

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский  
технический университет имени К.И.Сатпаева  
Институт автоматки и информационных технологий  
Кафедра кибербезопасность, обработка и хранение  
информации

Құлшықова Тогжан Абылайқызы

Применение классификационных методов при прогнозировании

**ДИПЛОМНАЯ РАБОТА**

Специальность 5В070300 – Информационные системы

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский  
технический университет имени К.И.Сатпаева  
Институт автоматизации и информационных технологий  
Кафедра кибербезопасность, обработка и хранение  
информации

**ДОПУЩЕН К ЗАЩИТЕ**

Заведующий  
кафедрой КОиХИ  
канд. техн. наук,  
ассоц. профессор  
 Р.Ж.Сатыбалдиева

« 19 »  
05 2022г.

**ДИПЛОМНАЯ РАБОТА**

На тему: Применение классификационных методов при прогнозировании

Специальность 5В070300 – Информационные системы

Выполнил: Кулшыкова Тогжан Абылайкызы

Рецензент

к.т.н., ассоц. проф.

Балгабаева Л.Ш. 

« 19 » 05 2022г.

Научный руководитель

сениор-лектор, к.т.н.

Байматаева Ш. М. 

« 19 » 05 2022г.

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН

Казахский национальный исследовательский  
технический университет имени К.И.Сатпаева  
Институт автоматики и информационных технологий  
Кафедра кибербезопасность, обработка и хранение  
информации  
5B070300 – Информационные системы

УТВЕРЖДАЮ

Заведующий кафедрой КОиХИ  
канд. техн. наук, ассоц.  
профессор

*Р.Ж. Сатыбалдиева* Р.Ж. Сатыбалдиева

« 19 » 05 2022г.

**ЗАДАНИЕ**  
**на выполнение дипломной работы**

Обучающемуся: Кулшыкова Тогжан Абылайкызы

Тема: Применение классификационных методов при прогнозировании  
Утверждена приказом Ректора Университета № 489-П/Ө от 24.12.2021.  
Срок сдачи законченной работы 26.05.2022г.

Исходные данные к дипломному проекту: Веб сайт прогноза погоды.  
Краткое содержание дипломной работы:

- а) анализ предметной области, разработка технического задания;
- б) исследование классификационных методов, используемых при прогнозировании;
- в) разработка программы.

Перечень графического материала (с точным указанием обязательных чертежей): представлены 21 слайдов презентации работы

Рекомендуемая основная литература: *из 12 наименований*

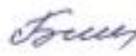
## ГРАФИК

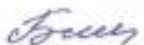
подготовки дипломной работы (проекта)

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
1. Анализ предметной области, разработка технического задания	28.02.2022	
2. Исследование классификационных методов, используемых при прогнозировании	7.04.2022	
3. Разработка программы	3.05.2022	

## Подписи

консультантов и нормоконтролера на законченную дипломную работу  
(проект) с указанием относящихся к ним разделов работы (проекта)

Наименование разделов	Консультанты, Ф.И.О. (уч.степень, звание)	Дата подписания	Подпись
Основная часть	Байматаева Ш.М. (к.т.н., сениор- лектор)	15.05.2022	
Нормоконтролер	Аристомбаева М.Т. (магистр техн.наук, лектор)	16.05.2022	

Научный руководитель:  Байматаева Ш. М.

Задание принял к исполнению обучающийся:  Кудшыкова Т. А.

Дата "26" 01 2022

## АННОТАЦИЯ

В дипломной работе рассмотрены вопросы применения классификационных методов на примере прогнозирования погоды. В качестве алгоритмов были выбраны метод ближайших соседей и наивный Байес. В результате обучения реальных данных с применением этих методов метод ближайших соседей оказался немного точнее по сравнению со вторым методом. Кроме того, выбраны средства разработки приложения; разработана архитектура работы бота; разработан интерфейс и Telegram Bot для прогнозирования погоды. С помощью Telegram Bot можно получить среднюю температуру, среднюю влажность, давление, состояние погоды (жаркая, дождливая).

Для реализации системы были использованы: Jupyter Notebook, Telegram Bot, BotFather.

## АНДАТПА

Дипломдық жұмыста ауа-райын болжау мысалында топтау әдістерін қолдану сұрақтары қарастырылған. Топтау тәсілдерінің алгоритмдері ретінде жақын көршілер әдісі мен аңғал Байес әдістері таңдалды. Бұл әдістерді аталған тәсілдерді қолданып шынайы мәліметтерде оқыту нәтижесінде жақын көршілер әдісі екінші әдіспен салыстырғанда дәлірек болды. Сондай ақ, қосымшаны әзірлеу құралдары таңдалды; боттың жұмыс істеу архитектурасы жасалды; ауа райын болжау үшін Telegram Bot әзірленді. Telegram Bot көмегімен температураны, орташа температураны, қысымды, ауа райы жағдайын (ыстық, жаңбырлы) алуға болады.

Жүйені іске асыру кезінде келесілер пайдаланылды: Jupyter Notebook, Telegram Bot, BotFather.

## ANNOTATION

The thesis discusses the application of classification methods on the example of weather forecasting. The nearest neighbor's method and naive Bayes were chosen as algorithms. As a result of learning real data using these methods, the nearest neighbor method turned out to be a little more accurate compared to the second method. In addition, the application development tools have been selected; the architecture of the bot has been developed; the interface and Telegram Bot for weather forecasting have been developed. With the help of Telegram Bot, you can get the average temperature, average humidity, pressure, weather conditions (hot, rainy).

To implement the system, Jupiter Notebook, Telegram Bot, BotFather were used.

