МЕТОДЫ РАЗРАБОТКИ, ХРАНЕНИЯ И ОБУЧЕНИЯ МОДЕЛЕЙ НА ОСНОВЕ ТЕКСТОВОЙ ИНФОРМАЦИИ В ОБРАЗОВАНИИ

Норматов Иброхимали Холмаматович
Профессор Национального университета Узбекистана имени Мирзо Улугбека
E-mail: i_normatov@nuu.uz, tel: +998977210920
Джумагулов Нургали Султанкельдиевич
Преподаватель Шымкентского университета Республика Казахстан
Жолбарыс Ерғали Нурлыбайулы
Преподаватель Шымкентского университета Республика Казахстан

Аннотация. В статье рассматриваются методы разработки, сохранения и обучения моделей машинного обучения на основе текстовых данных. Особое внимание уделено понятию «пустой модели», её архитектуре и особенностям применения для казахского языка как языка с ограниченными цифровыми ресурсами. Описаны основные этапы предварительной обработки текста: нормализация, удаление стоп-слов, лемматизация и векторизация с использованием метода TF-IDF. Для обучения и классификации данных применены алгоритмы наивного Байеса, логистической регрессии и простых нейронных сетей. В качестве практической реализации создан прототип чат-модели с графическим интерфейсом на Tkinter, обеспечивающий взаимодействие пользователя с системой вопросов и ответов. Результаты исследования демонстрируют перспективность создания адаптивных моделей для работы с казахским языком и подтверждают значимость развития национальных языков в цифровой среде.

Ключевые слова: модель, алгоритм, метод, функция, архитектура, сеть, NLP, tkinter, прогнозирование, классификация**,** регрессия, кластеризация.

Annotation. The article discusses methods of developing, saving, and training machine learning models based on textual data. Special attention is given to the concept of a "blank model," its architecture, and its application for the Kazakh language as a low-resource language. The main stages of text preprocessing are described, including normalization, stop-word removal, lemmatization, and vectorization using the TF-IDF method. For training and data classification, algorithms such as Naive Bayes, logistic regression, and simple neural networks are applied. As a practical implementation, a prototype of a chat model with a graphical interface built on Tkinter was created, providing user interaction with the question–answer system. The results of the study show the prospects of creating adaptive models for the Kazakh language and highlight the importance of developing national languages in the digital environment.

ISSN 3030-3907

Key words: model, algorithm, method, function, architecture, network, NLP, tkinter, forecasting, classification, regression, clustering.

Аннотация. Мақолада таълимда матнли маълумотлар асосида машинали ўрганиш моделларини ишлаб чиқиш, сақлаш ва ўқитиш усуллари муҳокама қилинади. "Бўш модел" тушунчасига, унинг архитектураси ва чекланган рақамли ресурсларга эга тил сифатида қозоқ тили учун қўллашга алоҳида эътибор қаратилади. Матнни қайта ишлашнинг асосий босқичлари: нормаллаштириш, тўхташ сўзларини олиб ташлаш, TF-IDF усули ёрдамида лематизация ва векторизация тасвирланган. Маълумотларни ўргатиш ва таснифлаш учун содда Байес, логистик регрессия ва оддий нейрон тармоқ алгоритмларидан фойдаланилади. Уларни амалга ошириш учун ткинтер-да график интерфейсга эга чат моделининг прототипи яратилди, бу фойдаланувчининг савол-жавоб тизими билан ўзаро таъсирини таъминлайди. Тадқиқот натижалари қозоқ тили билан ишлаш учун мослашувчан моделларни яратиш истиқболларини намойиш етади ва рақамли муҳитда миллий тилларни ривожлантириш муҳимлигини тасдиқлайди.

Калит сўзлар: model, algoritm, usul, funksiya, arxitektura, tarmoq, NLP, tkinter, prognozlash, tasniflash, regressiya, klasterlash.

Введение

Технологии искусственного интеллекта (ИИ) в последние десятилетия превратились в одно из ключевых направлений развития науки и техники. Среди них особое место занимают методы машинного обучения (Machine Learning, ML), которые стремительно развиваются и внедряются в различные сферы человеческой деятельности. В настоящее время модели машинного обучения широко применяются для автоматизации таких сложных процессов, как обработка данных, прогнозирование, принятие решений и понимание естественного языка. Особенно возрастает интерес к моделям, работающим с текстовыми данными [1-9].

