**Глоссарий основных терминов курса**

1. *Алгоритм (Algorithm).* Шаги или инструкции, которые описывают решение конкретной задачи или операцию.

2. *Багфиксинг (bug fixing)* – это процесс устранения неисправностей в ПО. Под неисправностью следует считать несоответствие поведения программы техническому заданию.

3. *Джун* (от англ. junior, младший) – работает над простыми задачами под руководством опытных коллег.

4. *Интерпретатор (Interpreter)*. Программа, которая выполняет и интерпретирует код программы построчно во время выполнения.

5. *Компилятор (Compiler*). Программа, которая преобразует весь исходный код программы в исполняемый машинный код до его выполнения.

6. *Костыль* – решение, которое сделано для устранения локальной проблемы в коде. Оно работает, но не решает проблемы в целом.

7. *Легаси* – с английского legacy переводится как «наследие». Легаси-код – это код, который перешёл «по наследству» от предыдущих разработчиков.

8. *Либа* – lib, library, библиотека.

9. *Мидл* (от англ. middle, средний) – может решать сложные задачи или писать код самостоятельно, старшие коллеги проверяют только результаты.

10. *Отладка (Debugging).* Процесс поиска и исправления ошибок в программе. Отладка помогает идентифицировать проблемные места в коде и устранить ошибки.

11. *ООП (Object-Oriented Programming) (объектно-ориентированное программирование)* - методология программирования, в которой код организуется в объекты, которые взаимодействуют друг с другом через методы и свойства.

12. *Релиз (release)* – готовая программа.

13. *Сеньор* (от англ. senior, старший) – всё делает сам: пишет код, занимается архитектурой, обучает младших коллег. Решает сложные задачи и отвечает за результаты.

14. *Технологический стек* (stack) – сложная комбинация, включающая языки программирования, программное обеспечение и спектр фреймворков, применяемых для разработки IT-проекта.

15. *Фича* – это сленговое слово, которое пришло в русский язык из компьютерного и в переводе с английского означает «особенность» (feature). Это не просто отличительная черта, а именно полезная дополнительная функция, классная доработка или привлекательная деталь гаджета, улучшенная характеристика продукта.

16. *API (Application Programming Interface)*. Набор инструкций и правил, которые определяют, как различные программы и компоненты программного обеспечения могут взаимодействовать между собой.

17. *Backend-разработка* – это набор аппаратно-программных средств, при помощи которых реализована логика работы сайта. Попросту говоря, это то, что скрыто от глаз пользователя и происходит вне его браузера и компьютера.

18. *Frontend -разработка* – это создание пользовательского интерфейса на клиентской стороне веб‑сайта или приложения. Это всё, что видит пользователь, когда открывает веб-страницу, и с чем он взаимодействует: кнопки, баннеры и анимация

19. *Framework (фреймворк)* – это программная платформа, которая упрощает разработку программного продукта, определяет структуру проекта и помогает удобно объединять в нём разные компоненты. Пример: Unity 3D, Django, Google.Colab, React.

20. *Git*. Распределенная система контроля версий, используемая для отслеживания изменений в коде и совместной работы над проектами.

21. *IDE (Integrated Development Environment).* Интегрированная среда разработки, которая предоставляет программисту удобную среду для написания, отладки и выполнения программного кода.

22. *PIP (Python Package Installer)*. Инструмент командной строки, используемый для установки и управления пакетами Python. PIP упрощает процесс установки сторонних библиотек и модулей.

23. *Package (пакет).* Способ организации и структурирования модулей Python. Пакеты позволяют группировать связанный функционал вместе для удобного использования.

**Раздел «Алгоритмы машинного обучения и продвинутое программирование на Python»**

1. Аналоговое изображение – двумерное изображение F(x,у), характеризующееся бесконечной точностью представления по пространственным параметрам х и у и бесконечной точностью представления значений интенсивности в каждой пространственной точке (х, у).

2. *Ансамбль* — это составная модель, объединяющая ряд классификаторов с низкой производительностью с целью создания улучшенного классификатора.

3. Аппроксимация — способ выбрать из семейства «простых» функций приближение для «сложной» функции на отрезке, при этом ошибка не должна превышать определенного предела.

4.*Boosting (бустинг)* — набор классификаторов низкой точности (weak classifier) для создания классификатора высокой точности (strong classifier).

5. *Бинарное изображение* – цифровое изображение, пиксели которого имеют значения 0 или 1

6. *Bagging (Бэггинг)* – начальная агрегация. Он объединяет нескольких классификаторов (регрессия, решающие деревья, kNN и т.д.) таким образом, чтобы уменьшить дисперсию оценок.

7. *Деревья решений* — это непараметрический контролируемый метод обучения, используемый для классификации и регрессии.

8. *Дисперсия* — это величина, показывающая, как именно и насколько сильно разбросаны значения — например, предсказания модели машинного обучения или доход за рассматриваемый период.

9. *Интерполяция* — способ выбрать из семейства функций ту, которая проходит через заданные точки.

10. *Линейная регрессия (англ. Linear regression)* — используемая в статистике регрессионная модель зависимости одной переменной y от другой или нескольких других переменных (факторов, регрессоров, независимых переменных) x с линейной функцией зависимости.

11. *Маркированное изображение* – это цифровое изображение I[r,c], пиксели которого представлены в виде символов конечного алфавита.

12. *Метаэвристика* – метод оптимизации, многократно использующий простые правила или эвристики для достижения оптимального или субоптимального решения.

13. *Метод k-ближайших соседей* (англ. k-nearest neighbors algorithm, k-NN) — метрический алгоритм для автоматической классификации объектов или регрессии.

14. *Многоспектральное (мультиспектральное) изображение* – двумерное изображение М[х, у], у которого каждой пространственной точке или пикселу соответствует вектор значений. В наиболее часто встречающемся случае цветных изображений вектор содержит 3 элемента.

15. *Недостаточная подгонка, обучение (Underfitting)* возникает, когда модель не может точно отразить зависимости между данными, обычно из-за своей простоты.

16. *Обучение с подкреплением (Q-learning)* — метод, применяемый в искусственном интеллекте при агентом подходе. Относится к экспериментам вида обучение с подкреплением. На основе получаемого от среды вознаграждения агент формирует функцию полезности Q, что впоследствии дает ему возможность уже не случайно выбирать стратегию поведения, а учитывать опыт предыдущего взаимодействия со средой.

17. *Переобучение (Overfitting)* происходит, когда модель учитывает как зависимости данных, так и случайные колебания (шум, погрешность измерений).

18. *Полутоновое изображение* – монохромное (черно­белое) цифровое изображение I[r,c], у которого каждому пикселю соответствует одно значение интенсивности. В случае полутонового изображения интенсивность часто также называется яркостью пикселей.

19. *Полиномиальная регрессия* — это обобщенный случай линейной регрессии.

20. *Регрессия* — способ выбрать из семейства функций ту, которая минимизирует функцию потерь.

21. *Функция интенсивности* – математическое представление изображения как функции f(x,у), зависящей от двух пространственных переменных х и у. Переменные х и у принимают действительные значения, которые задают положение точки на изображении. Значения f(x,y) обычно также являются действительными и определяют интенсивность изображения в точке (х,у).

22. *Цифровое изображение* – двумерное изображение I[r,c], представленное в виде двумерного массива дискретных значений интенсивности, каждое из которых представлено с ограниченной точностью.

23. *Экстраполяция* — предсказание поведения функции вне интервала.

**Раздел «Нейронные сети»**

1.

2.

3.

4.

5.

6.

7.

8.

9.

10.

11.

12.

13.

14.

15.