

УДК 004.432

https://doi.org/10.33619/2414-2948/96/37

## ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ С ИСПОЛЬЗОВАНИЕМ PYTHON: ТЕХНОЛОГИИ И ПРИМЕНЕНИЕ

©**Пирматов А. З.**, ORCID: 0009-0008-2343-5185, SPIN-код: 8965-9182, канд. физ.-мат. наук,  
Ошский государственный университет, г. Ош, Кыргызстан, pirmatov@ohsu.kg

©**Азимов Б. А.**, ORCID: 0000-0001-5849-8583, канд. физ.-мат. наук,

Ошский государственный университет, г. Ош, Кыргызстан, azimov@ohsu.kg

©**Камалов С. С.**, ORCID: 0009-0002-9944-9782, Ошский государственный университет,  
г. Ош, Кыргызстан, sultan.kamalov.96@mail.ru

## ARTIFICIAL INTELLIGENCE USING PYTHON: TECHNOLOGIES AND APPLICATIONS

©**Pirmatov A.**, ORCID: 0009-0008-2343-5185, SPIN-code: 8965-9182, Ph.D.,  
Osh State University, Osh, Kyrgyzstan, pirmatov@ohsu.kg

©**Azimov B.**, ORCID: 0000-0001-5849-8583, Ph.D., Osh State University,  
Osh, Kyrgyzstan, azimov@ohsu.kg

©**Kamalov S.**, ORCID: 0009-0002-9944-9782, Osh State University,  
Osh, Kyrgyzstan, sultan.kamalov.96@mail.ru

*Аннотация.* В статье рассматривается важная роль Python в разработке и применении искусственного интеллекта (ИИ). Освещены основные понятия ИИ, такие как нейронные сети, машинное обучение и глубокое обучение, а также ключевые библиотеки Python, используемые в этой области. Дается краткий обзор истории ИИ, начиная с его ранних концепций и заканчивая современными достижениями. Python рассматривается как основной инструмент в разработке ИИ, а также представлены популярные библиотеки и фреймворки, такие как TensorFlow, PyTorch и scikit-learn. Статья включает конкретные примеры кода и демонстрации операций с библиотеками Python для машинного и глубокого обучения. Рассматриваются области применения ИИ, такие как обработка естественного языка, компьютерное зрение, рекомендательные системы и робототехника, а также приводятся примеры успешных проектов в этих областях.

*Abstract.* This article explores the significant role of Python in the development and application of artificial intelligence (AI). It covers fundamental AI concepts such as neural networks, machine learning, and deep learning, as well as key Python libraries used in this field. A brief overview of the history of AI is provided, starting from its early concepts and progressing to modern achievements. Python is regarded as the primary tool in AI development, and popular libraries and frameworks like TensorFlow, PyTorch, and scikit-learn are introduced. The article includes specific code examples and demonstrations of operations with Python libraries for machine and deep learning. It discusses AI application areas such as natural language processing, computer vision, recommendation systems, and robotics, along with examples of successful projects in these domains.

*Ключевые слова:* искусственный интеллект, питон, технологии, нейронные сети.

*Keywords:* artificial intelligence, Python, technologies, neural networks.

Искусственный интеллект (ИИ) становится неотъемлемой частью нашей повседневной жизни и бизнес-среды. Рассмотрим, как Python, один из наиболее популярных языков программирования высокого уровня, играет ключевую роль в разработке и применении ИИ-технологий.

Объяснение ключевых терминов, связанных с искусственным интеллектом (ИИ), поможет читателям лучше понять основные концепции и технологии, которые используются в этой области. Опишем некоторые из этих терминов:

1. *Искусственный интеллект (ИИ)* — это область компьютерной науки, которая занимается созданием компьютерных систем и программ, способных выполнять задачи, которые обычно требуют интеллектуальных способностей человека, такие как распознавание образов, обучение, планирование и принятие решений.

2. *Нейронные сети* — это модели, вдохновленные структурой и функцией человеческого мозга. Они состоят из множества искусственных нейронов, которые связаны между собой и используются для выполнения задач машинного обучения и глубокого обучения.

3. *Машинное обучение* — это подраздел ИИ, который фокусируется на разработке алгоритмов и моделей, которые позволяют компьютерам «учиться» из данных и опыта. Машинное обучение может быть разделено на три основные категории: обучение с учителем, обучение без учителя и обучение с подкреплением.

4. *Глубокое обучение* — это подраздел машинного обучения, в котором используются глубокие нейронные сети с большим количеством слоев (глубоких архитектур). Глубокое обучение позволяет моделям извлекать сложные признаки и решать задачи, такие как распознавание образов и обработка естественного языка.

