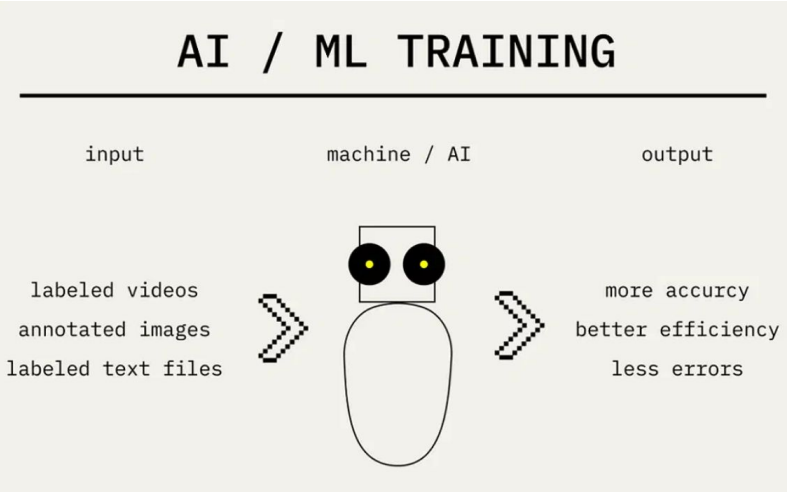
Машинное обучение на C#: ML.NET

Сазонова Ирина 23.Б10

Машинное обучение

- **Машинное обучение** (Machine learning) – это использование математических моделей данных, которые помогают компьютеру обучаться без непосредственных инструкций



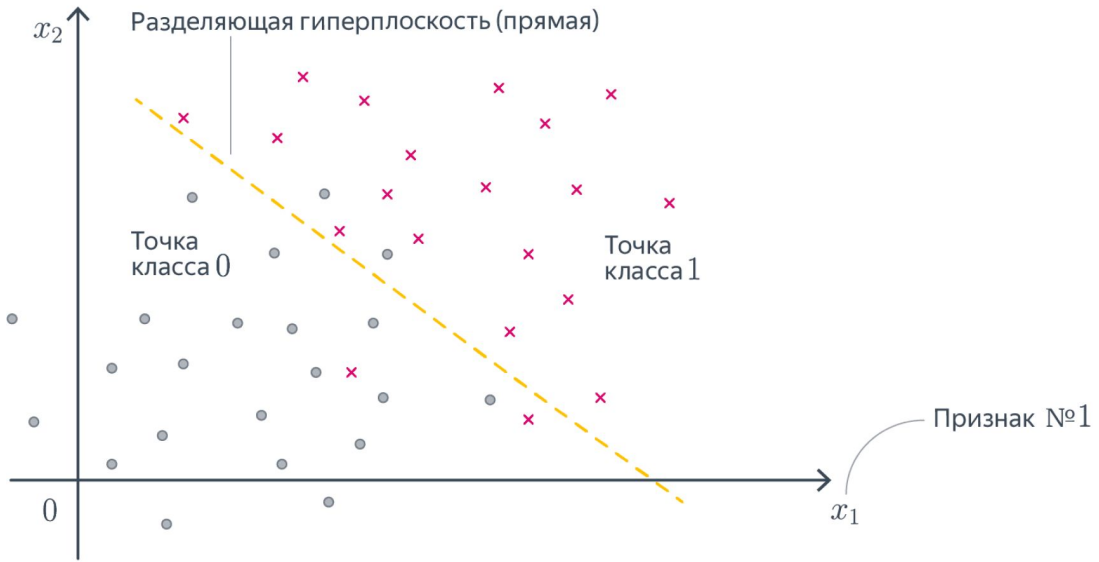
Источник

Задачи машинного обучения

1. **Регрессия** – прогноз на основе выборки объектов с различными признаками. На выходе – вещественное число
2. **Классификация** – задача, при которой нужно присвоить объекту класс: кошка, собака, машина, дома
3. **Кластеризация** – распределение данных на группы

Классификация (Classification)

