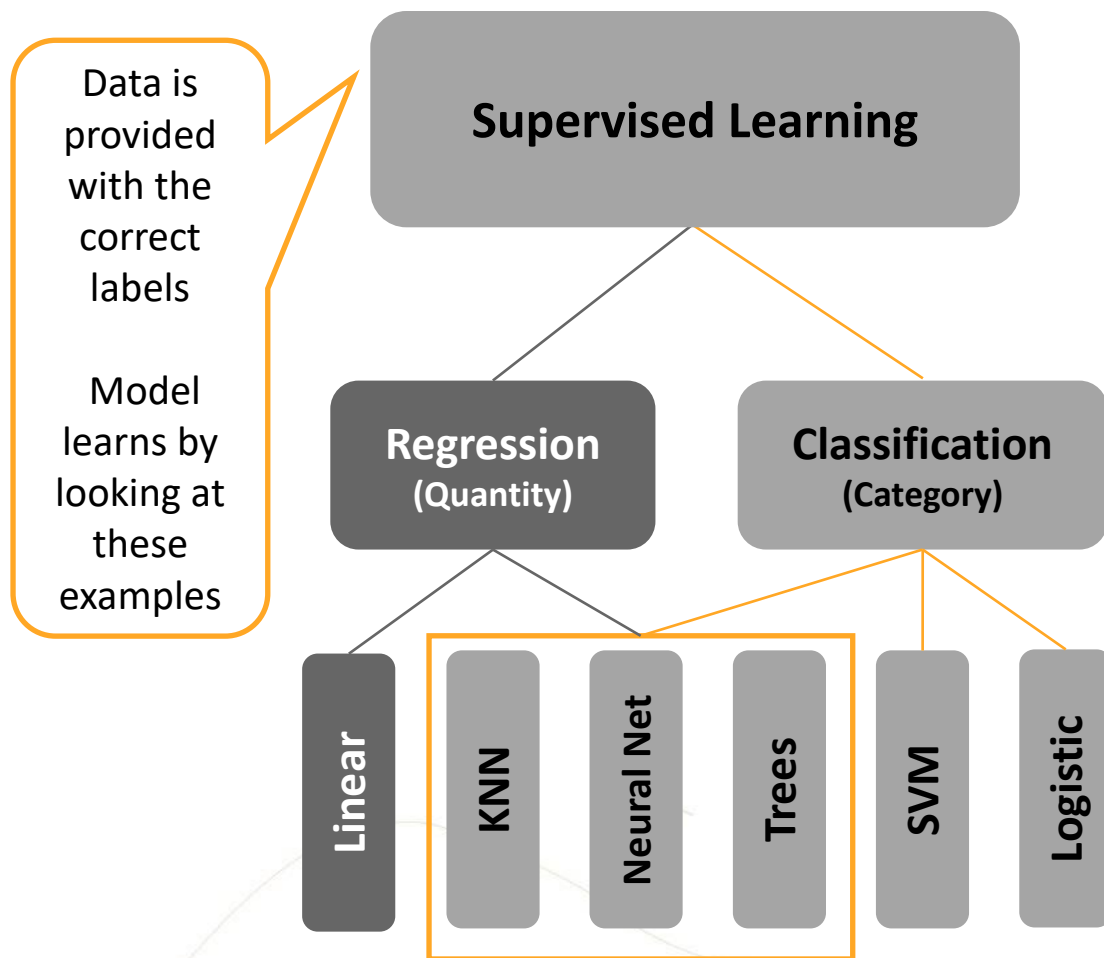
Data is provided with the correct labels

Model learns by looking at these examples



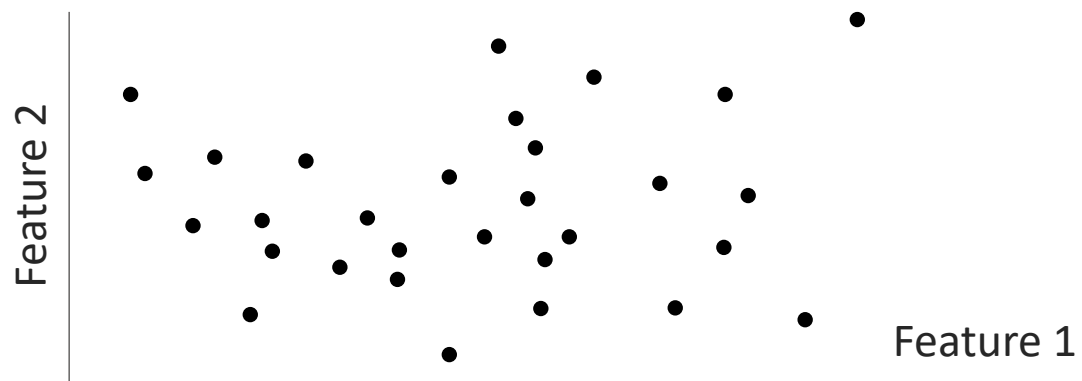
Label		Features		
Price	Bedrooms	SqFootage	Age	
280.000	3	3292	14	
210.030	2	2465	6	
...	