5. *Распознавание образов* — это задача, в которой компьютерные системы анализируют входные данные, такие как изображения или звук, и пытаются определить их содержание или классифицировать их.

6. *Обработка естественного языка (NLP)* — это область ИИ, которая занимается анализом и генерацией естественного языка человека. Она используется для разработки систем автоматического перевода, чат-ботов, анализа текстов и многих других приложений.

7. *Рекомендательные системы* — это приложения ИИ, которые предоставляют персонализированные рекомендации для пользователей на основе их предпочтений и поведения.

8. *Анализ данных* — это процесс извлечения информации и знаний из больших объемов данных с использованием методов машинного обучения и статистических подходов.

9. *Интеллектуальные агенты* — это компьютерные программы или системы, способные взаимодействовать с окружающей средой, обучаться, принимать решения и действовать автономно.

Эти термины представляют собой ключевую терминологию в области искусственного интеллекта и машинного обучения, и их понимание является важным для работы с этими технологиями и разработки ИИ-приложений на Python.

Python играет значительную роль в разработке искусственного интеллекта (ИИ) и считается одним из наиболее популярных и предпочтительных языков программирования в этой области. Вот преимущества роли Python в разработке ИИ:

1. *Простота и читаемость*: Python отличается простым и понятным синтаксисом, который делает код легким для написания и понимания. Это особенно важно в разработке ИИ, где чистота и понимание кода имеют важное значение.

2. *Богатая экосистема библиотек:* Python обладает обширной экосистемой библиотек и фреймворков, специально разработанных для ИИ и машинного обучения. Некоторые из наиболее известных библиотек включают TensorFlow, PyTorch, scikit-learn, Keras, и другие. Эти библиотеки предоставляют мощные инструменты для создания и обучения моделей ИИ.

3. *Машинное обучение и глубокое обучение:* Python является популярным языком для реализации методов машинного обучения и глубокого обучения. Библиотеки, такие как TensorFlow и PyTorch, предоставляют гибкие и эффективные инструменты для создания и обучения нейронных сетей.

4. *Обработка данных:* Python также широко используется для обработки и анализа данных, что является важной частью подготовки данных для обучения моделей ИИ. Библиотеки, такие как NumPy, pandas и Matplotlib, упрощают работу с данными.

5. *Интеграция и разработка прототипов:* Python легко интегрируется с другими языками программирования и позволяет быстро создавать прототипы ИИ-систем. Это особенно полезно при разработке и тестировании новых алгоритмов и моделей.

6. *Сообщество и ресурсы:* Python имеет активное и обширное сообщество разработчиков, что обеспечивает доступ к обучающим материалам, форумам, библиотекам и инструментам. Это упрощает обучение и разработку в области ИИ на Python.

7. *Кроссплатформенность:* Python поддерживается на различных платформах, что делает его удобным выбором для разработки ИИ-приложений, независимо от операционной системы.

8. *Интеграция с большим количеством приложений:* Python может быть легко интегрирован с другими приложениями и технологиями, что делает его универсальным инструментом для создания различных ИИ-решений, включая веб-службы, мобильные приложения и другие.

Исходя из этих факторов, Python остается одним из наиболее популярных и широко используемых языков программирования в области искусственного интеллекта и машинного обучения. Существует множество популярных библиотек и фреймворков, которые используются для создания и разработки искусственного интеллекта на Python. Вот некоторые из наиболее известных:

1. *TensorFlow:* TensorFlow разработан Google и является одним из наиболее популярных фреймворков для глубокого обучения.

Предоставляет гибкие инструменты для создания и обучения нейронных сетей, включая поддержку для развертывания моделей на различных платформах.

TensorFlow 2.0 включает в себя удобный высокоуровневый API Keras.

2. *PyTorch:* PyTorch разработан Facebook и широко используется исследователями и инженерами в области глубокого обучения.

Отличается динамическим графом, что упрощает создание и изменение нейронных сетей. Популярен в академических кругах и научных исследованиях.

3. *Scikit-learn:* Scikit-learn — это библиотека машинного обучения с открытым исходным кодом, которая предоставляет множество алгоритмов машинного обучения и инструментов для анализа данных. Используется для задач классификации, регрессии, кластеризации, извлечения признаков и многих других.

4. *Keras:* Keras — это высокоуровневый API для создания нейронных сетей, который может быть выполнен поверх TensorFlow, Theano или CNTK.

Предоставляет простой и интуитивно понятный способ создания и обучения нейронных сетей.