Текст является древнейшим и наиболее распространённым носителем информации в истории человеческой цивилизации. С момента появления письменности и до сегодняшнего дня текст служит основным средством хранения, передачи и распространения знаний. В современную эпоху практически все источники данных — интернет-ресурсы, социальные сети, электронная почта, научные статьи и официальные документы — представлены в текстовой форме. По оценкам экспертов, более 70% всей мировой информации хранится именно в текстовом виде. Следовательно, обработка текстовых данных, извлечение из них значимой информации и её

ISSN 3030-3907

практическое использование относятся к числу наиболее актуальных направлений современной науки.

Научное направление, обеспечивающее решение этих задач, — это обработка естественного языка (Natural Language Processing, NLP). С помощью методов NLP компьютер способен понимать человеческую речь, анализировать её и преобразовывать. Данные технологии широко применяются в системах автоматического перевода, при выделении ключевых идей из текста, в чат-ботах и интеллектуальных поисковых системах.

Однако многие существующие системы опираются исключительно на заранее подготовленные крупные языковые модели. Такие модели в основном обучены на англоязычных данных, что создаёт серьёзную проблему для языков с ограниченными ресурсами, в частности для казахского языка. Количество моделей, адаптированных для казахского языка, крайне невелико, а использование международных моделей не всегда оказывается эффективным. Это объясняется тем, что внутренняя структура таких моделей в большей степени отражает грамматические и семантические особенности английского языка [1-5].

В этой связи особое значение приобретает создание пустой модели и её обучение на основе текстовых данных на казахском и русском языках. Пустая модель — это архитектура, которая разработана, но ещё не обучена на конкретных данных. Такая модель изначально является нейтральной и может быть адаптирована к любому языку или предметной области. Подход, основанный на разработке пустых моделей, представляется как научно, так и практически более оправданным по сравнению с адаптацией готовых решений.

Цель исследования

Разработка пустой модели, её обучение на основе текстовых данных, сохранение и практическая реализация. Для достижения цели были поставлены следующие задачи:

1. Научный анализ понятия модели и характеристик пустой модели;
2. Рассмотрение методов сбора текстовых данных, их предварительной обработки и подготовки к обучению;
3. Описание архитектуры пустой модели;
4. Определение механизмов обучения, сохранения и повторного использования модели;
5. Экспериментальная проверка модели и создание чат-бота.

ISSN 3030-3907

Актуальность исследования связана с необходимостью разработки компактных и адаптивных моделей, способных работать с казахским языком и в многоязычной среде. Решение данной проблемы позволит интегрировать национальные языки в цифровую среду, создавать автоматизированные системы для образования, а также развивать интеллектуальные технологии анализа государственных и научных документов.

Таким образом, в данной статье рассматривается проблема разработки пустой модели и её обучения на текстовых данных. Результатом исследования является прототип чат-модели, способной отвечать на вопросы пользователей.

Понятие модели и ее разработка

Модель машинного обучения — это математическая или вычислительная конструкция, описывающая данные. Она применяется для прогнозирования, классификации или принятия решений на основе заданных входных данных. Модель является не только готовым алгоритмом, но и формальным отражением изучаемого явления.

С научной точки зрения модель представляет собой абстракцию, упрощающую описание сложных процессов реального мира. Она не охватывает всех свойств исследуемого объекта, а лишь фиксирует наиболее важные характеристики, необходимые для решения конкретной задачи. Например, при анализе текста модель учитывает порядок слов, их частотность и контекстные взаимосвязи, но игнорирует особенности графического написания слов.

Модели машинного обучения выполняют несколько ключевых функций:

1. **Прогнозирование** — предсказание будущих результатов на основе прошлых данных (например, определение тематики текста).
2. **Классификация** — распределение данных по заранее определённым категориям (например, отнесение электронного письма к «спаму» или «нормальной почте»).
3. **Регрессия** — вычисление непрерывных величин (например, прогнозирование оценочного балла на основе текстовых отзывов).
4. **Кластеризация** — выявление скрытых группировок без заранее заданных категорий (например, группировка текстов по тематическому сходству).

