Проверка связи

Отправьте «+», если меня видно и слышно

Если у вас нет звука или изображения:

- перезагрузите страницу
- попробуйте зайти заново
- откройте трансляцию в другом браузере (используйте Google Chrome или Microsoft Edge)

Правила занятия

- задавать вопросы можно и нужно:
  - на занятии пишем вопросы в чат
  - после семинара — спросите в ТГ-чате с хэштегом #вопрос_ксеминару
- запись семинара будет на учебной платформе

Введение в машинное обучение. Метод ближайших соседей

Ольга Манакова
преподаватель ДПО МФТИ

Цели занятия

МФТИ

1. Обзорно рассмотрим основные методы машинного обучения, в том числе алгоритмы классификации.
2. Познакомимся с алгоритмом классификации kNN (метод ближайших соседей).
3. Вспомним основные приемы подготовки датасетов.
4. Рассмотрим: методы балансировки, кросс-валидацию, подбор гиперпараметров методом сетки.

Давайте вспомним ...

Вопрос 1: Какой из следующих методов обучения использует размеченные данные для обучения модели?
a) Обучение без учителя
b) Обучение с учителем
c) Обучение с подкреплением

Вопрос 2: Какой метод обучения используется для выявления скрытых структур в неразмеченных данных?
a) Классификация
b) Регрессия
c) Обучение без учителя

Вопрос 3: Какая из следующих метрик расстояния чаще всего используется в KNN?
a) Манхэттенское расстояние
b) Евклидово расстояние
c) Косинусное расстояние

Давайте вспомним ...

Вопрос 4: Какой из следующих элементов матрицы ошибок показывает количество правильно классифицированных положительных примеров?
a) Истинно положительные (TP)
b) Ложно положительные (FP)
c) Ложно отрицательные (FN)

Вопрос 5: Как определяется точность (Precision) в контексте бинарной классификации?
a) TP / (TP + FN)
b) TP / (TP + FP)
c) (TP + TN) / (TP + TN + FP + FN)

Вопрос 6: Как определяется полнота (Recall) в контексте бинарной классификации?
a) TP / (TP + FP)
b) TP / (TP + FN)
c) (TP + TN) / (TP + TN + FP + FN)

Введение в машинное обучение


Машинное обучение
Классическое обучение
С учителем
Классификация
Naive Bayes
K-NN
SVM
Decision Trees
Logistic Regression
Регрессия
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Без учителя
Кластеризация
DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Поиск правил
Euclat
Apriori
FP-Growth
Уменьшение размерности (обобщение)
t-SNE
PCA
LSA
SVD
LDA
Ансамблевые методы
Стекинг
Беггинг
Random Forest
Бустинг
XGBoost
AdaBoost
LightGBM
CatBoost
Обучение с подкреплением
Генетический Алгоритм
Q-Learning
SARSA
Deep Q-Network (DQN)
A3C
Нейросети и глубокое обучение
Convolutional Neural Networks (CNN)
DCNN
Перцептроны (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq

Основные виды машинного обучения

МФТИ

- **Классическое обучение**
- **Обучение с подкреплением** (модель обучается через взаимодействие со средой, получая награды за правильные действия).
- **Ансамбли** (несколько моделей объединяются для улучшения качества предсказаний. Основная идея: комбинируя слабые модели, можно получить более точную и устойчивую модель).
- **Нейросети и глубокое обучение**. **Нейронные сети** — это вычислительные модели, имитирующие структуру и функции человеческого мозга. Они состоят из узлов (нейронов), организованных в слои (входной, скрытые и выходной). **Глубокое обучение** — это подмножество машинного обучения, использующее многослойные нейронные сети (глубокие нейросети). Чем больше слоев, тем более сложные паттерны может выявлять модель.

Классическое обучение

МФТИ

Классическое обучение в машинном обучении (ML) — это подход, при котором модель обучается на размеченных данных (с учителем, Supervised Learning) или на неразмеченных данных (без учителя, Unsupervised Learning) для решения задач классификации, регрессии, кластеризации и других. Это фундамент машинного обучения, который остается актуальным для многих практических задач.

Обучение с учителем: классификация, регрессия.

