Задания ДПП «Искусственный интеллект и машинное обучение»

Всего заданий: 15

**Задание № 1**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Обработка табличных данных с использованием Pandas.**

Необходимо загрузить датасет (CSV-файл), провести его очистку, анализ и визуализацию данных с использованием библиотеки Pandas. В ходе выполнения необходимо продемонстрировать умения по работе с табличными данными, включая фильтрацию, агрегацию, группировку и визуализацию данных.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта в области предикативной аналитики с использованием библиотек на языке программирования Python при условии постановки задачи и наличия данных»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Отсутствие каких-либо операций по очистке данных. o Неправильные или отсутствующие результаты при агрегации и группировке данных. o Отсутствие визуализации данных. • Удовлетворительно: o Данные загружены корректно. o Присутствуют базовые операции по очистке данных (удаление NaN, обработка дубликатов). o Выполнена простая агрегация данных. o Построены базовые графики (например, гистограмма). • Хорошо: o Данные загружены и очищены корректно. o Выполнены сложные операции по очистке данных (например, заполнение пропущенных значений). o Выполнена агрегация и группировка данных с пояснениями. o Построены несколько различных типов графиков, демонстрирующих различные аспекты данных. • Отлично: o Данные загружены, очищены и преобразованы без ошибок. o Выполнены все необходимые операции по очистке данных с пояснениями причин выбора методов. o Аггрегация и группировка данных выполнены с использованием сложных функций и предоставлены интерпретации результатов. o Построены информативные и эстетически оформленные графики, включающие несколько типов визуализаций. | Работа студента должна включать: 1. Загрузку данных из CSV-файла. 2. Очистку данных (удаление дубликатов, обработка пропущенных значений, преобразование типов данных). 3. Фильтрацию и выборку данных по различным критериям. 4. Аггрегацию данных (например, расчет среднего, медианы, максимума и минимума по группам). 5. Визуализацию данных (построение гистограмм, линейных графиков, scatter plots и т.д.). На входе: CSV-файл с сырыми данными. На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку Pandas и библиотеку для визуализации (например, Matplotlib или Seaborn). |

**Задание № 2**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 2: Анализ текстовых данных с использованием NLTK.**

Необходимо провести анализ текстовых данных (например, набор твитов или отзывов) с использованием библиотеки NLTK. Требуется выполнить предобработку текста, токенизацию, удаление стоп-слов, а также провести базовый анализ частотности слов.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта в области обработки естественного языка с использованием библиотек на языке программирования Python при условии постановки задачи и наличия данных»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Предобработка текста отсутствует или выполнена неправильно. o Неправильные или отсутствующие результаты анализа частотности слов. • Удовлетворительно: o Данные загружены корректно. o Проведены базовые операции предобработки текста (удаление пунктуации, приведение к нижнему регистру). o Выполнена токенизация текста и удаление стоп-слов. o Проведен базовый анализ частотности слов. • Хорошо: o Данные загружены и предобработаны корректно. o Выполнены все необходимые операции предобработки текста. o Выполнен детализированный анализ частотности слов с визуализацией. • Отлично: o Данные загружены, предобработаны и проанализированы без ошибок. o Выполнены все необходимые операции предобработки текста с пояснениями. o Проведен детализированный анализ частотности слов с пояснениями и визуализацией. | Работа студента должна включать: 1. Загрузку текстовых данных. 2. Предобработку текста (удаление пунктуации, приведение к нижнему регистру, токенизация, удаление стоп-слов). 3. Проведение анализа частотности слов. 4. Визуализацию результатов (например, WordCloud, bar plots). На входе: Текстовые данные (например, файл с отзывами или твитами). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку NLTK и библиотеку для визуализации (например, Matplotlib или Seaborn). |