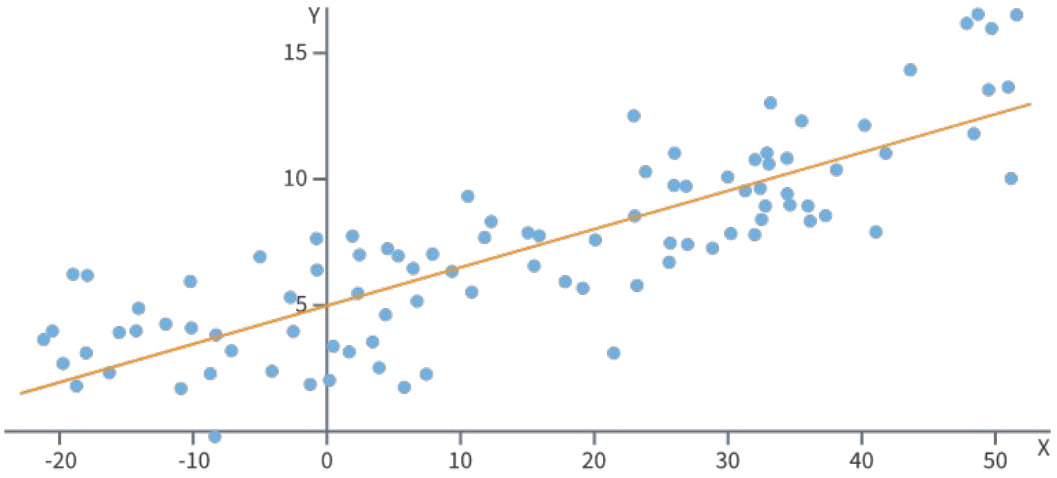
- Формально классификация производится на основе разбиения многомерного пространства признаков на области, в пределах каждой из которых многомерные векторы рассматриваются как идентичные



Источник

Регрессия (Regression)

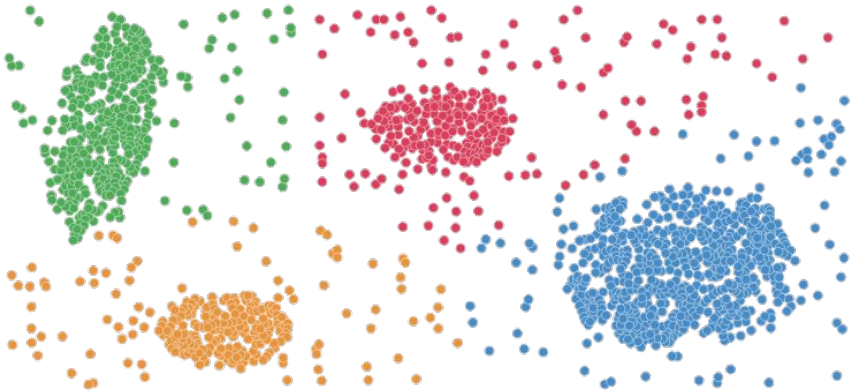
- Определяется уравнение регрессии y = ax+b и строится соответствующая прямая – линия регрессии
- a, b – параметры модели



Источник

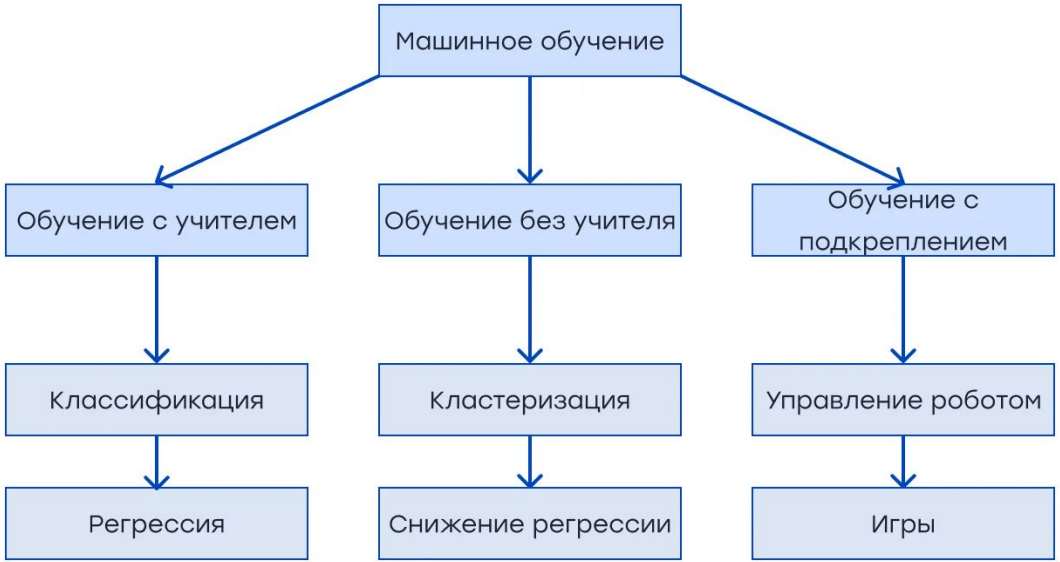
Кластеризация (Clustering)