Supervised Learning: Classification



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

K-Means

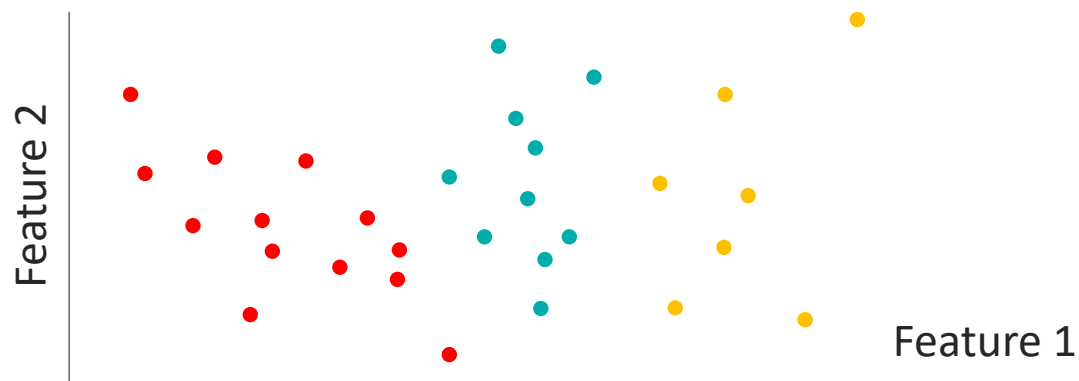
PCA

Collaborative
Filtering

Data is
provided
without
labels

Model
finds
patterns
in data

Unsupervised Learning: Clustering



Features

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

Unsupervised Learning

K-Means

PCA

Collaborative
Filtering

Data is
provided
without
labels

Model
finds
patterns
in data

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	MilesFromR estaurant 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
...

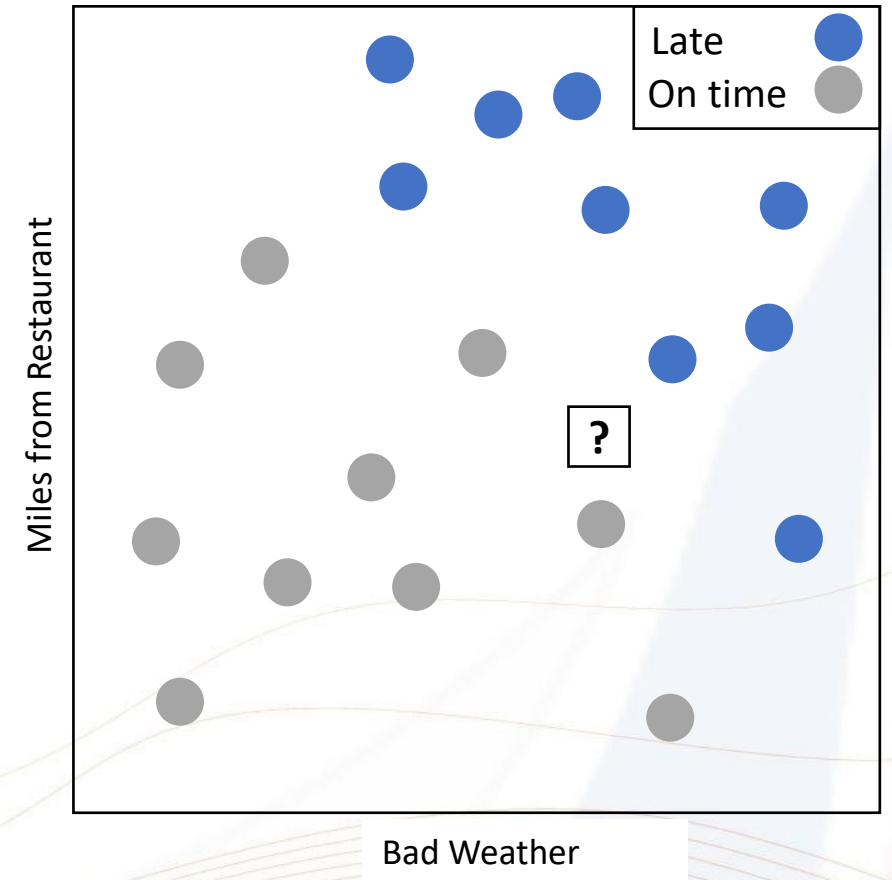
Two classes: 1/late and 0/on time



Food Delivery Problem

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

К какому классу относятся? ?



Food Delivery Problem

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

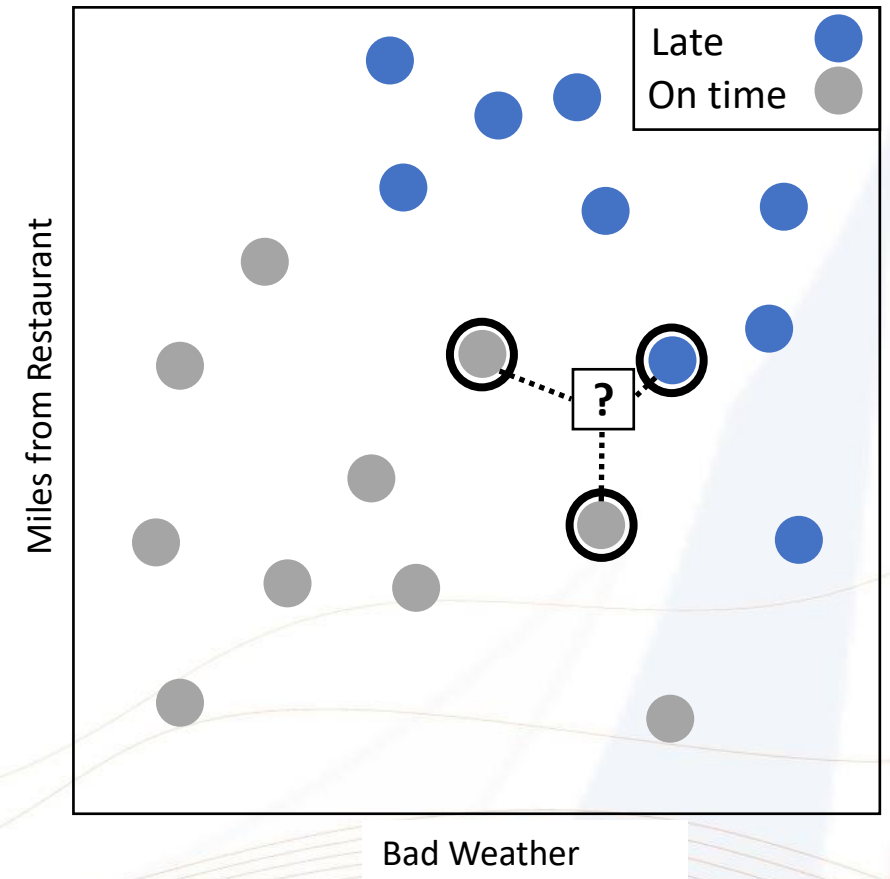
К какому классу относятся?

