**Методы машинного обучения в обработке естественного языка**

**Блок 1**

1. Опишите концепцию обработки естественных языков (NLP) и перечислите области, где она активно применяется.
2. Изложите ключевые этапы и операции обработки данных в NLP.
3. Опишите важные области NLP, такие как извлечение информации (information extraction), определение тональности (sentiment analysis), анализ мнений (opinion mining) и другие.
4. Охарактеризуйте задачи генерации текста (text generation), преобразования речи в текст (speech to text conversion), вопрос-ответной системы (question-answering system) и т.д.
5. Опишите область машинного перевода в NLP, ее основные характеристики и задачи.
6. Дайте характеристику основным методам построения графов в NLP.
7. Опишите ключевые понятия контекстно-свободной грамматики и логики предикатов.
8. Опишите основные этапы в обработке текстов в NLP, создайте схему.
9. Изложите методы стемминга и лемматизации слов текста. Приведите примеры.
10. Охарактеризуйте различные методы векторизации текстов, включая Bag of Words и Tf-idf.
11. Расскажите о векторизации текстов с помощью методов word embeddings, таких как Word2Vec, Doc2Vec, FastText и т.д.
12. Опишите классификацию данных методами обучения с учителем, такими как Naïve Bayes, Logistic Regression, Support Vector Machine и другие.
13. Дайте подробное описание работы алгоритмов классификации данных: метод k-ближайших соседей, Decision Tree, Random Forest и другие.
14. Охарактеризуйте классификацию текстов алгоритмами XgBoost и CatBoost.
15. Опишите подробно метод K-средних обучения без учителя. Приведите примеры.
16. Представьте метод иерархической кластеризации. Представьте его особенности и нарисуйте схему.
17. Опишите методы выборки признаков, такие как Хи-квадрат, Information gain, Lasso
18. Охарактеризуйте методы балансировки классов: Random Oversampling, Random Undersampling и SMOTE.
19. Опишите выполнение оценки точности классификации и наиболее важные метрики.
20. Опишите основные способы визуального представления оценки точности классификации: круговые диаграммы, гистограммы и AUC-ROC кривые.

**Блок 2**

1. Опиши методы обработки текста, необходимые перед использованием нейронных сетей для классификации текстов.
2. Опиши методы борьбы с переобучением нейронных сетей в задачах классификации текстов.
3. Охарактеризуй методы балансировки классов: RandomOversampling, RandomUndersampling, SMOTE.
4. Опишите модели GPT (Generative Pre-trained Transformer) и их применение в анализе текстов.
5. Охарактеризуй основные отличия между моделями GPT и GPT-2 в анализе текстов.
6. Опишите основные улучшения в анализе текстов, которые были достигнуты в GPT-3 по сравнению с предыдущими версиями.
7. Опишите применение ChatGPT для автоматической генерации ответов на основе текстового анализа.
8. Дайте характеристику работе ChatGPT с узкоспециализированными текстами.
9. Опишите наиболее популярные метрики оценки эффективности работы моделей нейронных сетей в классификации текстов: accuracy, precision, recall и F1-score.
10. Опишите парсинг HTML документов с помощью библиотек BeautifulSoup и Scrapy.
11. Опишите отличия между BeautifulSoup и Scrapy в подходах к парсингу данных.
12. Охарактеризуйте применение Selenium WebDriver для извлечения данных с веб-страниц, требующих взаимодействия с JavaScript.
13. Опишите интеграцию Selenium с другими библиотеками Python для улучшения процесса парсинга данных.
14. Опишите подходы, которые используются для управления данными и их хранением в Scrapy.
15. Дайте характеристику проблемам безопасности, которые возникают при парсинге данных.
16. Опишите архитектуру Python Django и объясните, как работают основные компоненты этой платформы.
17. Опишите, как выполняется взаимодействие с базой данных в Django Python.
18. Опишите настройку статических файлов (CSS, JavaScript, изображения) в Django.
19. Опишите, как можно интегрировать асинхронную обработку задач в Django с использованием Celery
20. Опишите стратегии для масштабирования веб-приложений на Django.