- Это задача группировки множества объектов на подмножества (кластеры) таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров по какому-либо критерию



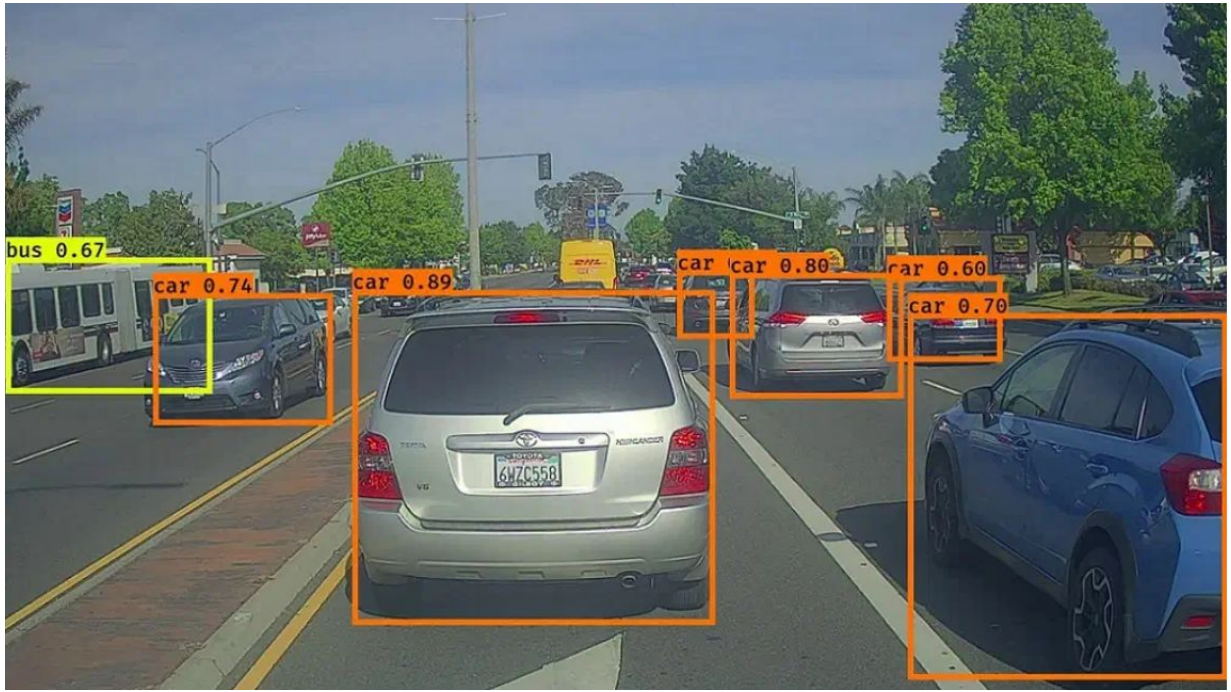
Источник

Виды машинного обучения



Источник

Обучение с учителем: разметка данных



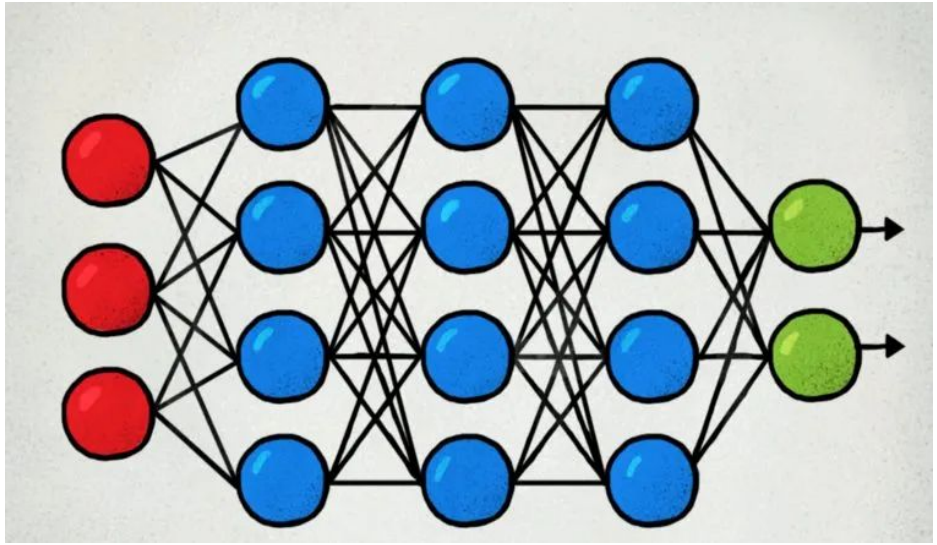
Источник

Основные алгоритмы ML

- **Линейная регрессия** – предсказывает простые линейные зависимости в данных и значения в рамках трендов
- **Логистическая регрессия** – позволяет разделять на два класса несложные объекты
- **Бустинг** – сильный классификатор создают на основе слабых: каждая новая модель учится на ошибках предыдущей