Обучение без учителя: кластеризация, уменьшение размерности

В машинном обучении существует множество алгоритмов классификации, каждый из которых имеет свои особенности и применяется в зависимости от задачи и типа данных:

k-ближайших соседей (k-Nearest Neighbors, k-NN)

Наивный байесовский классификатор (Naive Bayes Classifier)

Логистическая регрессия (Logistic Regression)

Метод опорных векторов (Support Vector Machines, SVM)

Деревья решений (Decision Trees)

Случайный лес (Random Forest)

Градиентный бустинг (Gradient Boosting)

Нейронные сети (Neural Networks)

Постановка задачи классификации

МФТИ

Задача классификации состоит в определении неизвестных значений зависимой переменной и отнесении объекта к какому-либо классу, минимизируя количество ошибочных классификаций.

Целью является предсказание класса (категории) для нового наблюдения на основе обучающего набора данных.

Перейдем к рассмотрению алгоритма классификации KNN.

k-ближайших соседей (k-Nearest Neighbors, k-NN) (теория)

Основные идеи метода k-NN

МФТИ

1. **Принцип работы**:
   - Для классификации: алгоритм находит K ближайших соседей к новому объекту и принимает решение на основе классов этих соседей. Обычно используется голосование — класс, который чаще всего встречается среди K соседей, становится предсказанным классом.
   - Для регрессии: предсказание значения для нового объекта основывается на среднем значении целевых переменных K ближайших соседей.
2. **Выбор K**:
   - Значение K влияет на качество модели. Малое значение K может привести к переобучению, в то время как слишком большое значение может привести к недообучению. Часто K выбирается с помощью кросс-валидации.
3. **Метрика расстояния**:
   - Обычно используется евклидово расстояние, но также могут применяться другие метрики, такие как манхэттенское расстояние, Чебышевское расстояние или косинусное сходство, в зависимости от задачи и типа данных.
4. **Нормализация данных**:
   - Важно нормализовать данные перед применением KNN, особенно если разные признаки имеют разные масштабы. Это поможет избежать ситуации, когда один признак доминирует над другими.

Основные идеи метода k-NN

МФТИ

5. **Преимущества и недостатки**:
   - **Преимущества**:
     Простота в реализации и интерпретации.
     Хорошо работает на небольших датасетах.
     Не требует предположений о распределении данных.
   - **Недостатки**:
     Производительность может ухудшаться на больших датасетах, так как требует вычисления расстояний до всех обучающих образцов.
     Чувствительность к шуму и выбросам.

Применение KNN:

- Классификация изображений.
- Рекомендательные системы.
- Анализ текстов.

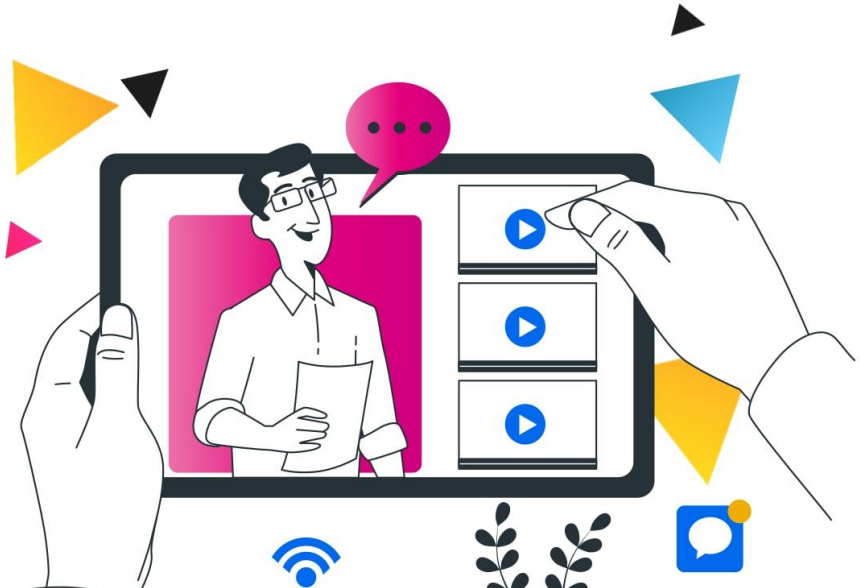
k-ближайших соседей (k-NN)