**Задание № 3**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 3: Обработка изображений с использованием OpenCV.**

Необходимо загрузить изображение, выполнить его предобработку (например, изменение размера, обрезка, изменение яркости/контрастности), а затем провести базовый анализ изображений (например, детекция краев) с использованием библиотеки OpenCV.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта в области компьютерного зрения с использованием библиотек на языке программирования Python при условии постановки задачи и наличия данных»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Изображение не загружено или загружено с ошибками. o Отсутствие предобработки изображения. o Неправильные или отсутствующие результаты анализа изображения. • Удовлетворительно: o Изображение загружено корректно. o Выполнены базовые операции предобработки изображения (изменение размера, обрезка). o Проведена базовая обработка изображения (например, детекция краев). • Хорошо: o Изображение загружено и предобработано корректно. o Выполнены все необходимые операции предобработки изображения. o Проведен детализированный анализ изображения с визуализацией результатов. • Отлично: o Изображение загружено, предобработано и проанализировано без ошибок. o Выполнены все необходимые операции предобработки изображения с пояснениями. o Проведен детализированный анализ изображения с пояснениями и визуализацией результатов. | Работа студента должна включать: 1. Загрузку изображения. 2. Предобработку изображения (изменение размера, обрезка, изменение яркости/контрастности). 3. Проведение анализа изображения (например, детекция краев, фильтрация). 4. Визуализацию результатов (отображение оригинального и обработанного изображений). На входе: Изображение (например, JPEG или PNG файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку OpenCV и библиотеку для визуализации (например, Matplotlib). |

**Задание № 4**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 4: Визуализация данных с использованием Matplotlib и Seaborn.**

Необходимо провести визуализацию предоставленных данных с использованием библиотек Matplotlib и Seaborn. Требуется построить несколько различных типов графиков, проанализировать и интерпретировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь анализировать исходные данные датасета с использованием инструментов визуализации данных с целью выявления их недостатков, влияющих на обучаемость моделей искусственного интеллекта»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Графики отсутствуют или построены некорректно. o Отсутствие анализа и интерпретации результатов. • Удовлетворительно: o Данные загружены корректно. o Построены базовые графики (например, гистограмма, scatter plot). o Проведен базовый анализ и интерпретация результатов. • Хорошо: o Данные загружены и визуализированы корректно. o Построены несколько различных типов графиков. o Проведен детализированный анализ и интерпретация результатов. • Отлично: o Данные загружены, визуализированы и проанализированы без ошибок. o Построены информативные и эстетически оформленные графики. o Проведен детализированный анализ и интерпретация результатов с пояснениями. | Работа студента должна включать: 1. Загрузку данных. 2. Построение различных типов графиков (гистограммы, линейные графики, scatter plots, box plots и т.д.). 3. Анализ и интерпретация результатов визуализации. На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеки Matplotlib и Seaborn. |

**Задание № 5**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 5: Реализация и оценка простого классификатора.**

Используя библиотеку scikit-learn, необходимо реализовать и обучить модель классификации (например, логистическая регрессия или дерево решений) на предоставленном наборе данных. Требуется выполнить предобработку данных, разбить данные на обучающую и тестовую выборки, обучить модель, провести оценку точности модели и интерпретировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта в области предикативной аналитики с использованием библиотек на языке программирования Python при условии постановки задачи и наличия данных»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Отсутствие обучения модели или обучение выполнено некорректно. o Отсутствие оценки точности модели. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели. • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели. • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели с визуализацией результатов. | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Разбиение данных на обучающую и тестовую выборки. 3. Построение и компиляцию модели классификации. 4. Обучение модели на тренировочных данных. 5. Оценку точности модели на тестовых данных. 6. Визуализацию результатов (например, матрица ошибок, ROC-кривая). На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку scikit-learn и библиотеку для визуализации (например, Matplotlib). |