- **Байесовский классификатор** – алгоритм определяет класс, к которому принадлежит объект (расчет вероятности, с которой объект относится к тому или иному типу данных)
- **K-средних** – группирует объекты по степени похожести
- **Метод опорных векторов** – создание гиперплоскости между двумя ближайшими точками данных

Основные алгоритмы ML

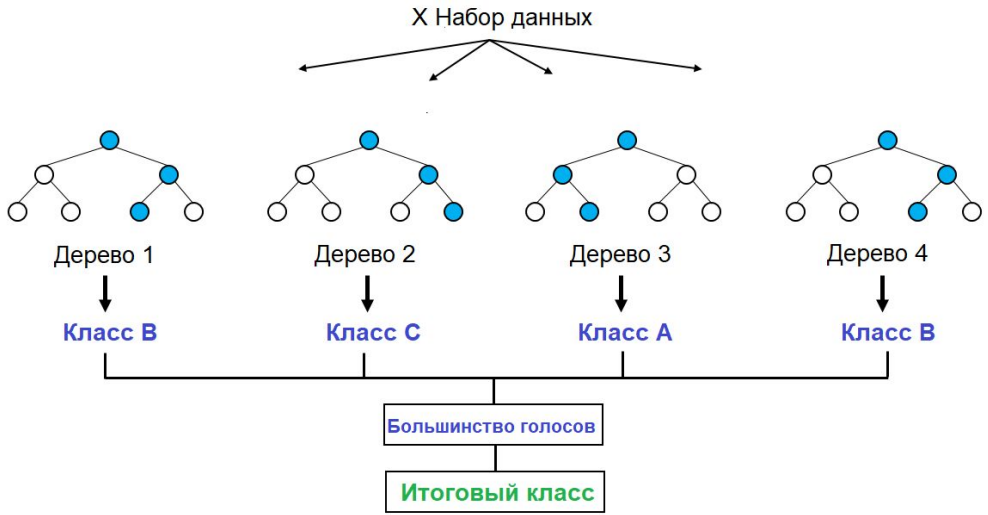
- **Нейронные сети** – “королевский алгоритм”, подходит для предсказания, распознавания и классификации
- **Сверточные нейросети** – для работы с изображениями и видео



Источник

Основные алгоритмы ML

- **Деревья принятия решений** – для классификации и предсказания состояний на основе имеющихся данных
- **Алгоритм случайного леса** – выдает самое популярное решение деревьев