?

Посмотрите на ближайшие K точки данных :

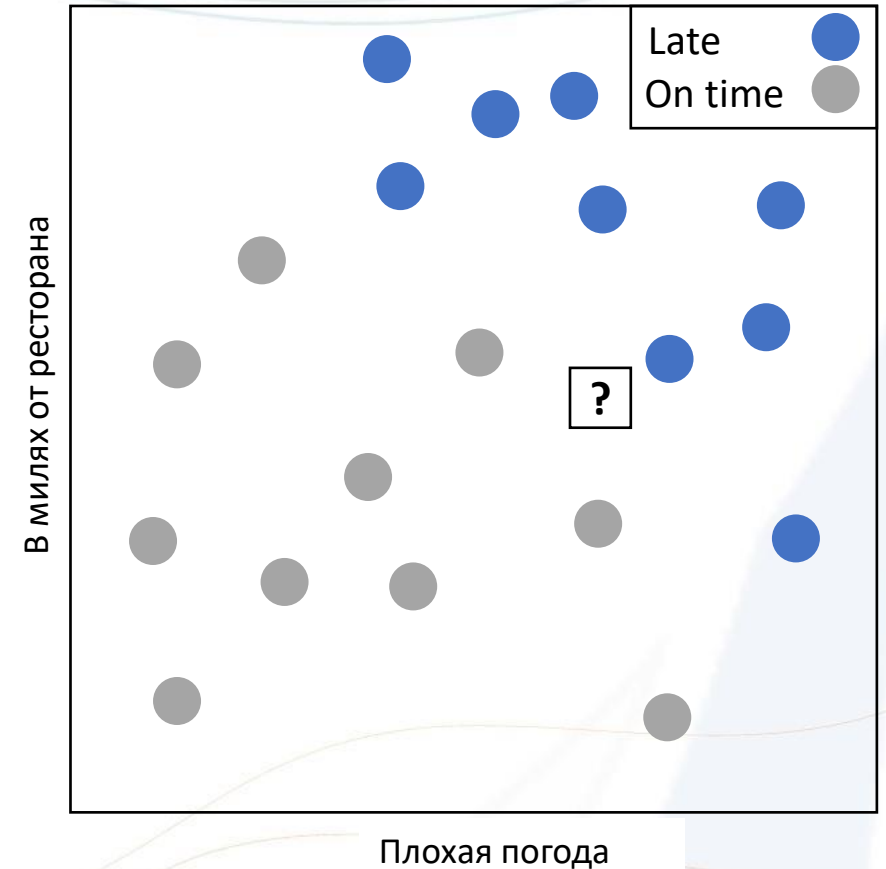
Выбрать $K = 3$

- Рассчитать расстояния от всех точек данных ?
- Найти ближайших соседей K ●
- 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.



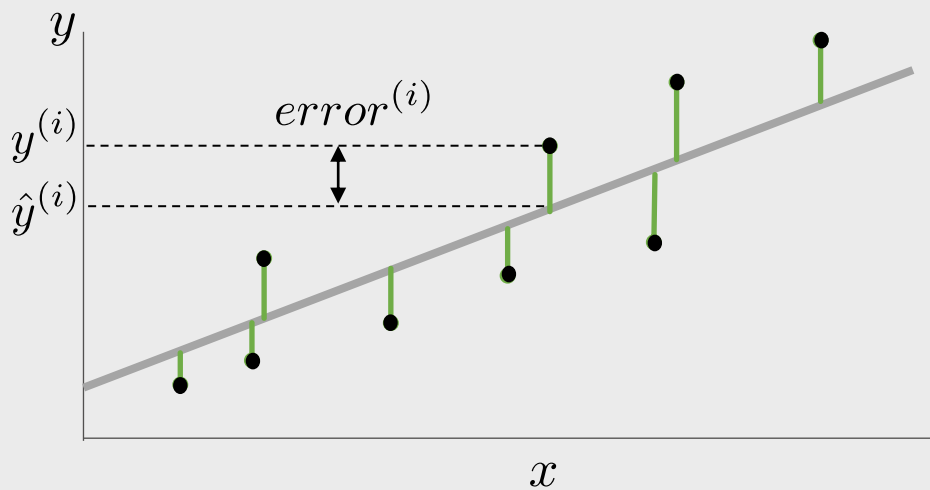
Model Evaluation

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 y^{(i)} - \hat{y}^{(i)} $
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,
- n : Number of data records