**Задание № 6**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 6: Кластеризация данных с использованием K-means.**

Используя библиотеку scikit-learn, необходимо реализовать алгоритм кластеризации K-means на предоставленном наборе данных. Требуется выполнить предобработку данных, выбрать оптимальное количество кластеров с использованием метода локтя, провести кластеризацию, визуализировать результаты и интерпретировать полученные кластеры.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта в области предикативной аналитики с использованием библиотек на языке программирования Python при условии постановки задачи и наличия данных»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Кластеризация не выполнена или выполнена некорректно. o Отсутствие визуализации и интерпретации результатов. • Удовлетворительно: o Данные загружены корректно. o Кластеризация выполнена. o Проведена базовая визуализация и интерпретация результатов. • Хорошо: o Данные загружены и предобработаны корректно. o Определено оптимальное количество кластеров. o Кластеризация выполнена корректно. o Проведена детализированная визуализация и интерпретация результатов. • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Определено оптимальное количество кластеров с пояснениями. o Кластеризация выполнена и интерпретирована подробно. o Проведена детализированная визуализация и интерпретация результатов. | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Определение оптимального количества кластеров (метод локтя). 3. Кластеризацию данных с использованием K-means. 4. Визуализацию результатов кластеризации (например, scatter plot с раскрашенными кластерами). 5. Интерпретацию полученных кластеров. На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку scikit-learn и библиотеку для визуализации (например, Matplotlib). |

**Задание № 7**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 7: Создание и обучение многослойной нейронной сети с использованием TensorFlow/Keras.**

Необходимо создать и обучить многослойную нейронную сеть для задачи классификации с использованием библиотеки TensorFlow/Keras. Требуется выполнить предобработку данных, построить модель, обучить её, провести оценку точности модели и визуализировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта в области предикативной аналитики с использованием библиотек на языке программирования Python при условии постановки задачи и наличия данных»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Отсутствие обучения модели или обучение выполнено некорректно. o Отсутствие оценки точности модели. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели. • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели. • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели с визуализацией результатов. | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Построение и компиляцию многослойной нейронной сети. 3. Обучение модели на тренировочных данных. 4. Оценку точности модели на тестовых данных. 5. Визуализацию результатов обучения (например, графики потерь и точности). На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку TensorFlow/Keras и библиотеку для визуализации (например, Matplotlib). |

**Задание № 8**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 8: Применение сверточной нейронной сети для классификации изображений.**

Необходимо создать и обучить сверточную нейронную сеть для задачи классификации изображений с использованием библиотеки TensorFlow/Keras. Требуется выполнить предобработку изображений, построить модель, обучить её, провести оценку точности модели и визуализировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта в области компьютерного зрения с использованием библиотек на языке программирования Python при условии постановки задачи и наличия данных»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Отсутствие обучения модели или обучение выполнено некорректно. o Отсутствие оценки точности модели. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели. • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели. • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели с визуализацией результатов. | Работа студента должна включать: 1. Загрузку и предобработку изображений. 2. Построение и компиляцию сверточной нейронной сети. 3. Обучение модели на тренировочных данных. 4. Оценку точности модели на тестовых данных. 5. Визуализацию результатов обучения (например, графики потерь и точности, отображение предсказаний на тестовых изображениях). На входе: Набор изображений (например, CIFAR-10 или MNIST). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку TensorFlow/Keras и библиотеку для визуализации (например, Matplotlib). |

**Задание № 9**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 9: Применение рекуррентной нейронной сети для анализа текстов.**

Необходимо создать и обучить рекуррентную нейронную сеть для задачи классификации текстов (например, анализ тональности) с использованием библиотеки TensorFlow/Keras. Требуется выполнить предобработку текстовых данных, построить модель, обучить её, провести оценку точности модели и визуализировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта в области обработки естественного языка с использованием библиотек на языке программирования Python при условии постановки задачи и наличия данных»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Отсутствие обучения модели или обучение выполнено некорректно. o Отсутствие оценки точности модели. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели. • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели. • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели с визуализацией результатов. | Работа студента должна включать: 1. Загрузку и предобработку текстовых данных. 2. Построение и компиляцию рекуррентной нейронной сети. 3. Обучение модели на тренировочных данных. 4. Оценку точности модели на тестовых данных. 5. Визуализацию результатов обучения (например, графики потерь и точности, примеры классифицированных текстов). На входе: Текстовые данные (например, набор отзывов с метками тональности). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку TensorFlow/Keras и библиотеку для визуализации (например, Matplotlib). |