## СОДЕРЖАНИЕ

Введение	9
1 Исследовательский раздел	10
1.1 Цели и задачи работы	10
1.2 Техническое задание	10
1.2.1 Применение классификационных методов при прогнозировании погоды	10
1.2.2 Основание для разработки	10
1.2.3 Назначение	10
1.2.4 Требования к функциональным характеристикам	10
1.2.5 Требования к надежности	11
1.2.6 Требования к составу и параметрам технических средств	11
1.2.7 Требования к информационной и программной совместимости	11
1.2.8 Требования к программной документации	11
1.3 Технологии Data Mining	11
1.4 Существующие системы. Яндекс погода	14
2 Технологический раздел	16
2.1 Используемые методы и средства разработки	16
2.1.1 Метод ближайших соседей	16
2.1.2 Наивный алгоритм Байеса	20
2.1.3 Jupyter Notebook	21
2.2 Боты в Telegram	21
2.2.1 BotFather	21
3 Проектная часть	23
3.1 Источник данных	23
3.2 Реализация прогноза	25
3.3 Реализация Telegram - бота	28
3.4 Выход/Результат	28
Заключение	32
Перечень сокращений	33
Литература	34
Приложение А	37

## ВВЕДЕНИЕ

Машинное обучение – это метод анализа данных, который позволяет искусственно созданной системе учиться на опыте и получать знания из заданного количества входных данных. Алгоритмы машинного обучения используют вычислительные методы для обучения непосредственно на входных данных, не полагаясь на заранее определенные уравнения. Алгоритмы машинного обучения адаптивно улучшают свою производительность по мере увеличения объема данных, доступных для обучения. В этой работе машинное обучение используется для реализации методов классификации для прогнозирования погоды с использованием реальных данных.

В работе реализуется Telegram Bot, который будет предоставлять детали, прогноз погоды, прогноз индекса качества воздуха и т. д. Набор данных для всей этой информации будет взят с csv файла. Бот предназначен для предоставления людям необходимую информацию о погоде. Основная проблема состоит в том, чтобы найти действительный набор данных для проекта. Также требуется очистить данные, чтобы обеспечить согласованность данных, чтобы убедиться, что аналитика не реализована на отсутствующие значения и удаление ненужных данных. Также важно удалить ненужные столбцы из набора данных, чтобы сократить время вычислений и данные на котором предстоит вычислить. Предполагается, что данные были собраны из сайта прогноза погоды обработан и сохранен в csv файл. Мы начинаем наш анализ и прогнозирование с этого набора данных.

Актуальность темы. Для общества в различных отраслях, а также для простых людей благоприятно знать прогноз погоды на месяцы и год. Этот метод прогнозирования, нацелены на прогноз погоды, которая влияет на людей. Реализация полностью точных прогнозов погоды, конечно, невозможно, ведь до сих пор нету таких методов или так скажем приборов для этого. В данной дипломной работе реализована в практику классификационные методы при прогнозировании.

## **1 Исследовательский раздел**

### **1.1 Цели и задачи работы**

Цель работы: реализовать программу для прогнозирования погоды с использованием классификационных методов машинного обучения.

Для достижения поставленной задачи были сформулированы следующие задачи:

- 1) Провести анализ классификационных методов, которые будут использованы при прогнозировании погодных явлений.
- 2) На основе анализа существующих моделей выбрать модель, позволяющую реализовать прогнозирование с использованием классификационных методов машинного обучения в оперативном режиме.
- 3) Реализовать алгоритм KNN и наивный Байес для использования в прогнозировании, оценить точность методов.
- 4) Разработать интерфейс и Telegram bot для прогнозирования погоды.

### **1.2 Техническое задание**

#### **1.2.1 Применение классификационных методов при прогнозировании погоды**

Техническое задание распространяется на разработку прогноза погоды. Создаем Telegram бот для прогноза погоды, ведь в данный момент и ситуацией в мире именно Telegram является для большинства людей платформой для социальной сети. С помощью разработанного приложения можно в любой получить сведения о погоде.

#### **1.2.2 Основание для разработки**

Система разрабатывается на основании дипломной работы.

#### **1.2.3 Назначение**

Платформа предназначена для получения сведений о погоде.

#### **1.2.4 Требования к функциональным характеристикам**

Список функций, которые приложение должно обеспечивать:

- отображение списка команд;
- отображение информации об командах;
- получение данных из csv;
- прогнозирование погоды;
- вывод данных об запрашиваемой команде.

#### **1.2.5 Требования к надежности**

Предусмотреть валидацию вводимых данных. Предусмотреть сбои и другие исключительные ситуации при работе пользователя с приложением. Обеспечить целостность данных.

#### **1.2.6 Требования к составу и параметрам технических средств**

Минимальные системные требования: 48 Мб свободной памяти; объем ОЗУ – 256 Мб и выше, Поддержка 4G – рекомендуется.

#### **1.2.7 Требования к информационной и программной совместимости**

Система предназначена для работы на любых устройствах с установленным Telegram и доступом в интернет.

#### **1.2.8 Требования к программной документации**

Тексты программ должны быть комментированы.

### **1.3 Технологии Data Mining**

Сбор данных — это процесс анализа данных, собранных и обобщенных различными способами для получения полезной информации, или процесс использования различных инструментов анализа данных для выявления закономерностей и взаимосвязей между различными данными, которые можно использовать для точных прогнозов. Это смесь статистики, математики и компьютерного программного обеспечения. В последние годы анализ данных получил широкое распространение в области вычислительной техники и в обществе в целом в связи с широкой доступностью больших объемов данных и острой необходимостью преобразования таких данных в полезную информацию и знания. Знания могут быть использованы для различных приложений, таких как медицина, управление производством и исследования.

Анализ данных может использоваться во всех областях, например, в области финансов и бизнеса он может прогнозировать прибыль или убыток, а также может использоваться в кредитных картах и ожидаемых рисках в этой области, и кто является ожидаемым клиентом, чтобы быть хорошим и кто плохой. также в области здравоохранения и прогнозирования возможных генетических заболеваний в будущем, таких как диабет, давление и болезни сердца. Его также можно использовать в сфере безопасности и в военной сфере, а также в погодных условиях.