Для пояснения сущности модели её часто представляют как «чёрный ящик» (black box): на вход подаются данные, а на выходе формируется прогноз или ответ. Качество модели напрямую зависит от того, как настроены её внутренние параметры и на каких данных она была обучена.

ISSN 3030-3907

Пустая модель — это алгоритм, архитектура которого определена, но обучение на конкретных данных ещё не проводилось. Такая модель в исходном состоянии не даёт полезного результата, однако обладает высокой гибкостью. Её преимущество заключается в возможности адаптации к любому языку, предметной области или задаче. Например, пустая нейронная сеть представляет собой лишь структуру, а её «знания» формируются позднее — в процессе обучения на данных.

Таким образом, модель — это не только технический инструмент, но и своеобразный мост между реальными данными и вычислительной системой. Она является главным механизмом обработки информации, извлечения смысла и генерации нового знания.

Модель можно рассматривать с трёх позиций:

1. **С математической точки зрения** — как функцию или вероятностное распределение, отражающее взаимосвязь данных.
2. **С программной точки зрения** — как совокупность алгоритмов, принимающих данные, обрабатывающих их и возвращающих результат.
3. **С когнитивной точки зрения** — как упрощённую копию реальной системы или процесса.

В машинном обучении **применяется** несколько основных типов моделей:

- **Линейные модели** (линейная регрессия, логистическая регрессия) — простые и легко интерпретируемые.
- **Вероятностные модели** (Наивный Байес, марковские модели) — отражают вероятностную природу данных.
- **Деревообразные модели** (деревья решений, Random Forest, Gradient Boosting) — выполняют прогнозирование на основе разветвлённых структур.
- **Нейронные сети** (MLP, CNN, RNN, Transformer) — способны выявлять скрытые закономерности в изображениях и текстах, а также обрабатывать сложные структуры данных.

Все эти модели опираются на определённые математические конструкции и в своём исходном состоянии представляют собой лишь архитектуру, ожидающую обучения.

Пустая модель и её значение

Пустая модель — это архитектура, которая уже определена, но ещё не обучена на конкретных данных. Её можно рассматривать как исходную точку: структура существует, но опыта работы с данными у неё нет.

В качестве примера можно привести нейронную сеть. Архитектура сети задаётся количеством слоёв, числом нейронов в каждом слое,

ISSN 3030-3907

функциями активации. Однако весовые коэффициенты на начальном этапе принимают случайные значения. Такая сеть представляет собой пустую модель.

Основные преимущества пустой модели:

1. **Адаптивность** — пользователь может обучить её на данных, относящихся к необходимой предметной области.
2. **Лёгкость** — отсутствие избыточной информации позволяет экономить вычислительные ресурсы.
3. **Научная ценность** — исследователь получает полный контроль над процессом обучения.

Например, адаптация готовых англоязычных моделей, таких как GPT, к ограниченным корпусам специализированных казахских текстов сопряжена с трудностями. В то же время создание пустой модели и её обучение непосредственно на казахских данных является более эффективным и научно обоснованным подходом.

Оценка качества модели

Модель — это не только архитектура. Для определения её эффективности используются специальные метрики:

- **Accuracy** — доля правильных прогнозов;
- **Precision** и **Recall** — показатели, учитывающие различные виды ошибок;
- **F1-score** — гармоническое среднее между Precision и Recall;
- **Perplexity** — метрика для языковых моделей, оценивающая способность предсказывать текст.

Эти показатели позволяют количественно оценить, насколько эффективно обучена пустая модель.

Таким образом, модель машинного обучения является ключевым инструментом для выявления закономерностей в данных и прогнозирования новых результатов. А пустая модель — наиболее удобная отправная точка для адаптации под конкретные задачи. Она играет важную роль в создании интеллектуальных систем, ориентированных на национальные языки, а также в разработке специализированных моделей для узкопрофильных текстовых данных.