**Задание № 10**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 10: Оценка точности модели классификации.**

Используя библиотеку scikit-learn, необходимо обучить модель классификации (например, логистическая регрессия или дерево решений) на предоставленном наборе данных. Требуется провести оценку точности модели, используя кросс-валидацию, построить и интерпретировать метрики оценки модели (точность, полнота, F-мера, ROC-AUC), а также визуализировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь проводить оценку точности работы моделей искусственного интеллекта с использованием библиотек на языке программирования Python выявляя факторы оказывающие влияние на показатели точности»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Оценка точности модели отсутствует или выполнена некорректно. o Метрики оценки модели не рассчитаны или рассчитаны некорректно. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели (например, только точность). o Рассчитаны основные метрики оценки (точность, полнота, F-мера). • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели с использованием кросс-валидации. o Рассчитаны и интерпретированы основные метрики оценки модели (точность, полнота, F-мера, ROC-AUC). • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели с использованием кросс-валидации и дополнительных методов (например, ансамблирование). o Рассчитаны и детально интерпретированы все ключевые метрики оценки модели. o Визуализированы результаты (ROC-кривая, матрица ошибок). | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Построение и компиляцию модели классификации. 3. Оценку точности модели с использованием кросс-валидации. 4. Расчет и интерпретацию метрик оценки (точность, полнота, F-мера, ROC-AUC). 5. Визуализацию результатов (ROC-кривая, матрица ошибок). На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку scikit-learn и библиотеку для визуализации (например, Matplotlib). |

Предмет оценки: «Уметь выявлять проблему переобучения моделей искусственного интеллекта с помощью оценки точности работы моделей на различных размерах датасета и количестве циклов обучения»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Оценка точности модели отсутствует или выполнена некорректно. o Метрики оценки модели не рассчитаны или рассчитаны некорректно. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели (например, только точность). o Рассчитаны основные метрики оценки (точность, полнота, F-мера). • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели с использованием кросс-валидации. o Рассчитаны и интерпретированы основные метрики оценки модели (точность, полнота, F-мера, ROC-AUC). • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели с использованием кросс-валидации и дополнительных методов (например, ансамблирование). o Рассчитаны и детально интерпретированы все ключевые метрики оценки модели. o Визуализированы результаты (ROC-кривая, матрица ошибок). | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Построение и компиляцию модели классификации. 3. Оценку точности модели с использованием кросс-валидации. 4. Расчет и интерпретацию метрик оценки (точность, полнота, F-мера, ROC-AUC). 5. Визуализацию результатов (ROC-кривая, матрица ошибок). На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку scikit-learn и библиотеку для визуализации (например, Matplotlib). |

**Задание № 11**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 10: Оценка модели регрессии.**

Используя библиотеку scikit-learn, необходимо обучить модель регрессии (например, линейная регрессия или решающее дерево) на предоставленном наборе данных. Требуется провести оценку точности модели, используя кросс-валидацию, построить и интерпретировать метрики оценки модели (MAE, MSE, R²), а также визуализировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь проводить оценку точности работы моделей искусственного интеллекта с использованием библиотек на языке программирования Python выявляя факторы оказывающие влияние на показатели точности»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Оценка точности модели отсутствует или выполнена некорректно. o Метрики оценки модели не рассчитаны или рассчитаны некорректно. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели (например, только R²). o Рассчитаны основные метрики оценки (MAE, MSE, R²). • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели с использованием кросс-валидации. o Рассчитаны и интерпретированы основные метрики оценки модели (MAE, MSE, R²). • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели с использованием кросс-валидации и дополнительных методов. o Рассчитаны и детально интерпретированы все ключевые метрики оценки модели. o Визуализированы результаты (графики ошибок, распределение остатков). | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Построение и компиляцию модели регрессии. 3. Оценку точности модели с использованием кросс-валидации. 4. Расчет и интерпретацию метрик оценки (MAE, MSE, R²). 5. Визуализацию результатов (графики ошибок, распределение остатков). На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку scikit-learn и библиотеку для визуализации (например, Matplotlib). |