5. *Theano*: Theano — это библиотека глубокого обучения, разработанная для оптимизации вычислений с использованием графовых вычислений.

Хотя Theano больше не активно разрабатывается, она оставила значительный след в исследованиях по глубокому обучению.

6. *MXNet*: MXNet — это фреймворк для глубокого обучения, разработанный Apache Software Foundation. Он обеспечивает высокую производительность и поддерживает множество языков программирования, включая Python.

7. *CNTK (Microsoft Cognitive Toolkit)*: CNTK — это фреймворк глубокого обучения, разработанный Microsoft, который может использоваться для создания и обучения нейронных сетей.

8. *XGBoost*: XGBoost — это библиотека машинного обучения, специализирующаяся на градиентном бустинге, и часто используется для задач классификации и регрессии.

Это лишь несколько примеров библиотек и фреймворков, доступных в экосистеме Python для разработки искусственного интеллекта и машинного обучения. Каждый из них имеет свои особенности и преимущества, и выбор зависит от конкретных задач и предпочтений разработчика.

Приведем пример кода и демонстрации базовых операций с вышеуказанными библиотеками для машинного обучения и глубокого обучения на Python.

Установка *TensorFlow* с помощью менеджера пакетов Python pip

```
# Требуется последняя версия pip
```

```
pip install --upgrade pip
```

```
# Установка tensorflow
```

```
pip install tensorflow
```

Создание пустого графа

```
import tensorflow as tf # импорт библиотеки tensorflow
```

```
# сохраняем граф по-умолчанию в переменную
```

```
default_graph = tf.get_default_graph()
```

```
# объявляем константу в графе по-умолчанию
```

```
c1 = tf.constant(1.0)
```

```
# создаём пустой граф
```

```
second_graph = tf.Graph()
```

```
with second_graph.as_default():
```

```
    # в этом блоке мы работаем во втором графе
```

```
    c2 = tf.constant(101.0)
```

```
print(c2.graph is second_graph, c1.graph is second_graph)
```

```
print(c2.graph is default_graph, c1.graph is default_graph)
```

Установка *PyTorch* с помощью менеджера пакетов Python pip

```
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl/cu117
```

```
# импорт библиотеки pytorch
```

```
import torch
```

```
import torch.nn as nn
```

Проекты и примеры успешного применения искусственного интеллекта (ИИ) в различных областях с каждым днем становятся все более разнообразными и впечатляющими. Вот несколько примеров успешных проектов в разных областях:

1. Обработка естественного языка (NLP): *GPT-3 от OpenAI: GPT-3 (Generative Pre-trained Transformer 3)* — это мощная модель генерации текста, которая может генерировать естественно звучащие тексты, отвечать на вопросы, создавать статьи и даже писать код на нескольких языках.

2. *Компьютерное зрение: Proximus AI (беспилотные дроны для агрокультур):* Этот проект использует дроны с компьютерным зрением, чтобы помочь фермерам улучшить управление своими полями, выявлять болезни растений и оптимизировать урожай.

3. *Медицина: IBM Watson for Oncology:* IBM Watson используется в онкологии для анализа медицинских данных и предоставления рекомендаций по лечению рака. Он помогает врачам выбирать наилучшие методы лечения для пациентов.

4. *Автономные транспортные средства: Waymo (бывший проект Google Self-Driving Car):* Waymo разрабатывает технологию автономных автомобилей, которая успешно проходит испытания на дорогах и используется в пилотных проектах по такси без водителей.

5. *Финансы: Alibaba's Ant Financial (Alipay):* Использует ИИ для анализа транзакций, определения мошеннических операций и предоставления финансовых услуг.

6. *Рекомендательные системы: Netflix's Recommendation System:* Netflix использует ИИ для рекомендации фильмов и телешоу на основе истории просмотров пользователя и схожих интересов.

7. *Игровая индустрия: AlphaGo от DeepMind:* AlphaGo стал первой программой, которая обыграла чемпиона мира по го, древней игры, считающейся одной из самых сложных для искусственного интеллекта.

8. *Робототехника: Boston Dynamics' Spot:* Этот робот используется в различных сферах, включая монтаж и обслуживание оборудования, проверку на опасные участки и даже развлекательную индустрию.

9. *Образование: Duolingo:* Duolingo использует технологии ИИ для персонализированного обучения и оценки знаний в изучении иностранных языков.

10. *Научные исследования: DeepMind's AlphaFold:* AlphaFold использует глубокое обучение для предсказания структуры белков, что имеет фундаментальное значение для биологических и медицинских исследований.