Используемые инструменты

- **Python** — универсальный язык с простым синтаксисом и широкой экосистемой библиотек;
- **Tkinter** — модуль Python для построения графического интерфейса пользователя;

ISSN 3030-3907

- **Scikit-learn** — библиотека с алгоритмами классификации, кластеризации и регрессии;
- **TensorFlow и PyTorch** — современные платформы для построения нейронных сетей;
- **python-docx** — инструмент для работы с документами Word;
- **Pickle и Joblib** — библиотеки для сохранения и загрузки обученных моделей.

Выбор архитектуры

Для работы с текстовыми моделями применяются следующие подходы:

- **Bag-of-Words (BoW)** — представление текста как набора слов;
- **TF-IDF** — оценка значимости слов в пределах документа и корпуса;
- **Нейронные сети** — использование многослойных архитектур для извлечения семантических особенностей текста.

Обучение модели

Процесс обучения включает несколько этапов: предварительную обработку данных, их преобразование в числовой формат, выбор алгоритма и оценку полученных результатов.

Предварительная обработка текстовых данных

Сырые текстовые данные не готовы к использованию в моделях. В них содержатся лишние символы, форматирование, пунктуация и стоп-слова. Очистка и стандартизация данных — важный этап, обеспечивающий качество обучения.

Основные шаги предварительной обработки:

1. Приведение текста к нижнему регистру, чтобы избежать дублирования слов («Казах» и «казах»).
2. Удаление избыточной пунктуации, если она не влияет на смысл.
3. Исключение стоп-слов («и», «как», «один», «о» и т. п.), не несущих семантической нагрузки.
4. Лемматизация — приведение слов к начальной форме (например, «пошёл», «идёт», «идти» → «идти»).

В результате текст становится более компактным и однородным, что повышает эффективность его обработки моделью.

Векторизация текста

Компьютер не способен напрямую работать с текстом, поэтому слова и предложения преобразуются в числовые векторы.

Наиболее распространённый метод — **TF-IDF (Term Frequency – Inverse Document Frequency)**. Он учитывает частоту появления слова в

ISSN 3030-3907

документе и снижает вес слишком часто встречающихся слов, которые мало влияют на определение темы.

Современные методы, такие как **Word2Vec, FastText, BERT**, позволяют моделям учитывать семантическую близость слов, располагая их векторные представления ближе друг к другу в многомерном пространстве.

Алгоритмы обучения

Для анализа векторизованных данных применяются различные алгоритмы:

- **Наивный Байес** — простой и эффективный метод классификации, хорошо работающий на малых выборках.
- **Логистическая регрессия** — классическая модель для бинарной и многоклассовой классификации.
- **Нейронные сети** — гибкие и универсальные модели, способные выявлять сложные зависимости.
- **Ансамблевые методы** (Random Forest, Gradient Boosting) — объединяют несколько «слабых» моделей, повышая итоговую точность.

Практический пример:

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.linear_model import LogisticRegression
import pickle
docs = ["Язык Python используется для рограммирования",
        "Футбол – популярный вид спорта",
        "Модель используется в области искусственного интеллекта"]
labels = [0, 1, 0]  # 0 - информатика, 1 - спорт
vectorizer = TfidfVectorizer()
X = vectorizer.fit_transform(docs)
model = LogisticRegression()
model.fit(X, labels)
with open("model.pkl", "wb") as f:
        pickle.dump((vectorizer, model), f)
```

Методы хранения и применения модели

- **Pickle** – используется для сериализации и десериализации объектов Python. Позволяет сохранить обученную модель в файл и впоследствии загрузить её без повторного обучения.
- **Joblib** – более эффективный инструмент для моделей, содержащих большие массивы данных или матрицы. Отличается высокой скоростью работы и меньшим объёмом сохраняемых файлов.

Повторное использование модели

ISSN 3030-3907

```
    with open("model.pkl", "rb") as f:
vectorizer, model = pickle.load(f)
text = ["язык програмирование Python"]
X_new = vectorizer.transform(text)
prediction = model.predict(X_new)
print("категория ответа:", prediction)
```

Приложение Tkinter