**Блок 3**

1. Создайте небольшой скрипт по загрузке данных с помощью библиотеки Pandas и очистке данных от лишних символов с помощью регулярных выражений.
2. Создайте небольшой скрипт на Python с добавлением стемминга слов.
3. Создайте скрипт на Python с применением лемматизации слов.
4. Создайте скрипт использования one-hot encoding для категориальных переменных с помощью Pandas или Scikit-learn.
5. Создайте скрипт по работе с регулярными выражениями в Python для очистки текстовых данных.
6. Напишите скрипт векторизации текстовых данных с помощью метрики *Tf-idf* на Python.
7. Создайте скрипт на Pandas для загрузки нескольких данных и их слияния в единый DataFrame.
8. Создайте скрипт по предобработке текстовых данных и их разделения на train и test части.
9. Создайте скрипт масштабирования данных в библиотеке Scikit-learn.
10. Создай небольшой скрипт классификации текстов данных с помощью библиотеки Scikit-learn.
11. Напишите небольшой скрипт векторизации с помощью Bag of words на Python
12. Дан небольшой скрипт. Дополни его команда для классификации с помощью Decision Tree.

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Создание набора данных

data = {

 'headline': [

 'Новый президент избран вчера',

 'Компания X выпускает новый продукт',

 'Местный спортивный клуб победил в чемпионате',

 'Знаменитый актер получил награду',

 'Цены на нефть падают на мировом рынке',

 'Новый вирус обнаружен в Южной Америке',

 'Рекордное количество осадков зафиксировано в городе',

 'Новый фильм студии Y стал кассовым хитом',

 'Политический кризис вызывает протесты',

 'Новейший смартфон показывает впечатляющие продажи'

 ],

 'category': [

 'политика',

 'бизнес',

 'спорт',

 'культура',

 'экономика',

 'здоровье',

 'погода',

 'культура',

 'политика',

 'бизнес'

 ]

}

df = pd.DataFrame(data)

1. Дан небольшой скрипт. Дополни его команда для классификации с помощью Random Forest.

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Создание набора данных

data = {

 'headline': [

 'Новый президент избран вчера',

 'Компания X выпускает новый продукт',

 'Местный спортивный клуб победил в чемпионате',

 'Знаменитый актер получил награду',

 'Цены на нефть падают на мировом рынке',

 'Новый вирус обнаружен в Южной Америке',

 'Рекордное количество осадков зафиксировано в городе',

 'Новый фильм студии Y стал кассовым хитом',

 'Политический кризис вызывает протесты',

 'Новейший смартфон показывает впечатляющие продажи'

 ],

 'category': [

 'политика',

 'бизнес',

 'спорт',

 'культура',

 'экономика',

 'здоровье',

 'погода',

 'культура',

 'политика',

 'бизнес'

 ]

}

df = pd.DataFrame(data)

1. Дан небольшой скрипт. Дополни его команда для классификации с помощью Dense neural network

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Создание набора данных

data = {

 'headline': [

 'Новый президент избран вчера',

 'Компания X выпускает новый продукт',

 'Местный спортивный клуб победил в чемпионате',

 'Знаменитый актер получил награду',

 'Цены на нефть падают на мировом рынке',

 'Новый вирус обнаружен в Южной Америке',

 'Рекордное количество осадков зафиксировано в городе',

 'Новый фильм студии Y стал кассовым хитом',

 'Политический кризис вызывает протесты',

 'Новейший смартфон показывает впечатляющие продажи'

 ],

 'category': [

 'политика',

 'бизнес',

 'спорт',

 'культура',

 'экономика',

 'здоровье',

 'погода',

 'культура',

 'политика',

 'бизнес'

 ]

}

df = pd.DataFrame(data)

1. Дан небольшой скрипт. Дополни его команда для классификации с помощью Long short-term memory neural network

import pandas as pd

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.tree import DecisionTreeClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Создание набора данных

data = {

 'headline': [

 'Новый президент избран вчера',

 'Компания X выпускает новый продукт',

 'Местный спортивный клуб победил в чемпионате',

 'Знаменитый актер получил награду',

 'Цены на нефть падают на мировом рынке',

 'Новый вирус обнаружен в Южной Америке',

 'Рекордное количество осадков зафиксировано в городе',

 'Новый фильм студии Y стал кассовым хитом',

 'Политический кризис вызывает протесты',

 'Новейший смартфон показывает впечатляющие продажи'

 ],

 'category': [

 'политика',

 'бизнес',

 'спорт',

 'культура',

 'экономика',

 'здоровье',

 'погода',

 'культура',

 'политика',

 'бизнес'

 ]

}

df = pd.DataFrame(data)

1. **Напишите функцию на Python, использующую** BeautifulSoup**, для извлечения всех заголовков (тегов <**p**>) с веб-страницы.**
2. Объясните, как с помощью BeautifulSoup изменить атрибут href у всех тегов <a> в документе.
3. Объясните способы фильтрации элементов при поиске в BeautifulSoup.
4. **Опишите процесс использования XPath с библиотекой** lxml **для извлечения данных из HTML.**
5. **Напишите код на Python с использованием** Scrapy**, который собирает заголовки и URLы всех статей в блоге по заданной теме.**