МФТИ

В подмодуле “sklearn.neighbors” библиотеки “scikit-learn” представлены несколько основных алгоритмов, которые используются для задач классификации, регрессии и поиска ближайших соседей:

1. K-ближайших соседей (K-Nearest Neighbors, KNN) - алгоритм, который предсказывает класс объекта на основе классов его K ближайших соседей:
    1. **KNeighborsClassifier** - алгоритм классификации, который предсказывает класс объекта на основе классов его K ближайших соседей.
    2. KNeighborsRegressor - алгоритм регрессии, который предсказывает значение на основе средних значений целевой переменной K ближайших соседей.
2. Алгоритм ближайшего соседа (Nearest Neighbors) - алгоритм, который позволяет находить K ближайших соседей для заданного набора данных, но не производит классификацию или регрессию.
3. Алгоритм локальной взвешенной регрессии (Locally Weighted Regression) и др.

Ваши вопросы



k-ближайших соседей (k-Nearest Neighbors, k-NN) (практика)

k-ближайших соседей (практика)

МФТИ

KNN + GridSearchCV + балансировка + кросс-валидация (cross_val)

```
# базовые
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from collections import Counter
import warnings
warnings.filterwarnings('ignore')

# preprocessing
from sklearn.preprocessing import LabelEncoder # кодирование категориальныех признаков
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split # разделение на обучающую и тестовую выборки
from sklearn.model_selection import GridSearchCV #подбор гиперпараметров методом сетки

# ML classification
from sklearn.neighbors import KNeighborsClassifier # метод ближайших соседей
```

k-ближайших соседей (практика)

МФТИ

```
# metrics
from sklearn.metrics import make_scorer, f1_score, recall_score
from sklearn.metrics import (confusion_matrix,  # матрица ошибок
                             accuracy_score,    # точность
                             precision_score,   # способность определять конкретный класс
                             recall_score,      # полнота алгоритма
                             f1_score,          # средневзвешенная
                             auc,               # значение auc
                             roc_curve,         # кривая
                             roc_auc_score      # метрика
                             )

# балансировка
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import NearMiss, RandomUnderSampler


from sklearn.model_selection import cross_val_score
from imblearn.pipeline import Pipeline
```

k-ближайших соседей (практика)

МФТИ

> Загрузка данных

↳ Скрыто 7 ячеек.

> Анализ данных

↳ Скрыто 16 ячеек.

k-ближайших соседей (практика)

МФТИ

```
# 1. Определение методов балансировки
methods = {
    'NearMiss1': NearMiss(version=1),  # Используем NearMiss-1
    'NearMiss2': NearMiss(version=2),  # Используем NearMiss-2
    'NearMiss3': NearMiss(version=3),  # Используем NearMiss-3
    'RUS': RandomUnderSampler(sampling_strategy = 'majority'),
    'SMOTE': SMOTE(sampling_strategy='minority'),
}

# 2. Кросс-валидация для оценки производительности
results = {}

for method_name, method in methods.items():
  X_balance, y_balance = method.fit_resample(X, y)
  # Оценка модели с использованием кросс-валидации
  f1_scores = cross_val_score(kNN, X_balance, y_balance, cv=5, scoring=make_scorer(f1_score))
  recall_scores = cross_val_score(kNN, X_balance, y_balance, cv=5, scoring=make_scorer(recall_score))
  ROC_AUC = cross_val_score(kNN, X_balance, y_balance, cv=5, scoring=make_scorer(roc_auc_score))
  results[method_name] = [f1_scores.mean(), recall_scores.mean(), ROC_AUC.mean()]

# 3. Вывод результатов
for method_name, score in results.items():
    print(f"{method_name}: F1 Score = {score[0]:.4f},  Recall = {score[1]:.4f}, ROC-AUC = {score[2]:.4f}")
```

k-ближайших соседей (практика)

МФТИ

Подбор гиперпараметров методом сетки

```
# Создание пайплайна
pipeline = Pipeline([
    ('scaler', StandardScaler()),
    ('knn', KNeighborsClassifier())
])

param_grid = {
    'knn__n_neighbors': [3, 5, 7, 9, 11],
    'knn__metric': ['euclidean', 'manhattan']
    }

grid_search = GridSearchCV(pipeline, param_grid, cv=5, scoring='f1') # accuracy, recall, f1, precision
grid_search.fit(X_balance, y_balance)
print(f'Лучшие параметры: {grid_search.best_params_}')
print(f'Лучшая точность: {grid_search.best_score_:.4f}')
```

```
Лучшие параметры: {'knn__metric': 'manhattan', 'knn__n_neighbors': 3}
Лучшая точность: 0.9244
```

k-ближайших соседей (практика)