Предмет оценки: «Уметь выявлять проблему переобучения моделей искусственного интеллекта с помощью оценки точности работы моделей на различных размерах датасета и количестве циклов обучения»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Оценка точности модели отсутствует или выполнена некорректно. o Метрики оценки модели не рассчитаны или рассчитаны некорректно. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели (например, только R²). o Рассчитаны основные метрики оценки (MAE, MSE, R²). • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели с использованием кросс-валидации. o Рассчитаны и интерпретированы основные метрики оценки модели (MAE, MSE, R²). • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели с использованием кросс-валидации и дополнительных методов. o Рассчитаны и детально интерпретированы все ключевые метрики оценки модели. o Визуализированы результаты (графики ошибок, распределение остатков). | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Построение и компиляцию модели регрессии. 3. Оценку точности модели с использованием кросс-валидации. 4. Расчет и интерпретацию метрик оценки (MAE, MSE, R²). 5. Визуализацию результатов (графики ошибок, распределение остатков). На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку scikit-learn и библиотеку для визуализации (например, Matplotlib). |

**Задание № 12**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 12: Анализ переобучения модели.**

Необходимо обучить модель классификации (например, решающее дерево) на предоставленном наборе данных и провести анализ переобучения. Требуется провести оценку точности модели на обучающей и тестовой выборках, построить и интерпретировать метрики оценки, а также визуализировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь выявлять проблему переобучения моделей искусственного интеллекта с помощью оценки точности работы моделей на различных размерах датасета и количестве циклов обучения»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Оценка точности модели отсутствует или выполнена некорректно. o Метрики оценки модели не рассчитаны или рассчитаны некорректно. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведена базовая оценка точности модели (например, только точность). o Рассчитаны основные метрики оценки (точность, полнота, F-мера). • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведена детализированная оценка точности модели на обучающей и тестовой выборках. o Рассчитаны и интерпретированы основные метрики оценки модели (точность, полнота, F-мера, ROC-AUC). • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведена детализированная оценка точности модели на обучающей и тестовой выборках. o Рассчитаны и детально интерпретированы все ключевые метрики оценки модели. o Визуализированы результаты (графики обучения, матрица ошибок). | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Построение и компиляцию модели классификации. 3. Оценку точности модели на обучающей и тестовой выборках. 4. Расчет и интерпретацию метрик оценки (точность, полнота, F-мера, ROC-AUC). 5. Визуализацию результатов (графики обучения, матрица ошибок). На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеку scikit-learn и библиотеку для визуализации (например, Matplotlib). |

**Задание № 13**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 13: Подбор гиперпараметров для модели классификации с использованием GridSearchCV.**

Необходимо обучить модель классификации (например, SVM) на предоставленном наборе данных, используя метод GridSearchCV для подбора гиперпараметров. Требуется провести оценку точности модели до и после подбора гиперпараметров, а также визуализировать результаты.

**Место выполнения**

На учебном оборудованиии

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь подбирать параметры алгоритмов и технологий обучения моделей искусственного интеллекта с использованием библиотек на языке программирования Python для достижения лучших показателей обучаемости моделей искусственного интеллекта»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Подбор гиперпараметров не проведен или проведен некорректно. o Оценка точности модели не проведена или проведена некорректно. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведен базовый подбор гиперпараметров. o Проведена базовая оценка точности модели. • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведен детализированный подбор гиперпараметров. o Проведена детализированная оценка точности модели. • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведен тщательный подбор гиперпараметров с использованием GridSearchCV. o Проведена детализированная оценка точности модели до и после подбора гиперпараметров. o Визуализированы результаты (например, графики зависимости точности от параметров). | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Построение и компиляцию модели классификации. 3. Подбор гиперпараметров с использованием GridSearchCV. 4. Оценку точности модели до и после подбора гиперпараметров. 5. Визуализацию результатов. На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеки scikit-learn и Matplotlib. |

**Задание № 14**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 13: Автоматический подбор гиперпараметров с использованием RandomizedSearchCV.**