$$\bar{y} = \frac{1}{n} \sum_{i=0}^n y^{(i)}$$



Classification Metrics

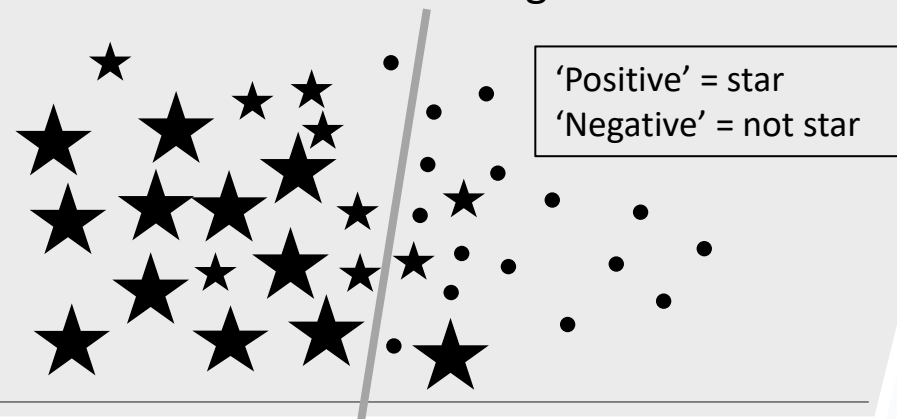
		Prediction	
		Positive	Negative
True State	Positive	True Positive 18	False Negative 3
	Negative	False Positive 1	True Negative 15

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

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

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

		Prediction	
		Positive	Negative
True State	Positive	True Positive 2	False Negative 8
	Negative	False Positive 2	True Negative 88

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

		Prediction	
		Positive	Negative
True State	Positive	True Positive 2	False Negative 8
	Negative	False Positive 2	True Negative 88

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

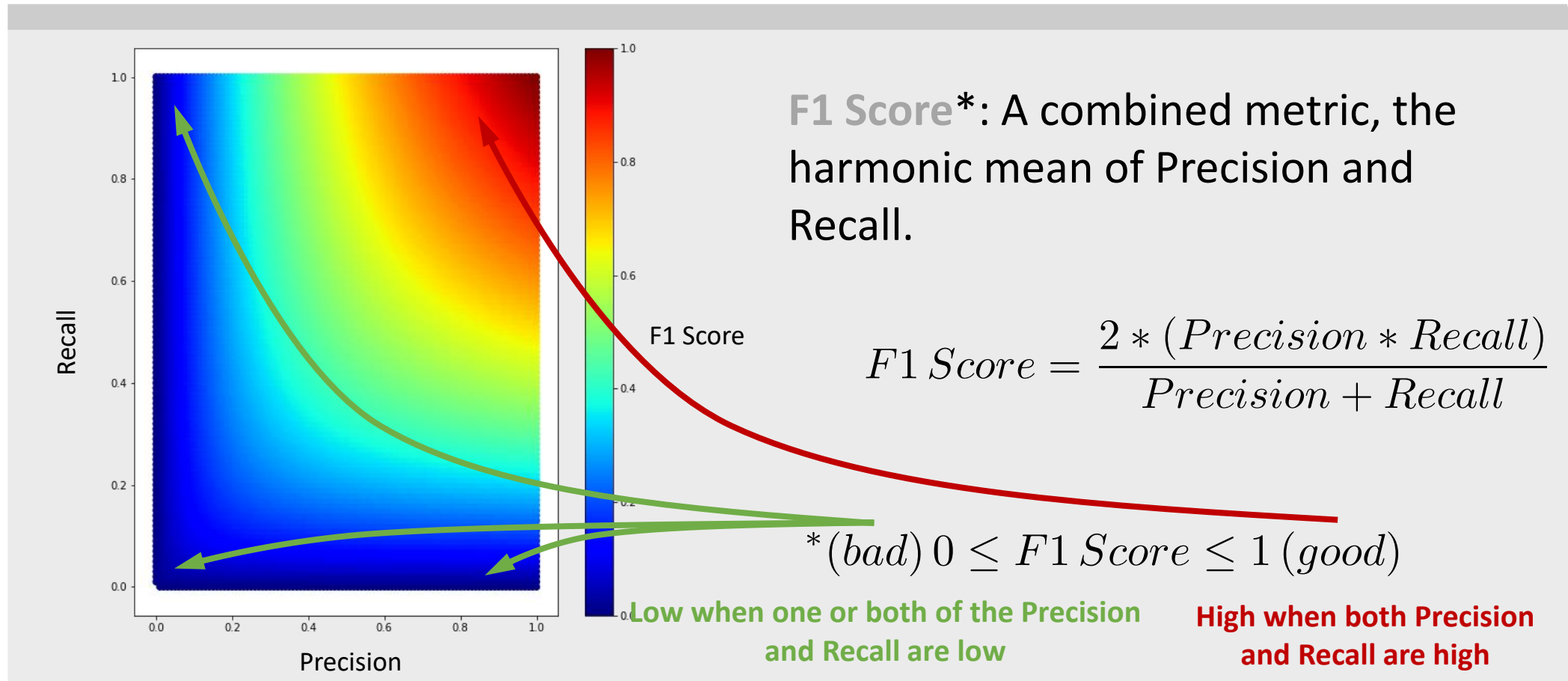
		Prediction	
		Positive	Negative
True State	Positive	True Positive 2	False Negative 8
	Negative	False Positive 2	True Negative 88