МФТИ

```
kNN = KNeighborsClassifier(n_neighbors=3, metric='manhattan')
```

```
kNN.fit(x_train_balance_scl, y_train)
```

```
KNeighborsClassifier
KNeighborsClassifier(metric='manhattan', n_neighbors=3)
```

```
y_pred = kNN.predict(x_test_balance_scl)
```

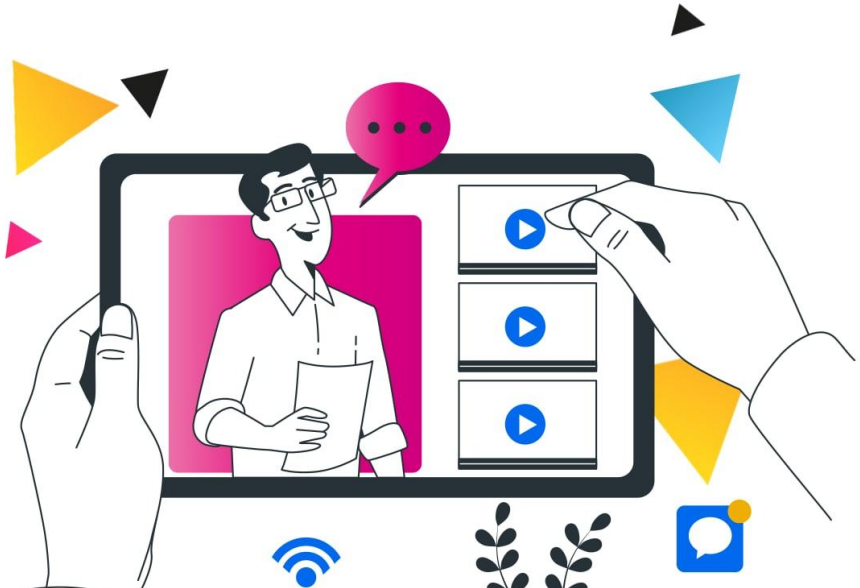
k-ближайших соседей (практика)

МФТИ

```
get_metrics(model_list, model_name_list, x_test_balance_scl, y_test)
```

```
для модели kNN
Confisuion matrix =
[[713  88]
 [ 45 754]]
Accuracy = 0.9169
Precision_score = 0.8955
Recall = 0.9437
f1 = 0.9190
AUC_ROC = 0.9169
```

Ваши вопросы



Итоги занятия

1. Изучили работу алгоритма классификации kNN на реальном датасете.
2. Рассмотрели методы балансировки, кросс-валидацию.
3. Осуществили подбор гиперпараметров методом сетки.

Дополнительные материалы по теме занятия

Интернет-источники

1. https://habr.com/ru/articles/774844/ - оценка классификации
2. https://deepmachinelearning.ru/docs/Machine-learning/Base-concepts/Supervised-learning - Обучение с учителем
3. https://scikit-learn.org/stable/api/sklearn.neighbors.html - sklearn.neighbors
4. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html#sklearn.neighbors.KNeighborsClassifier - KNeighborsClassifier
5. https://imbalanced-learn.org/stable/over_sampling.html - Over-sampling
6. https://imbalanced-learn.org/stable/under_sampling.html - Under-sampling
7. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html - GridSearchCV
8. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_val_score.html - cross_val_score
9. https://imbalanced-learn.org/stable/references/generated/imblearn.pipeline.Pipeline.html - Pipeline
10. https://archive.ics.uci.edu/ - Репозиторий машинного обучения UCI

МФТИ

Оцените по 5-балльной шкале

- насколько удовлетворены работой сегодня?
- как оцениваете свои навыки в рамках текущей темы?

МФТИ

Пожалуйста, оставьте свой отзыв о семинаре

https://forms.yandex.ru/cloud/696f6c2e90fa7b72ba8db393/

До встречи!