Источник

Библиотека ML.NET: класс DataView

- Представляет допускающее привязку данных, настраиваемое представление DataTable для сортировки, фильтрации, поиска, изменения и навигации
- Изменения данных DataView повлияют на DataTable
- Изменения данных DataTable повлияют на все связанные с ним DataView
- Отложенное вычисление (lazy evaluation)

```
// Создание среды ML.NET.
var context = new MLContext();
```

Загрузка данных в DataView из файла

```
// Загрузка данных из CSV-файла, который содержит строку заголовка, разделитель ","
IDataView data = context.Data.LoadFromTextFile<Input>("PATH_TO_DATA_FILE", hasHeader: true,
     separatorChar: ',');

// Загрузка данных из TSV-файла, не содержащего заголовок. Разделитель по кавычкам и
удаление пробелов
var data = context.Data.LoadFromTextFile<Input>("PATH_TO_DATA_FILE", allowQuoting: true,
     trimWhitespace: true);

// Загрузка данных из нескольких CSV-файлов
var loader = context.Data.CreateTextLoader<Input>(hasHeader: true, separatorChar: ',');
var data = loader.Load("PATH1", "PATH2", "PATH3");
```

Загрузка данных в DataView: LoadFromEnumerable

```
var input = new[]
{
      new Input { Age = 30, YearsExperience = 10, ... },
      new Input { Age = 40, YearsExperience = 20, ... },
      new Input { Age = 50, YearsExperience = 30, ... }
};

var data = context.Data.LoadFromEnumerable<Input>(input);
```

Изменение данных в DataView

```
// Удаление строк с пропущенными значениями в столбцах "Age" и "YearsExperience"
var view = context.Data.FilterRowsByMissingValues(data, "Age", "YearsExperience");

// Удаление строк, в которых значение возраста меньше 20 или больше 80
var view = context.Data.FilterRowsByColumn(data, "Age", lowerBound: 20, upperBound: 80);

// Удаление столбца "Age"
var estimator = context.Transforms.DropColumns("Age");
var view = estimator.Fit(data).Transform(data);
```

Алгоритмы обучения регрессии

- LbfgsPoissonRegressionTrainer
- LightGbmRegressionTrainer
- SdcaRegressionTrainer
- OlsTrainer
- OnlineGradientDescentTrainer
- FastTreeRegressionTrainer

  dotnet add package Microsoft.ML.FastTree --version 3.0.1

- FastTreeTweedieTrainer
- FastForestRegressionTrainer
- GamRegressionTrainer

Некоторые методы

- **Data.TrainTestSplit** – разделение данных на обучающий и тестовый набор
- **Data.Transform** – преобразование данных с помощью обученной модели
- **Transform.Append** – добавление преобразований данных к конвейеру
- **Transforms.Concatenate** – объединение нескольких признаков (features) в один

Создание регрессионной модели

```
// Разделение данных на обучающий и тестовый набор с помощью метода TrainTestSplit
var trainTestData = context.Data.TrainTestSplit(data, testFraction: 0.2);
var trainData = trainTestData.TrainSet;
var testData = trainTestData.TestSet;

// Построение пайплайна, выполнение алгоритма обучения
var pipeline = context.Transforms.Concatenate("Features", "Col1", "Col2", "Col3")
    .Append(context.Regression.Trainers.FastForest());

// Обучение модели
var model = pipeline.Fit(trainData);
```

Оценка регрессионной модели

- Коэффициент детерминации R2 (от 0 до 1)
- Среднеквадратическая ошибка
- Средняя абсолютная ошибка
- Перекрестная проверка (cross-validation)

```
// Вывод коэффициента детерминации
var predictions = model.Transform(testData);
var metrics = context.Regression.Evaluate(predictions);
Console.WriteLine($"R2 score: {metrics.RSquared:0.##}");
```

Унитарная кодировка (one-hot encoding)

- В линейную модель можно подать только численные признаки, поэтому категориальный признак придётся как-то закодировать

	pet_type	color	weight
0	carrot	red	0.600000
1	cat	white	3.000000
2	hamster	brown	0.800000
3	cat	gray	5.000000
4	dog	black	7.000000

	weight	pet_type_cat	pet_type_dog	pet_type_hamster	color_brown	color_gray	color_red	color_white
0	0.600000	0	0	0	0	0	1	0
1	3.000000	1	0	0	0	0	0	1
2	0.800000	0	0	1	1	0	0	0
3	5.000000	1	0	0	0	1	0	0
4	7.000000	0	1	0	0	0	0	0