Recall*: Измеряет способность модели прогнозировать положительный результат

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

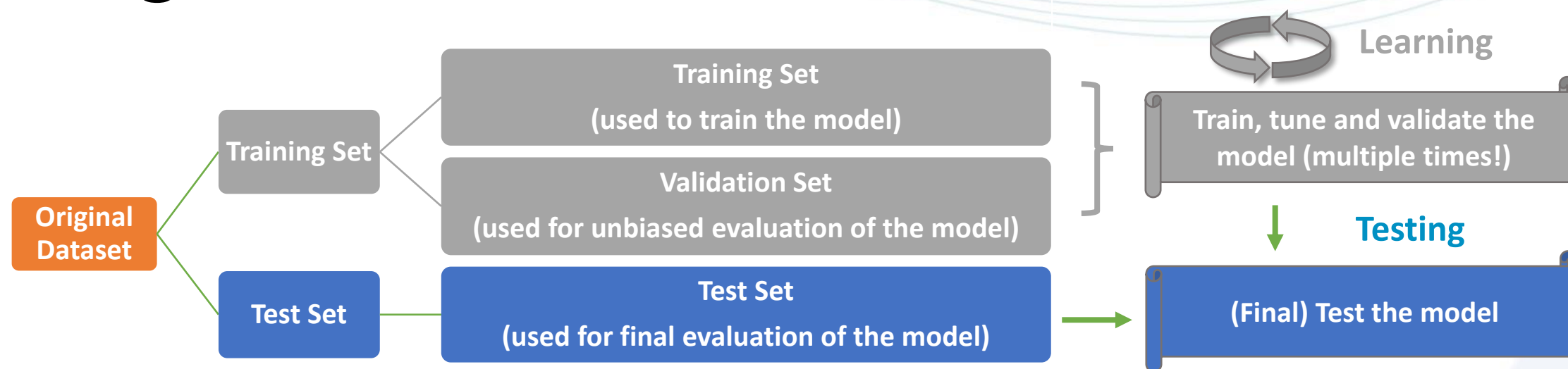
*(bad) $0 \leq Recall \leq 1$ (good)

Classification Metrics: F1 Score



**Train – Validation – Test
Datasets**

Training – Validation – Test Sets



Тестовый набор недоступен модели для обучения, он используется только обобщения (прогноза) модель на новых данных (данных, на которых она не обучалась)".

Training – Validation – Test Sets



	bad_weather	is_rush_hour	miie_distance	urban_address	iate
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

The table shows 11 data points (0-10) with 5 features: bad_weather, is_rush_hour, miie_distance, urban_address, and iate. The data is split into three sets:

- Training Set**: Rows 0-5 (6 data points)
- Validation Set**: Rows 6-8 (3 data points)
- Test Set**: Rows 9-10 (2 data points)

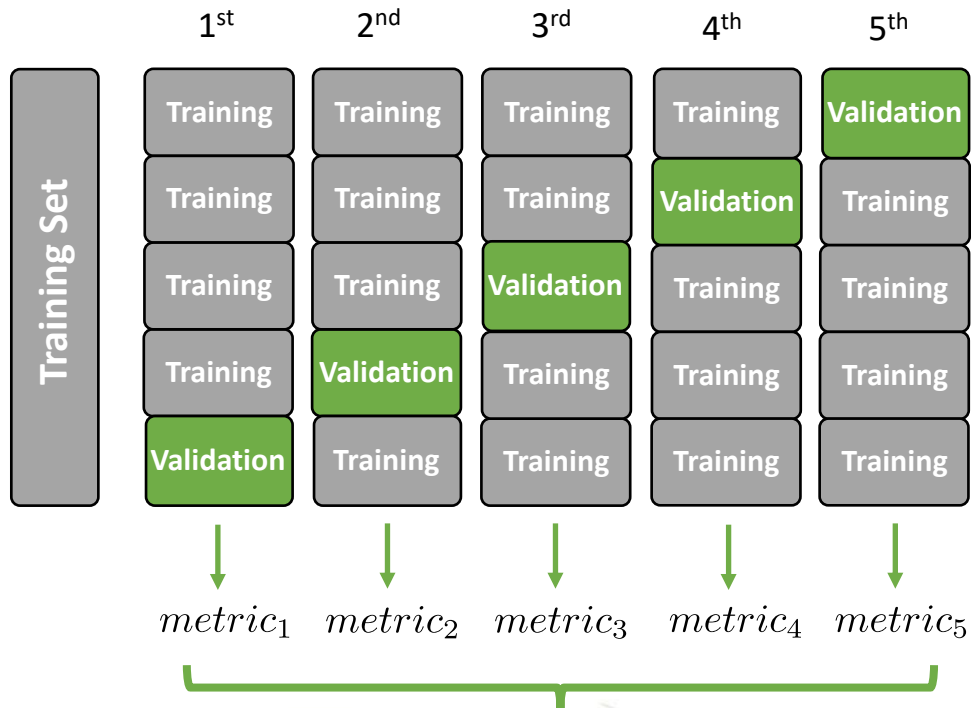
Это хорошая практика, чтобы перетасовать набор данных до разделения, чтобы избежать смещения в результате наборов.