Варианты использования прогнозов погоды бесчисленны, например, предупреждения о погоде; Он защищает и уменьшает материальные потери жизни и имущества. Прогнозы погоды, которые сосредоточены на знании температуры и вероятности осадков, поскольку повседневная деятельность может измениться из-за сильного дождя, снега, ветра или града, прогнозы погоды могут использоваться для планирования дат и мест проведения различных мероприятий в этих погодных условиях.

Все данные, собранные организациями, не будут служить никакой цели без раскрытия знаний. Анализ данных помогает организациям визуализировать набор данных, которые раньше могли быть невидимы. Какие бы идеи ни открывались, они приведут к четким решениям.

Анализ данных направлен на обнаружение и извлечение полезных шаблонов, и это полезная технология, которая стала важной в связи с быстрым развитием и распространением использования баз данных, а также использование анализа данных во всех областях облегчает возможность изучения наиболее важной информации. в базах данных. Используются для построения будущих прогнозов и изучения поведения и тенденций, что позволяет оценивать правильные решения. Он также дает ответы на многие вопросы за короткое время.

Анализ данных означает обнаружение знаний из данных, и состоит из итерационной последовательности следующих шагов:

а) data cleaning (очистка данных): сначала необходимо удалить пустые и ненужные данные;

б) data integration (интеграция данных): объединения и обработка источников данных;

г) data transformation (преобразование данных): используют для анализа, где преобразуют необработанные данные в подходящие формы путем суммирования;

з) knowledge representation (представление знаний): использование методов визуализации и представления для представления данных пользователю.

## Методы прогнозирования

Этот метод использует реальные переменные для прогнозирования неизвестных или будущих значений других переменных. Этот метод содержит следующие типы:

Классификация используется для решения многих проблем путем анализа набора данных и помещения их в форму категорий, которые можно использовать позже. прогнозировать будущие данные.

Анализ данных для прогнозирования дает возможность использовать многочисленные источники данных временных рядов, внутренних и внешних, которые теперь легко доступны для лиц, принимающих бизнес-решения, в действенных стратегиях, которые могут напрямую влиять на прибыльность. Современные технологии Data Mining используют информацию с целью автоматического поиска шаблонов (паттернов), характерных для каких-либо фрагментов неоднородных многомерных данных.

Data Mining – это объединения нескольких различных друг от друга методов обнаружения информации. Выбор метода отталкивается от типа имеющихся данных и от того, какую информацию вы хотите получить. Data Mining применяется везде, где есть какие-либо данные.

Одно из основных различий между традиционным анализом данных и анализом данных временных рядов заключается в том, что в последнем ключевую роль играет временной горизонт прогноза. Для справочных целей краткосрочные прогнозы определяются здесь как прогнозы на один-три года, а среднесрочные прогнозы определяются на период от 3 до 5 лет, а долгосрочные прогнозы определяются на период более 5 лет. Автор согласна с тем, что все, что превышает 10 лет, следует рассматривать как сценарий, а не как прогноз. Финансовые группы обычно контролируют процесс «планирования» для корпораций и предоставляют «то» число, которое компания независимо от потребностей и целей различных групп заинтересованных сторон, дифференциация процесса «планирования» от процесса «прогнозирования» имеет решающее значение.

Компаниям действительно нужен «план», к которому они стремятся. Бизнес-лидеры действительно должны нести ответственность за план. Но утверждать, что этот план на самом деле является «прогнозом», может быть катастрофическим. Планы есть то, что мы «чувствуем, что можем сделать», в то время как прогнозы являются математическими оценками того, что наиболее вероятно. Это не то же самое, но оба должны быть сохранены. На самом деле точность обоих следует отслеживать в течение длительного периода времени. Когда сообщил Уолл-стрит, точность важнее точности. Быть ближе к неправильному номеру не помогает.

## 1.4 Существующие системы. Яндекс погода

В настоящее время широко используется сервис «Яндекс. Погода», которая наглядно показывает прогноз погоды: с её помощью люди могут узнать, где, когда и какая будет погода на улице. Дополнительный виджет погоды в верхней части основного экрана можно настроить для отображения погоды в зависимости от текущего местоположения или местоположения, заданного вручную, также она показывает световой день в часах и минутах.