Источник

Алгоритмы обучения классификации

- AveragedPerceptronTrainer
- SdcaLogisticRegressionBinaryTrainer
- SdcaNonCalibratedBinaryTrainer
- SymbolicSgdLogisticRegressionBinaryTrainer
- LbfgsLogisticRegressionBinaryTrainer
- LightGbmBinaryTrainer
- FastTreeBinaryTrainer
- FastForestBinaryTrainer
- GamBinaryTrainer
- FieldAwareFactorizationMachineTrainer
- PriorTrainer
- LinearSvmTrainer

Модель классификации

1. Оценка: показатель F1, показатель площади под кривой, матрица несоответствия (confusion matrix)
2. Перекрестная проверка
3. Векторизация текста - метод FeaturizeText
4. Анализ тональности текста

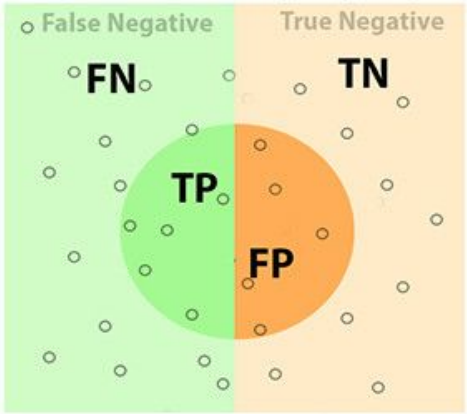
Матрица несоответствия (confusion matrix)

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

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

		алгоритм	
		0	1
истина	0	True Negative	False Positive
	1	False Negative	True Positive

=>



Источник

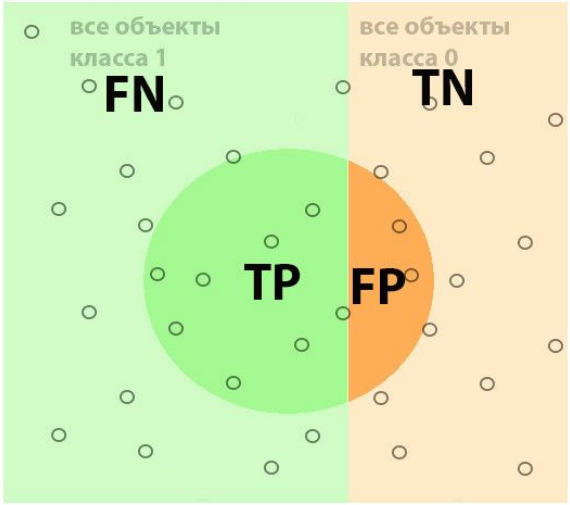
Среднегармоническая F-мера

- Частный случай при равном балансе классов: F1-мера

$$F1 = \frac{2precision * recall}{precision + recall}$$

- Среднегармоническая F-мера
  - Для нивелирования перекоса – β коэффициент

$$F_\beta = (1 + \beta^2)\frac{precision * recall}{\beta^2 * precision + recall}$$



Источник

Порядок работы с кодом

1. Сбор и загрузка обучающих данных в объект IDataView
2. Указание конвейера операций для извлечения функций и применение алгоритма машинного обучения
3. Обучение модели путем вызова функции Fit() для конвейера
4. Оценка модели и итерации для ее улучшения
5. Сохранение модели в двоичном формате для использования в приложении
6. Загрузка модели обратно в объект ITransformer
7. Прогнозирование с помощью функции CreatePredictionEngine.Predict()

Пример регрессионной модели на гитхабе

Ссылки

- Машинное обучение: что это, для чего нужно, основы и принципы
- Машинное обучение на C#: введение в ML.NET
- Введение в машинное обучение
- Что такое ML.NET и принципы работы этой системы
- Линейные модели
- Линейная регрессия (Linear regression)
- 10 наиболее распространенных алгоритмов машинного обучения
- Confusion Matrix