Это всего лишь несколько примеров успешных проектов в разных областях, и список постоянно расширяется. Разработка искусственного интеллекта (ИИ) — это сложный процесс, включающий несколько этапов. Ниже приведено подробное описание этих этапов:

1. *Сбор данных:* собранные данные являются основой для разработки модели ИИ. Этот этап включает в себя:

*Определение целей:* определите, какие задачи вы хотите решить с помощью ИИ, и какие данные вам нужны для этого.

*Поиск и сбор данных:* на этом этапе вы ищете и собираете необходимые данные. Данные могут быть структурированными (например, таблицы данных) или неструктурированными (например, текст, изображения, аудио).

*Очистка и обработка данных:* Обработка данных может включать в себя удаление дубликатов, заполнение пропущенных значений, нормализацию данных и другие операции для обеспечения качества данных.

2. *Подготовка данных:* на этом этапе данные подготавливаются для использования в обучении моделей. Это включает в себя:

*Разделение данных:* данные обычно разделяют на обучающий набор (для обучения модели) и тестовый набор (для оценки производительности модели).

*Инженерия признаков:* выбор и создание признаков (факторов), которые будут использоваться для обучения модели.

*Преобразование данных:* масштабирование, кодирование категориальных признаков и другие операции для подготовки данных к обучению.

3. Выбор алгоритмов и моделей: на этом этапе выбираются алгоритмы и модели машинного обучения или глубокого обучения, которые будут использоваться для решения задачи. Выбор зависит от типа задачи (классификация, регрессия, кластеризация и др.) и характера данных.

Выбор архитектуры модели: выберите тип модели, такой как линейная регрессия, случайный лес, нейронные сети и др.

Гиперпараметры: настройте параметры модели, такие как скорость обучения, глубина деревьев (для деревьев решений), количество слоев и нейронов (для нейронных сетей) и другие параметры.

4. Обучение моделей: на этом этапе модели обучаются на обучающем наборе данных с использованием выбранных алгоритмов и параметров.

Процесс обучения: модель подстраивается под данные, минимизируя функцию потерь. Это происходит путем корректировки весов (параметров) модели.

5. Оценка моделей: после обучения моделей оценивается их производительность на тестовом наборе данных.

Метрики оценки: используйте метрики, такие как точность, F1-мера, средняя абсолютная ошибка и др., чтобы оценить, насколько хорошо модель выполняет задачу.

6. Тестирование и настройка: на этом этапе происходит тестирование модели в реальных условиях и, если необходимо, настройка параметров модели.

Кросс-валидация: применяется для проверки стабильности и обобщающей способности модели.

Настройка гиперпараметров: если модель не удовлетворяет требованиям, можно попробовать разные значения гиперпараметров и выбрать наилучшие.

7. Внедрение и мониторинг: после успешного обучения и тестирования модели она готова к внедрению в реальное окружение.

Интеграция в систему: модель интегрируется в существующие системы и приложения.

Мониторинг и обслуживание: модель регулярно мониторится и обслуживается для обеспечения ее актуальности и производительности.

Эти этапы могут различаться в зависимости от конкретной задачи и проекта, но общий процесс разработки ИИ включает в себя сбор и подготовку данных, выбор и обучение модели, ее оценку и настройку, а также внедрение и мониторинг в реальном мире.

Пример кода и инструкции для каждого этапа на примере задачи классификации с использованием библиотеки *scikit-learn* в Python.

Установка *scikit-learn* с помощью менеджера пакетов Python *pip*  
`pip install -U scikit-learn`

```
Чтобы проверить установку, вы можете использовать
# чтобы узнать, какая версия и где установлена scikit-learn
python -m pip show scikit-learn
# чтобы увидеть все пакеты, установленные в активной виртуальной среде
python -m pip freeze
python -c "import sklearn; sklearn.show_versions()"
```

```
# импорт библиотеки scikit learn
from sklearn import datasets
# Загрузить данные
iris= datasets.load_iris()
# Распечатайте форму данных, чтобы подтвердить загрузку данных.
print(iris.data.shape)
```

№4

```
#Установка keras с помощью менеджера пакетов Python pip  
pip install keras
```

```
# импорт библиотеки pytorch  
import keras  
from keras.datasets import mnist
```

Искусственный интеллект (ИИ) является одной из самых динамично развивающихся областей в современной технологии, и его влияние на мир становится все более значительным. В данной статье была представлена обширная информация о ИИ, его роли в современном мире и ключевых аспектах его разработки и применения. Вот основные выводы:

*Роль ИИ в современном мире:* ИИ играет центральную роль в решении различных задач и проблем в различных областях, включая медицину, транспорт, финансы, образование и многие другие. Он способствует автоматизации и оптимизации процессов, предоставляет персонализированные рекомендации и улучшает качество жизни людей.