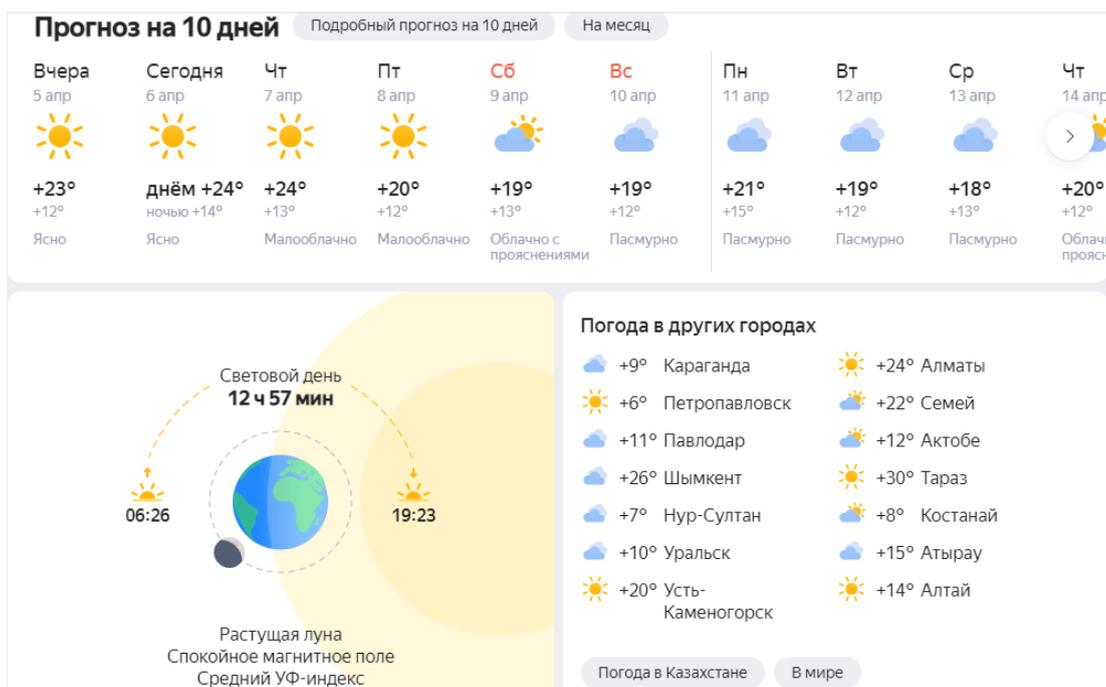


Рисунок 1 – Погода в Алматы

### Функции:

- карта сообщает об температуре, осадке, скорости ветра и давления. В частности, отслеживается влажность воздуха;
- «Яндекс. Погода» показывает прогноз осадков на 2 часа вперёд;
- благодаря актуальной карте погоды люди, к примеру, смогут определить, куда уехать на те даты, когда ожидается прекрасная погода.



Рисунок 2 – Карта мира

## 2 Технологический раздел

### 2.1 Используемые методы и средства разработки

#### 2.1.1 Метод ближайших соседей

Метод KNN представляет собой алгоритм машинного обучения, который считается простым методом, применяемым в многомерный анализ данных. Хотя этот метод прост, этот метод имеет преимущества по сравнению с другими методами, которые могут обобщать относительно небольшой набор обучающих данных.

Алгоритм ближайшего соседа K подпадает под категорию контролируемого обучения и используется для классификации (чаще всего) и регрессии. Это универсальный алгоритм, также используемый для вменения недостающих значений и повторной выборки наборов данных. Как следует из названия (K Ближайший сосед), он рассматривает K ближайших соседей (точек данных) для прогнозирования класса или непрерывного значения для новой точки данных. Метод k-ближайших соседей (k-NN) – это метод классификация объектов на основе данных обучения, которые находятся на самом близком расстоянии от объекта. Данные обучения прогнозируются в многомерное пространство, где каждое измерение представляет особенности данных. Это пространство разделено на разделы на основе классификации обучающих данных. Точка в этом пространстве помечается как класс c, если класс c является наиболее распространенная классификация по ближайшему k точки. Ближние или дальние соседи обычно рассчитываются на основе евклидовых расстояний, уравнения которых выглядят следующим образом.

$$X = ((x_1, x_2, \dots, x_n), A = (a_1, a_2, \dots, a_n) \sqrt{(\sum_{i=1}^n (x_i - a_i)^2)} \quad (1)$$

Обучение алгоритма:

а) обучение на основе экземпляров: здесь мы не изучаем веса из обучающих данных для прогнозирования вывода (как в алгоритмах на основе моделей), а используем целые обучающие экземпляры для прогнозирования вывода невидимых данных;

б) непараметрический: В KNN нет predetermined формы функции отображения.

Необходимая подготовка данных:

а) масштабирование данных: чтобы найти точку данных в многомерном пространстве объектов, было бы полезно, если бы все объекты были в одном масштабе. Следовательно, нормализация или стандартизация данных помогут;

б) уменьшение размерности: KNN может плохо работать, если слишком много функций. Следовательно, могут быть реализованы методы уменьшения размерности, такие как выбор функций, анализ главных компонент;

в) обработка недостающих значений: если из  $M$  объектов отсутствуют данные одного объекта для конкретного примера в обучающем наборе, мы не можем найти или вычислить расстояние от этой точки. Поэтому требуется удаление этой строки или вменения.

Классификатор KNN работает следующим образом:

- значение  $K$  инициализируется;
- рассчитывается расстояние между входной выборкой и обучающей выборкой:

$$\text{dist}((x, y), (a, b)) = \sqrt{(x - a)^2 + (y - b)^2} \quad (2)$$

- отсортировать расстояния;
- возьмем высокие  $K$ -ближайшие соседи;
- применить простое большинство;
- предсказать метки категорий с дополнительными соседями для входной выборки.

Алгоритм KNN на этапе обучения просто сохраняет набор данных, а когда он получает новые данные, он классифицирует эти данные в категорию, которая очень похожа на новые данные. На рис.1.1 показаны этапы предлагаемой системы. Прогноз погоды часто был замечательной областью исследований, потому что изменения климатических условий сильно влияют на людей, мы используем алгоритм  $k$ -ближайших соседей для прогноза погоды. Основная идея, на которой работает  $k$ -Nearest Neighbours, заключается в том, что информация связана в чрезвычайно функциональном пространстве. Поэтому для поиска места среди точек знаний все точки рассматриваются по порядку. Расстояние рассчитывается с использованием евклидова расстояния на основе типа данных используемых классов данных.

Задается одно значение  $K$ , и оно используется для нахождения общего числа ближайших соседей, определяющих метку класса для неизвестной выборки. Если значение  $K=1$ , то это называется классификацией ближайшего соседа. Предположим, что есть два класса, т. е. класс  $A$  и класс  $B$ , и что у нас есть совершенно новый  $x_1$ , поэтому эта информация может принадлежать этим классам чтобы решить эту проблему, нам нужен алгоритм KNN. С помощью KNN мы просто определим класс или категорию конкретного набора данных. На следующем рисунке показан алгоритм работы метода ближайших соседей применительно к системе:

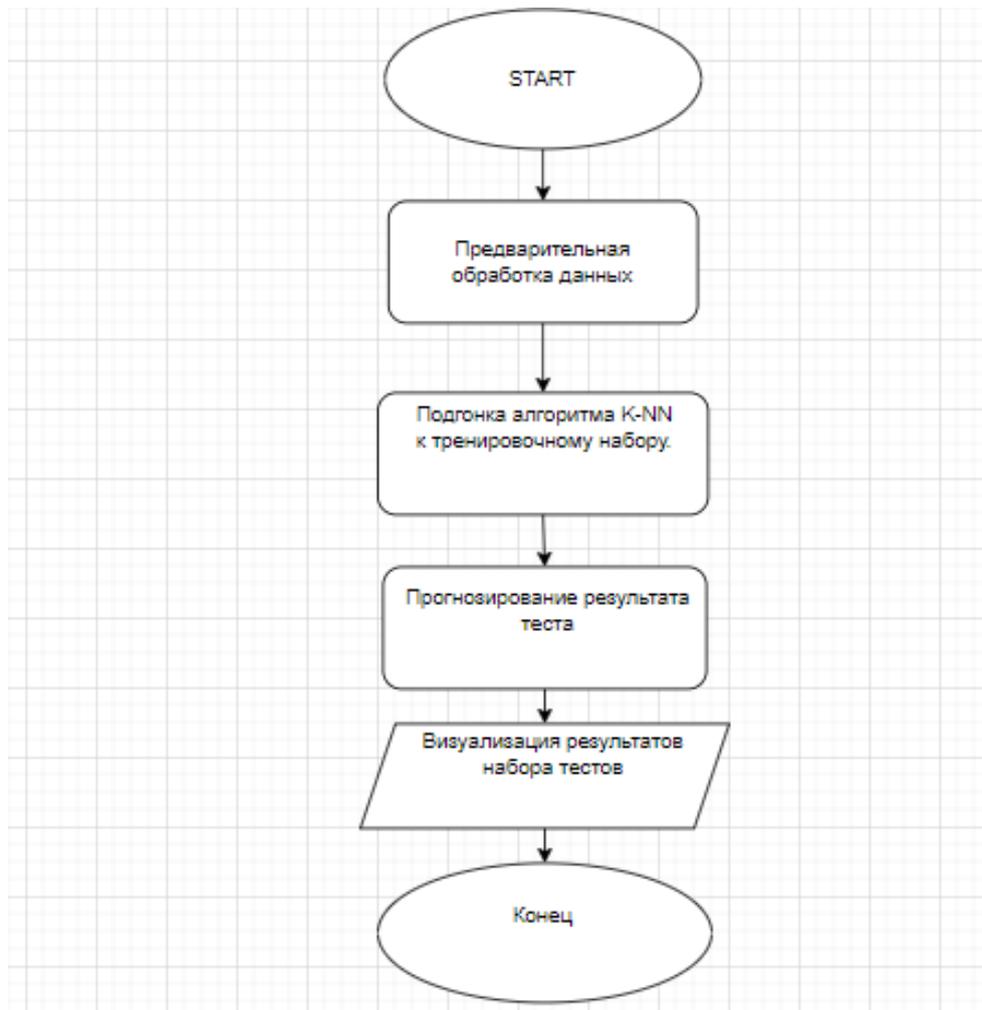


Рисунок 3 - Алгоритм работы метода ближайших соседей

### 2.1.2 Наивный алгоритм Байеса

Наивный байесовский алгоритм – вероятностный алгоритм машинного обучения, который можно использовать в самых разных задачах классификации. Наивная модель Байеса проста в построении и особенно полезна для очень больших наборов данных. Наряду с простотой, наивный Байес, как известно, лучше даже очень сложных методов классификации. Теорема Байеса вычисляет апостериорную вероятность  $P(c|x)$  из  $P(c)$ ,  $P(x)$  и  $P(x|c)$ :

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)} \quad (3)$$

$$P(c|x) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c) \quad (4)$$

- где,
- $P(x|c)$  – является вероятностью предиктора данного класса;
  - $P(c|x)$  – это апостериорная вероятность класса ( $c, target$ ) с заданным предиктором ( $x, attributes$ );
  - $P(c)$  – это априорная вероятность класса;
  - $P(x)$  – это априорная вероятность предиктора.

Плюсы:

- легко и быстро предсказать тестовые данные;
- когда речь идет об независимости, наивный байесовский классификатор работает лучше по сравнению с другими моделями и вам нужно меньше обучающих данных;
- он хорошо работает в случае категориальных входных переменных по сравнению с числовыми переменными. Для числовой переменной предполагается нормальное распределение (колоколообразная кривая, что является сильным предположением).

Минусы:

- наивный Байес также известен как плохой оценщик, поэтому вероятностные выходы из `predict_proba` не следует воспринимать слишком серьезно;
- еще одним ограничением наивного Байеса является предположение о независимых предикторах. В реальной жизни практически невозможно получить набор предикторов, которые полностью независимы.

Применения наивных байесовских алгоритмов

- предсказание в реальном времени: Naïve Bayes - это нетерпеливый классификатор обучения, и он уверен, что быстро. Таким образом, он может быть использован для прогнозирования в реальном времени;
- мультиклассовое предсказание: этот алгоритм также хорошо известен функцией мультиклассового предсказания. Здесь мы можем предсказать вероятность нескольких классов целевой переменной;
- классификация текста/ фильтрация спама/ анализ настроений: Наивные байесовские классификаторы, в основном используемые в классификации текста (из-за лучшего результата в многоклассовых задачах и правила независимости), имеют более высокий показатель успеха по сравнению с другими алгоритмами;

– система рекомендаций: Наивный Байесовский классификатор и совместная фильтрация вместе создают систему рекомендаций, которая использует методы машинного обучения и интеллектуального анализа данных для фильтрации невидимой информации и прогнозирования того, понравится ли пользователю данный ресурс или нет.

### **2.1.3 Jupyter Notebook**

Jupyter Notebook – веб-приложение содержащий живой код, уравнения, визуализации и текст, которое можно использовать для создания и обмена документами. Побочный проект проекта IPython, у которого раньше был собственный проект IPython Notebook.

Jupyter Notebook обладают тремя особенно важными преимуществами:

– он очень хорош для демонстрации вашей работы. Во время процесса вы сразу можете видеть код и результаты;

– так же вы можете запустить кусочек кода, чтобы лучше понять, что делает код;

– в целях безопасности, очень легко размещать серверную часть. Многие данные защищены и должны быть конфиденциальными, и одним из шагов к этому является отсутствие хранения данных на локальных машинах.