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 для проверки модели, каждый раз обучая остальные образцы:

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



Underfitting & Overfitting

Model Evaluation: Underfitting

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



- Модель **слишком проста** для определения важных закономерностях в учебных данных.
- Модель **будет плохо работать** на обучение и проверку (и / или тест).

Model Evaluation: Overfitting

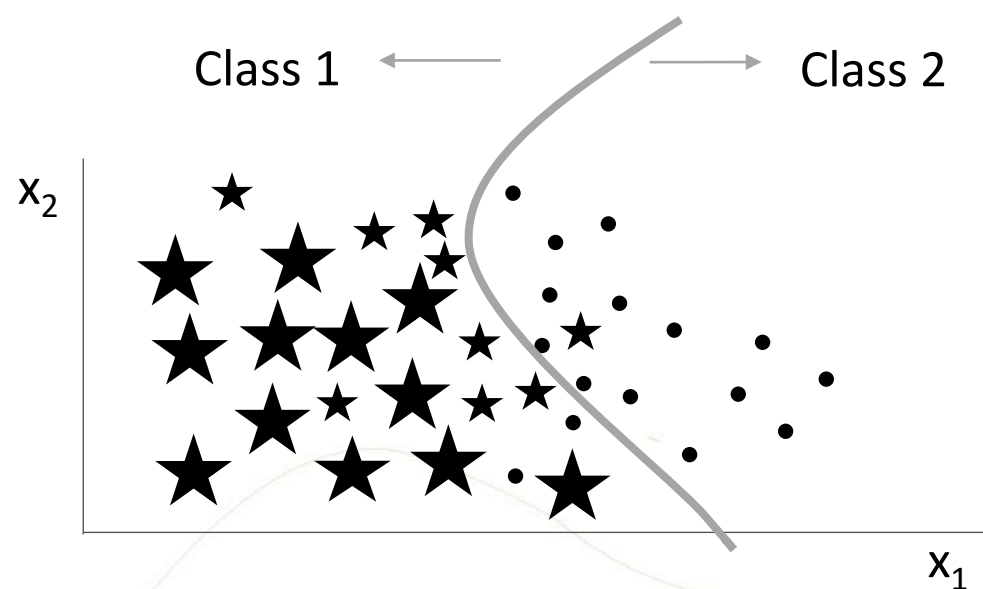
Overfitting: Модель запоминает или имитирует учебные данные и не может хорошо обобщить новые «невидимые» данные (тестовые данные).



- Модель **слишком сложна**.
- Модель подхватывает шум вместо лежащих в основе отношений.
- Модель будет хорошо работать на обучение, но плохо на проверку (и / или тест).

Model Evaluation: Good Fit

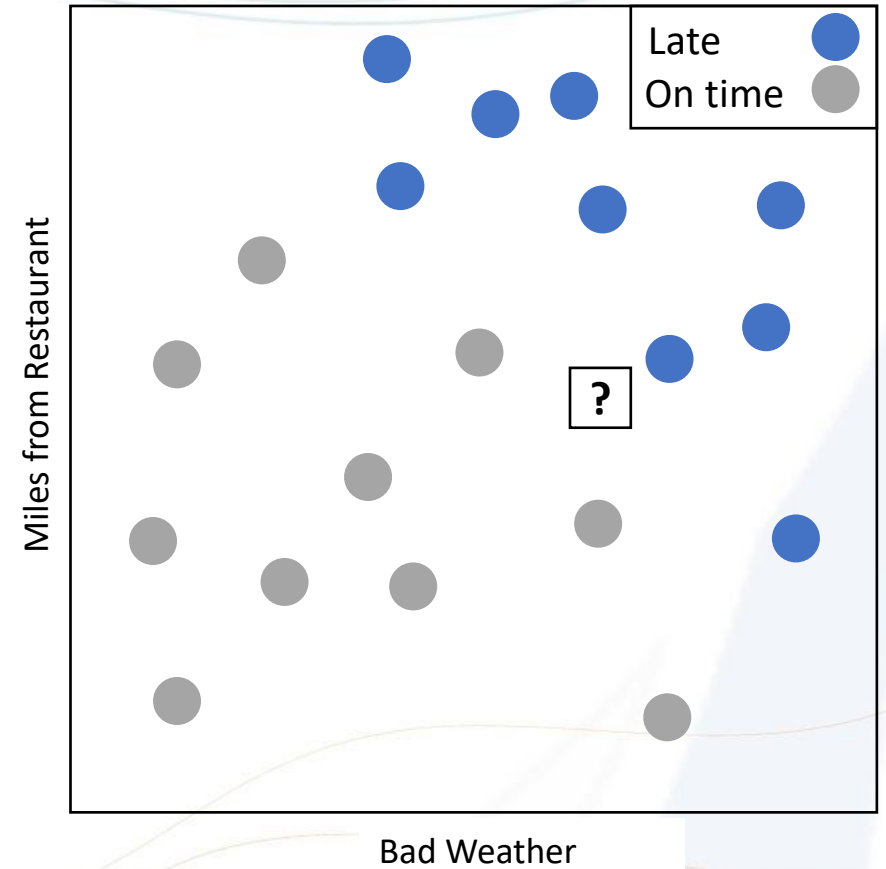
Appropriate fitting: Модель фиксирует общую взаимосвязь между входными данными (x_1, x_2) и выходом y : {Class 1, Class 2}.