```
import tkinter as tk
def ask():
user_text = entry.get()
X = vectorizer.transform([user_text])
pred = model.predict(X)[0]
result.set("категория ответа: " + str(pred))
root = tk.Tk()
root.title("Модель «Вопрос–Ответ» ")
entry = tk.Entry(root, width=50)
entry.pack()
result = tk.StringVar()
tk.Label(root, textvariable=result).pack()
tk.Button(root, text="Получение ответа", command=ask).pack()
root.mainloop()
```

Результаты и обсуждение

В ходе экспериментального исследования была разработана модель на основе небольшого объёма текстовых данных. Эффективность модели напрямую зависела от качества предварительной обработки данных.

- При сохранении стоп-слов результаты модели оказывались неточными, что подтверждает важность их удаления.
- Метод TF-IDF продемонстрировал более высокую точность по сравнению с простой моделью «мешка слов» (BoW).
- Логистическая регрессия показала хорошие результаты при работе с ограниченными объёмами данных.

Опыт сохранения и повторного использования модели подтвердил её практическую значимость. Использование интерфейса Tkinter обеспечило удобную среду для взаимодействия с пользователем.

Заключение

В данной статье были рассмотрены методы подготовки, сохранения и обучения «пустой» модели на основе текстовых данных. В ходе исследования был проведён анализ понятия модели, выделены особенности пустой модели,

ISSN 3030-3907

а также рассмотрены этапы предварительной обработки данных, их преобразования в числовой формат и эффективность различных алгоритмов.

Основные выводы работы:

1. Язык программирования **Python** предоставляет достаточно инструментов для построения и обучения моделей. Библиотеки **scikit-learn, pandas, numpy** позволяют эффективно реализовать основные этапы машинного обучения.
2. Качество предварительной обработки текстовых данных напрямую влияет на точность модели. Удаление стоп-слов, лемматизация и векторизация текста значительно повышают производительность алгоритмов.
3. Возможность сохранения и повторного использования обученной модели повышает её универсальность и экономит время, что делает модель удобным инструментом для различных проектов.
4. Разработка простого приложения на основе библиотеки **Tkinter** продемонстрировала практическую ценность подхода: пользователи получили возможность взаимодействовать с моделью в формате чат-бота.

Результаты исследования показали перспективность обработки текстовых данных на казахском языке и выявили ряд нерешённых проблем.

Перспективы дальнейших исследований

- Формирование крупного корпуса текстов на казахском языке;
- Использование нейронных сетей, включая рекуррентные и трансформерные архитектуры;
- Разработка многоязычных моделей, способных одинаково эффективно работать с казахским, русским и английским языками;
- Интеграция чат-моделей в образование, государственные сервисы и информационные системы.

Таким образом, исследование методов подготовки и обучения пустой модели является не только научно значимым, но и практически полезным. Проведённая работа способствует развитию казахского языка в цифровом пространстве, созданию интеллектуальных систем и формированию инновационных решений на национальном языке.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. Géron, A. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly, 2022.
2. Bird, S., Klein, E., Loper, E. *Natural Language Processing with Python*. O'Reilly, 2009.
3. Raschka, S. *Machine Learning with Python*. Packt Publishing, 2015.
4. Chollet, F. *Deep Learning with Python*. Manning, 2021.
5. ГОСТ 7.1–2003. Библиографическая запись.

ISSN 3030-3907

6. Кабулов В.К., Кабулов А.В., Норматов И.Х. Алгоритмизация в теории управляющих систем//Монография: Ташкент-2017, Изд. “Навруз” С.176.
7. Кабулов В.К., Кабулов А.В., Норматов И.Х. Логические методы алгоритмизации в теории управляющих систем. Монография: Германия.2018, Изд. “Ламберт” С.191.
8. Норматов И.Х., Бабаджанов О.Ф., Кудайбергенов А. Исследование методов поиска максимальных совместных подсистем систем нелинейных булевых уравнений// Научный вестник НамГУ, № 2024-11, С. 81-85.
9. Normatov Ibroximali, Doschanova Aziza Xusainboy qizi, Abduqaxxorova Nigora Doniyor qizi Internet tarmogida videotasvirlarni himoya qilish model va usullari// Development of science Volume 3 Iyun 2025/6, C. 90-97, ISSN 3030-3907.

ISSN 3030-3907