Код в Jupyter Notebooks обычно представляет собой код Python, хотя вы можете добавить в свою среду Jupyter поддержку других языков, таких как R или Julia. Данные могут быть предоставлены в отдельном файле вместе с файлом .ipynb, или могут быть импортированы программным способом.

## **2.2 Боты в Telegram**

Telegram - бот — это особый тип пользователя, который является не человеком, а компьютерной программой, которая может обслуживать компании или бренды с множеством функций, таких как отправка информации, напоминания, воспроизведение мелодий, заказ и многое другое. Бот может опубликовать сообщение в группе или в канале. Обычные пользователи могут подписаться на любого бота телеграммы, которого они хотели бы держать в курсе. Из всех четырех типов пиров бот Телеграм обладает множеством функций, которые действительно полезны для бизнеса. Телеграм предоставляет API для создания ботов для социального взаимодействия, производительности, игр и услуг электронной коммерции на платформе.

Боты бывают информаторами и ассистентами. Информаторы имеют задачу информировать пользователей бота, а ассистенты играют роль вспомогательной помощи, они консультируют пользователей по данным тематикам.

Стандартные команды, которыми можно пользоваться почти во всех ботах: /start – запуск бота, /settings – настройки бота, /help – инструкция по пользованию бота.

Бывают ситуации, когда бот вообще не отвечает ни на какие команды. Значит он не работает из-за ошибки в коде или ошибка на сервере.

Если рассматривать TelegramBot в контексте источника данных, фактические данные обычно остаются на локальных серверах источника данных. Там бот обращается к нужным наборам данных, а затем подготавливает их для вывода сообщения по запросу на конечное устройство. Все сообщения и, следовательно, общение с ботом защищены с помощью шифрования клиент-сервер/сервер-клиент. В облаке Телеграм эти данные также распределяются по ряду серверов и хранятся там в зашифрованном виде.

### 2.2.1 BotFather

Botfather – это, по сути, Телеграм - бот и реагирует на команды. Он помогает с созданием собственного бота, организует, например, уникальные имена ботов и назначает права доступа для программирования. Фактическое создание собственного бота, как правило, очень просто. Виртуальный пользователь «Botfather», который является центральным инструментом разработки ботов Телеграм. Вам сначала нужно найти этого пользователя в Телеграм и начать с ним чат. Выбрать проверенного (с синей галочкой) пользователя.

Для того чтобы начать процесс создания бота нужно сперва открыть мессенджер Telegram, войти в свою учетную запись.

- ввести @Botfather во вкладку поиска и выбрать этот бот;
- выбрать или ввести команду /newbot и отправить ее;
- выбрать имя для своего бота;
- перейти к боту @BotFather и отправить в конце команду /token.

Берем токен для работы с ботом посредством http-протокола. Токен это личная информация бота нельзя его показывать другим и желательно не терять. Надежнее скопировать его и сохранить в текстовом файле и положить в доступное место которая может быть надежным – например, облачное хранилище.

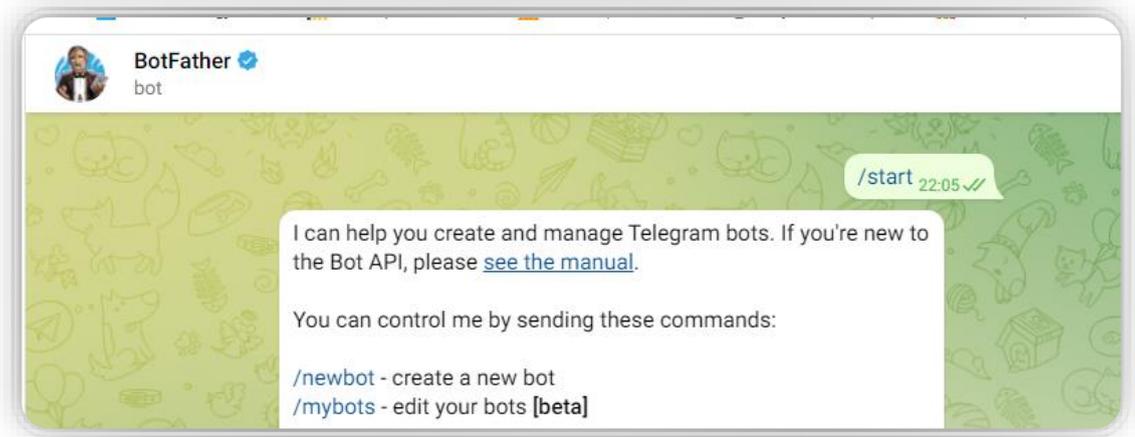


Рисунок 4 – BotFather

## 3 Проектная часть

### 3.1 Источник данных

Исходные данные для моделирования были получены с помощью метеорологической информации на сайте: [rogoda/almaty](http://rogoda/almaty) на рисунке 5 алгоритм работы метода ближайших соседей.