- Модель не слишком простая, не слишком сложная.
- Модель находит основные отношения, а не шум в обучении.
- Модель будет достаточно хорошо работать на обучение и проверку (и / или тест).

Overfitting Hands-on

- Давайте еще раз возьмем пример доставки еды Джона.
- Мы обучаем модель K 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

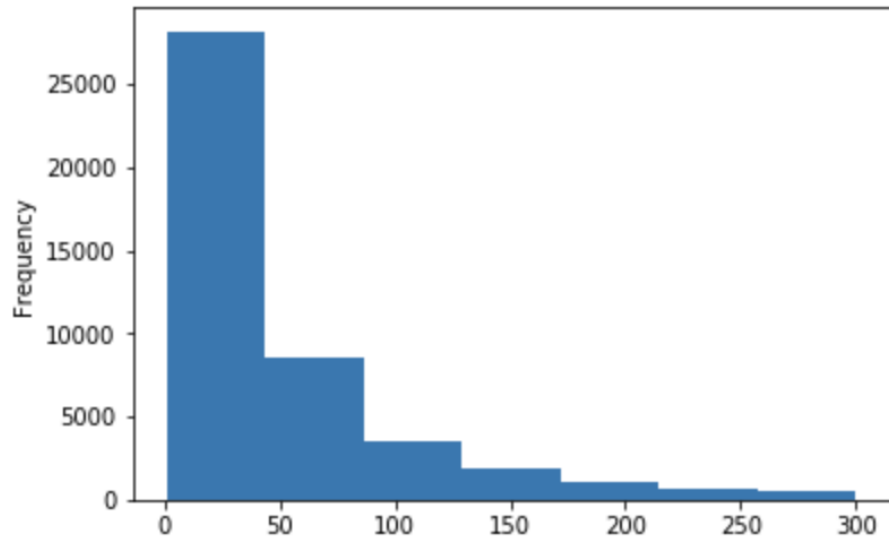
Descriptive Statistics

- **Overall statistics** `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 категориальным 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 distribution `df[target].value_counts()` or `np.bincount(y)`
- **Multivariate statistics** (more than one feature)
 - Correlation `df.plot.scatter(feature1, feature2)`, `df[[feature1, feature2]].corr()`

Univariate Statistics: Histograms

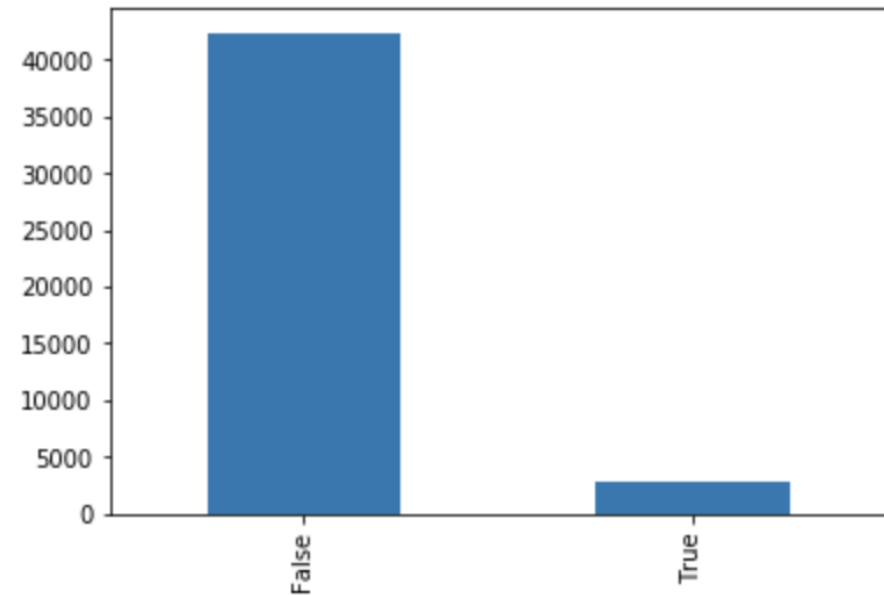
Numerical features:

```
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()
```