Необходимо обучить модель классификации (например, RandomForest) на предоставленном наборе данных, используя метод RandomizedSearchCV для подбора гиперпараметров. Требуется провести оценку точности модели до и после подбора гиперпараметров, а также визуализировать результаты.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь подбирать параметры алгоритмов и технологий обучения моделей искусственного интеллекта с использованием библиотек на языке программирования Python для достижения лучших показателей обучаемости моделей искусственного интеллекта»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Данные не загружены или загружены с ошибками. o Модель не построена или построена некорректно. o Подбор гиперпараметров не проведен или проведен некорректно. o Оценка точности модели не проведена или проведена некорректно. • Удовлетворительно: o Данные загружены корректно. o Модель построена и обучена. o Проведен базовый подбор гиперпараметров. o Проведена базовая оценка точности модели. • Хорошо: o Данные загружены и предобработаны корректно. o Модель построена и обучена корректно. o Проведен детализированный подбор гиперпараметров. o Проведена детализированная оценка точности модели. • Отлично: o Данные загружены, предобработаны и использованы без ошибок. o Модель построена, обучена и оптимизирована. o Проведен тщательный подбор гиперпараметров с использованием RandomizedSearchCV. o Проведена детализированная оценка точности модели до и после подбора гиперпараметров. o Визуализированы результаты (например, графики зависимости точности от параметров). | Работа студента должна включать: 1. Загрузку и предобработку данных. 2. Построение и компиляцию модели классификации. 3. Подбор гиперпараметров с использованием RandomizedSearchCV. 4. Оценку точности модели до и после подбора гиперпараметров. 5. Визуализацию результатов. На входе: Табличные данные (например, CSV-файл). На выходе: Jupyter Notebook файл с выполненными задачами и комментариями. Ограничения: Вся работа должна быть выполнена в рамках одного Jupyter Notebook. Необходимо использовать библиотеки scikit-learn и Matplotlib. |

**Задание № 15**

**Тип задания**

Задание на применение умений и навыков в реальных или модельных условиях

**Описание ситуации и постановка задачи**

**Задание 15. Сегментация объектов транспортной инфраструктуры на основе облака точек в 3D**

Вам предоставляется набор облаков точек, полученных с помощью LiDAR сканера, содержащих информацию о транспортной инфраструктуре (например, дороги, тротуары, здания, автомобили). Задача состоит в создании и обучении модели глубокого обучения для сегментации объектов транспортной инфраструктуры на основе этих данных.

**Место выполнения**

На учебном оборудовании

Максимальное время выполнения: 120 минут

**Критерии оценки**

Предмет оценки: «Уметь разрабатывать модели искусственного интеллекта с использованием библиотек на языке программирования Python для решения задач транспортной области»

|  |  |
| --- | --- |
| Объект оценки | Модельный элемент |
| Предметом оценки является исходный код представленный в Jupyter Notebook. Критерии оценки: • Неудовлетворительно: o Отсутствие или неполное выполнение предобработки облака точек. o Неудовлетворительное построение модели сегментации. o Отсутствие оценки качества модели или ее недостаточная точность. • Удовлетворительно: o Выполнена предобработка облака точек с учетом особенностей задачи. o Построена базовая модель сегментации на основе обученной нейронной сети. o Проведена базовая оценка качества модели. • Хорошо: o Проведена качественная предобработка облака точек с учетом особенностей транспортной инфраструктуры. o Построена и оптимизирована модель сегментации с использованием продвинутых методов. o Проведена детализированная оценка качества модели с использованием метрик, таких как IoU (Intersection over Union). • Отлично: o Выполнена качественная предобработка облака точек с использованием дополнительных источников данных. o Построена и оптимизирована высокоточная модель сегментации с использованием продвинутых архитектур нейронных сетей. o Проведена тщательная оценка качества модели с использованием кросс-валидации и ансамблей моделей. | Работа студента должна включать: 1. Загрузку облака точек и его предобработку (например, фильтрацию выбросов, нормализацию значений). 2. Анализ особенностей данных и выбор подходящей архитектуры нейронной сети для сегментации. 3. Обучение модели сегментации на тренировочных данных с использованием подходящих оптимизаторов и функций потерь. 4. Оценку качества модели на отложенной тестовой выборке с использованием метрик, таких как Intersection over Union (IoU). На входе: Набор облаков точек, содержащих информацию о транспортной инфраструктуре. На выходе: Jupyter Notebook с кодом, комментариями и визуализациями, демонстрирующими процесс предобработки данных, обучения модели и оценки ее качества. Ограничения: Использование библиотек для глубокого обучения, таких как TensorFlow или PyTorch, а также для работы с облаками точек, таких как open3d или PyntCloud. |