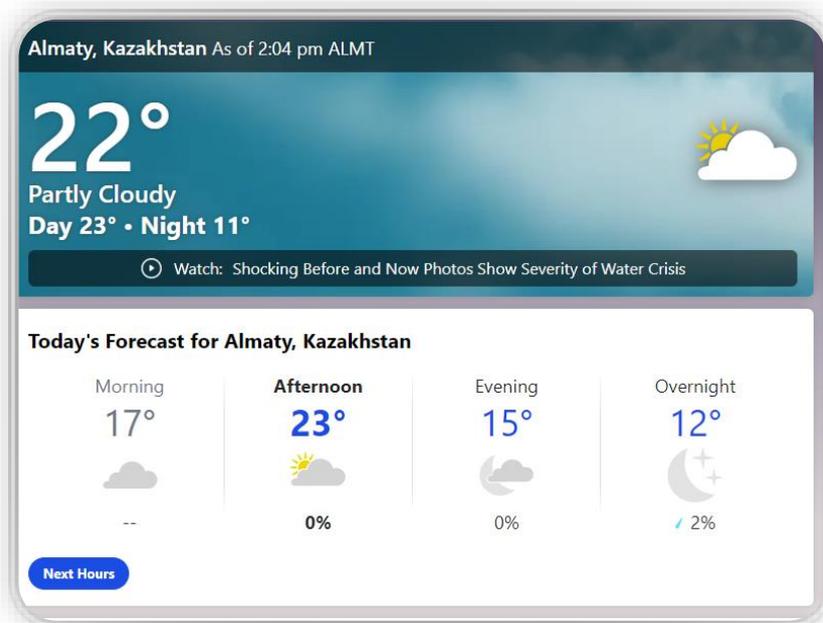


Рисунок 5 – Сайт прогноза погоды

В качестве входных данных полученные с помощью формирования входных данных из сайта. Далее на их основе производится моделирование сначала следует вручную проверить по какому индексу находится интересующая вас таблица. И после этого использовать в коде найденный индекс. Используя таким образом модуль Pandas можно легко, быстро решить задачу. На рисунке 6 показано схема процесса обработки данных.

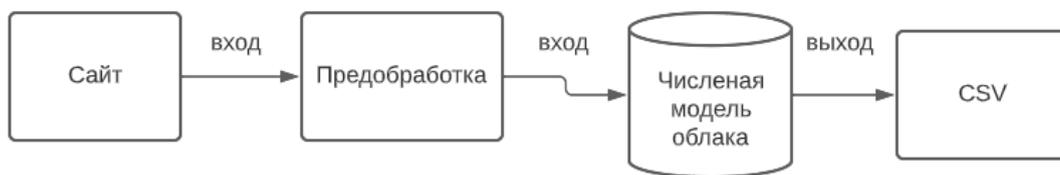


Рисунок 6 – Схема процесса обработки данных

На следующем рисунке показана диаграмма спецификации процесса:

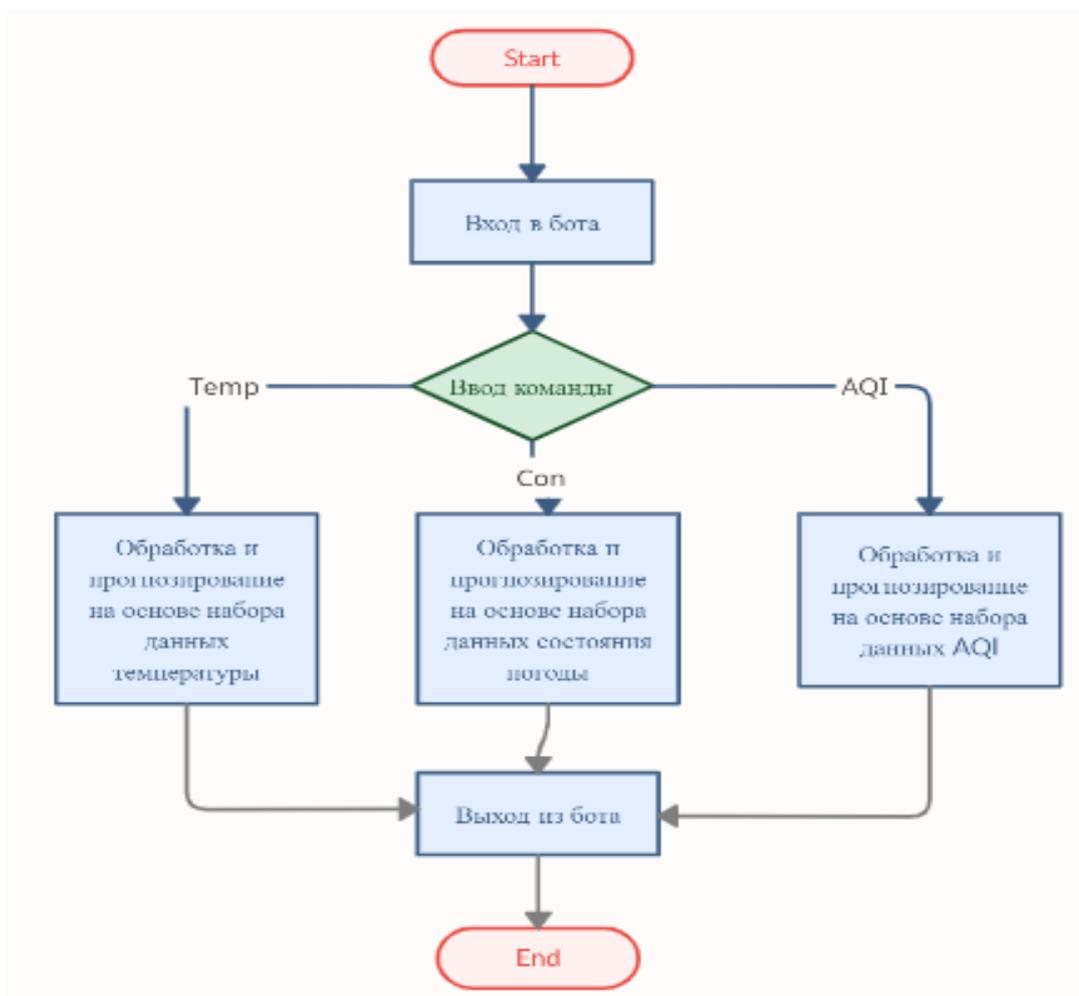


Рисунок 7 - Диаграмма спецификации процесса

Для того чтобы взять данные нам нужно запустить цикл `getweather()`. Берем определенные данные и присваиваем им соответствующие названия. Далее запускаем цикл, где создаем csv файл и сохраняем там данные.

```

    "Temperature" : temp,
    "Condition" : condition,
    "High/Low" : {}/{}.format(high,low),
    "Wind " :wind,
    "Humidity" : Humidity,
    "Dew Point" : dew,
    "Pressure" : pressure,
    "Visibility" : visibility,
    "Heat" : heat,
    "Wet" : wet,
    })
return weather

def files_writer(flats):
    #with open('HeadHunter.csv', 'a', encoding='utf-8') as file:
    with open(r"KNKZ#5.csv", "a", encoding = 'utf-8') as file:
        a_pen = csv.writer(file)
        a_pen.writerow('Day', 'Month','Year','Time','Temperature','Condition', 'High/Low', 'Wind', 'Humidity',
            for flat in flats:
                a_pen.writerow((flat['Day'],int(flat['Temperature'][0].replace(" ", '')),flat['Wind'],flat['Month'

flats = getweather()
files_writer(flats)