*Основные аспекты разработки ИИ:* Разработка ИИ включает в себя этапы сбора и подготовки данных, выбора алгоритмов и моделей, обучения моделей и их оценки. Каждый этап требует специфических навыков и инструментов.

*Вызовы и ограничения ИИ:* В разработке и применении ИИ существует ряд вызовов и ограничений, таких как ограниченная интерпретируемость моделей, недостаток данных, алгоритмический bias, конфиденциальность данных и юридическое регулирование. Решение этих вызовов требует усилий в области этики, технологии и законодательства.

*Будущее ИИ:* ИИ продолжает развиваться, и его будущее обещает более сложные и интеллектуальные системы, способные решать более широкий спектр задач. Прогнозы в области глубокого обучения, автономных систем и робототехники обещают увеличить влияние ИИ на мир.

*Безопасность и этика:* С увеличением мощности и влияния ИИ становится все важнее обеспечивать безопасность и соблюдение этических норм в его разработке и применении.

Искусственный интеллект представляет собой важную технологическую и научную область, которая продолжит изменять наш мир в будущем. Он может решать сложные задачи и привносить инновации, но его разработка и использование также влечет за собой ответственность перед обществом. Научные и инженерные усилия в области ИИ продолжают расширять горизонты и создавать новые возможности для улучшения качества жизни и решения глобальных проблем.

#### Список литературы:

1. Гудфеллоу Я., Иошуа Б., Аарон К. Глубокое обучение. М.: ДМК Пресс, 2018. 651 с.
2. Рашка С., Мирджалили В. Python и машинное обучение: машинное и глубокое обучение с использованием Python, scikit-learn и TensorFlow 2. М.: Диалектика, 2019. 656 с.
3. Джоши П. Искусственный интеллект с примерами на Python: создание приложений искусственного интеллекта с помощью Python для взаимодействия с окружающим миром. М.: Диалектика, 2019. 444 с.
4. Пирматов А. З., Идрисова Г. И. Python программалоо тилинде файлдар менен иштөөнүн ыкмалары // Вестник Жалал-Абадского государственного университета. 2021. №2(47). С. 44-48.

5. Пирматов А. З., Камалов С. С., Абдукадыр К. А., Сүйөркул К. Н. Python тилиндеги объектке-багытталган программалоо // Вестник Жалал-Абадского государственного университета. 2022. №4. С. 22-28.

*References:*

1. Gudfellou, Ya., Ioshua, B., & Aaron, K. (2018). *Glubokoe obuchenie*. Moscow. (in Russian).
2. Rashka, S., & Mirdzhalili, V. (2019). *Python i mashinnoe obuchenie: mashinnoe i glubokoe obuchenie s ispol'zovaniem Python, scikit-learn i TensorFlow 2*. Moscow. (in Russian).
3. Dzhoshi, P. (2019). *Iskusstvennyi intellekt s primerami na Python: sozдание prilozhenii iskusstvennogo intellekta s pomoshch'yu Python dlya vzaimodeistviya s okruzhayushchim mirom*. Moscow. (in Russian).
4. Pirmatov, A. Z., & Idrisova, G. I. (2021). Python programmaloo tilinde faildar menen ishtöönyn ykmalary. *Vestnik Zhalal-Abadskogo gosudarstvennogo universiteta*, (2(47)), 44-48. (in Russian).
5. Pirmatov, A. Z., Kamalov, S. S., Abdukadyr, K. A., & Syörkul, K. N. (2022). Python tilindegi ob'ektke-bagyttalغان программалоо. *Vestnik Zhalal-Abadskogo gosudarstvennogo universiteta*, (4), 22-28. (in Russian).

*Работа поступила  
в редакцию 04.10.2023 г.*

*Принята к публикации  
18.10.2023 г.*

*Ссылка для цитирования:*

Пирматов А. З., Азимов Б. А., Камалов С. С. Искусственный интеллект с использованием Python: технологии и применение // Бюллетень науки и практики. 2023. Т. 9. №11. С. 288-295. <https://doi.org/10.33619/2414-2948/96/37>

*Cite as (APA):*

Pirmatov, A., Azimov, B., & Kamalov, S. (2023). Artificial Intelligence Using Python: Technologies and Applications. *Bulletin of Science and Practice*, 9(11), 288-295. (in Russian). <https://doi.org/10.33619/2414-2948/96/37>