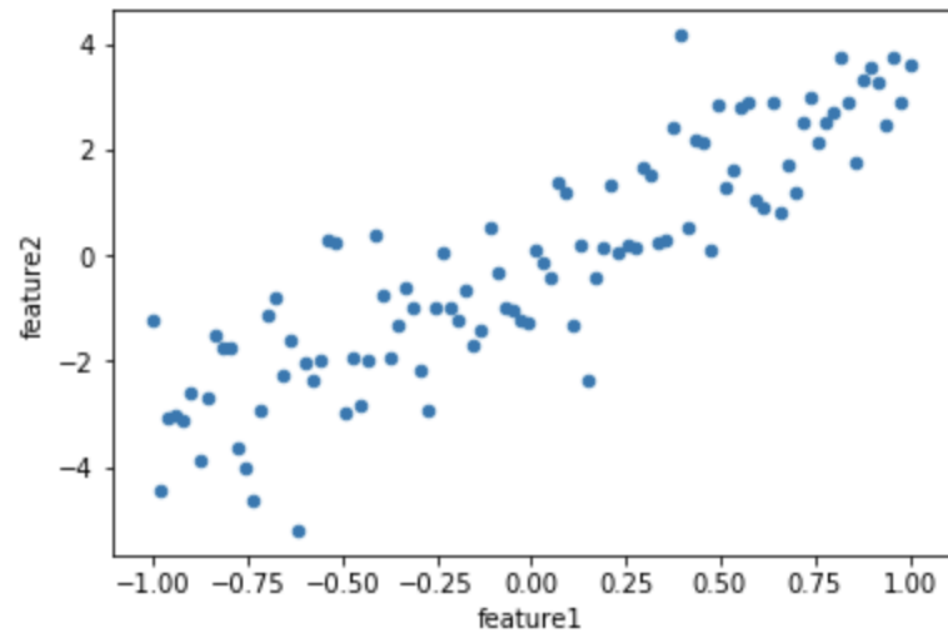
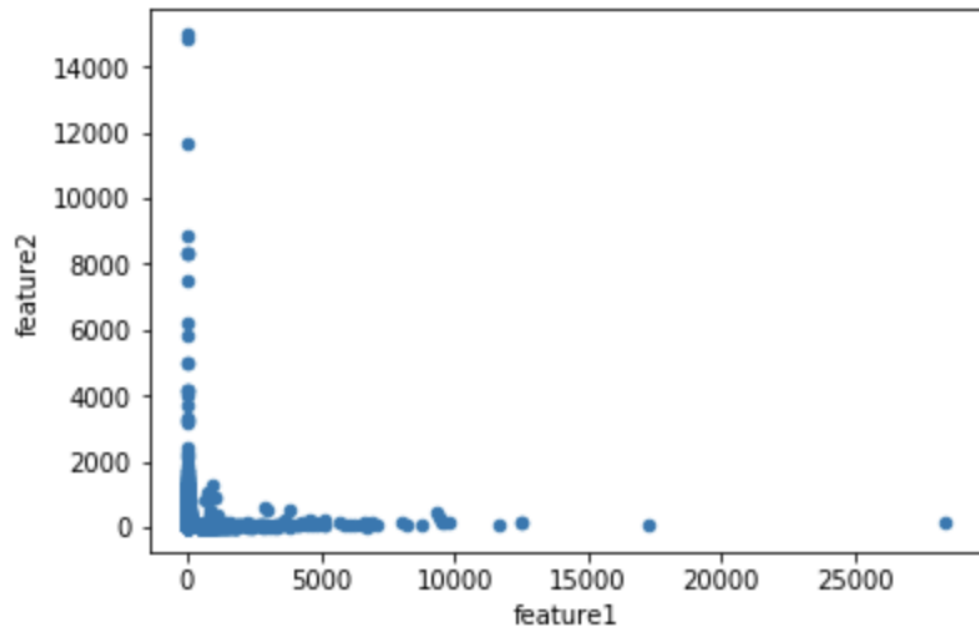


Correlations: Scatterplot

- **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

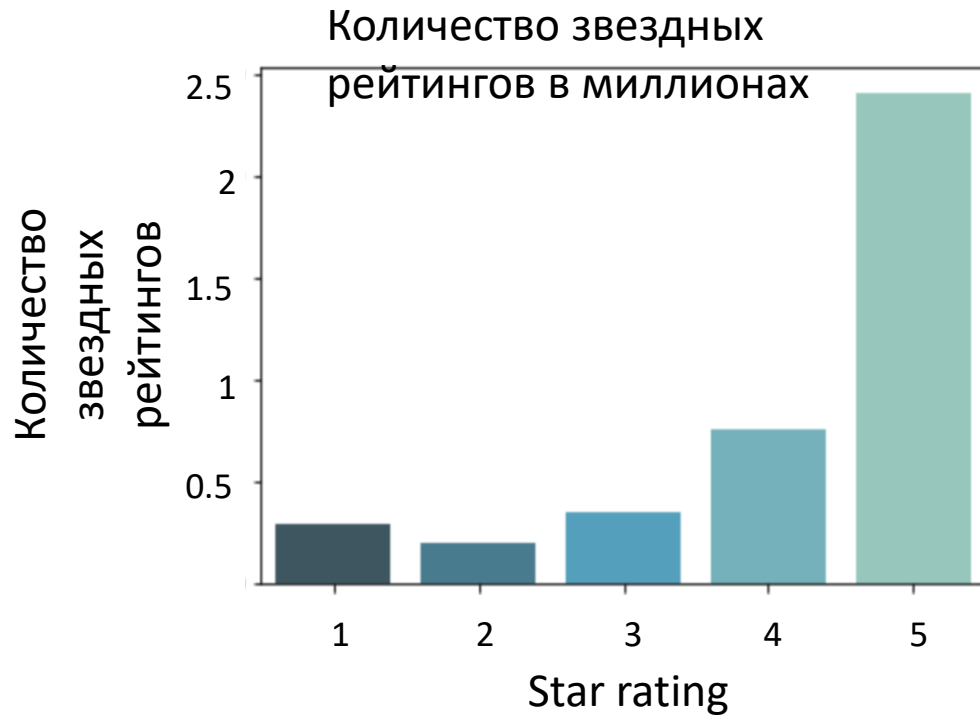
Значения **корреляции** между -1 и 1: -1 означает идеальную отрицательную корреляцию, 1 означает идеальную положительную корреляцию, а 0 означает отсутствие связи между двумя переменными.

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

Уменьшите размер доминирующего или частого класса

Up-sampling

Увеличьте размер редкого или малого класса

Data generation

Создавайте новые записи (примеры), похожие, но не идентичные.

Sample weights

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/aws-labs/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 -

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

`.fit(), .transform()`

```
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]]
```