```

Status 200

Рисунок 8 – Сохранения данных в csv файл

Проверяем созданный csv файл на содержание:

```

[2]: df=pd.read_csv('data.csv')

[3]: df.head()

```

[3]:	Day	Month	Year	mean_temp	max_temp	min_temp	meanhum	meandew	pressure	heat	wet
0	1	8	1999	30.0	32.6	28.3	77	25.5	1002.1	YES	NO
1	2	8	1999	29.8	32.6	27.4	78	25.4	999.1	NO	NO
2	3	8	1999	29.7	31.4	28.3	80	25.8	998.4	NO	YES
3	4	8	1999	28.3	30.3	25.7	87	25.8	999.6	NO	YES
4	5	8	1999	28.4	32.6	25.3	82	24.9	999.6	NO	YES

Рисунок 9 - Фрагмент выходного файла CSV

Выходные данные представлены в формате CSV (Comma-Separated Values). Рассматривались следующие параметры:

- дата (день, месяц, год);
- средняя температура;
- максимальная температура;

- минимальная температура;
- средняя влажность;
- средняя точка росы;
- давление;
- жаркая пагода;
- дождливые.

Пример фрагмента выходного файла приведен на рис.9

### 3.2 Реализация прогноза

Реализация алгоритма K Nearest Neighbor с использованием библиотеки Python scikit-learn:

```
: import numpy as np
import pandas as pd
%matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report
from matplotlib.colors import ListedColormap
from sklearn.model_selection import train_test_split
from matplotlib.colors import ListedColormap
from sklearn import neighbors
import pandas as pd
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import confusion_matrix, accuracy_score
from sklearn.model_selection import cross_val_score
```

Рисунок 10 – Библиотеки Python scikit-learn

После загрузки важных библиотек мы создаем наши данные, используя `sklearn.datasets`. Затем данные разбиваются на обучающие (80%) и тестовые (20%).

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=1)
```

```
x_train.shape
```

```
(2727, 6)
```

```
x_test.shape
```

```
(1169, 6)
```

Рисунок 11 – Формы

Для выбора значения  $K$  мы используем кривые ошибок и значение  $K$  с оптимальной дисперсией, а ошибка смещения выбирается как значение  $K$  для целей прогнозирования. С кривой ошибки, построенной выше, мы выбираем  $K=12$  для прогнозирования.

```
[115]: <matplotlib.legend.Legend at 0x1ed174e0a60>
```

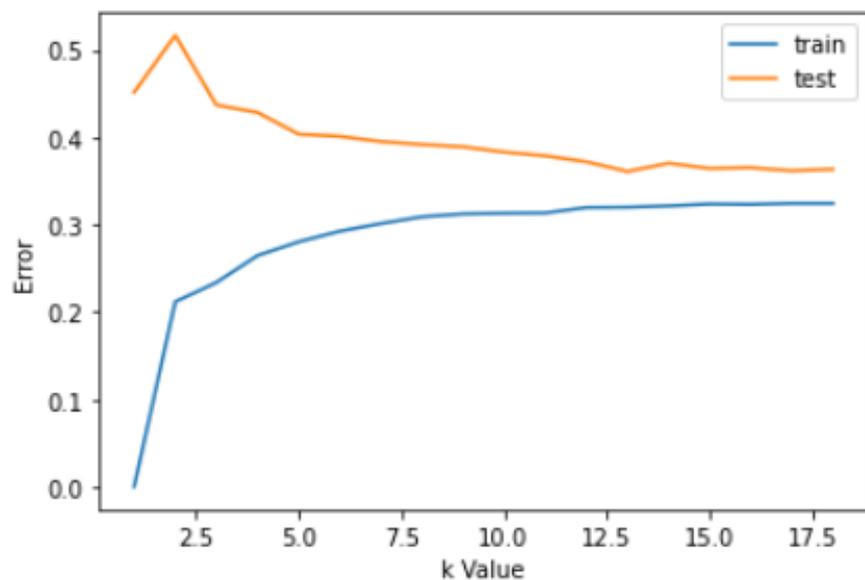


Рисунок 12 - Кривая ошибок для обучаемого и тестового набора

Дальше мы выбрали значение  $K$  равным 12. Теперь мы подставляем это значение и получаем показатель точности = 0,62 для тестовых данных.

```
[39]: accuracy = accuracy_score(y_test, y_predict6)*100
print('Точность нашей модели равна ' + str(round(accuracy, 2)) + ' %.')

Точность нашей модели равна 62.79 %.
```

```
[40]: test_acc_knncla = round(knncla.fit(x_train,y_train).score(x_test, y_test)* 100, 2)
train_acc_knncla = round(knncla.fit(x_train, y_train).score(x_train, y_train)* 100, 2)
```

```
[41]: model1 = pd.DataFrame({
    'Model': ['KNN'],
    'Train Score': [train_acc_knncla],
    'Test Score': [test_acc_knncla]
})
model1.sort_values(by='Test Score', ascending=False)
```

```
[41]:
```

	Model	Train Score	Test Score
0	KNN	68.02	62.79

Рисунок 13 - Показатель точности

Давайте попробуем получить показатель точности для Наивного алгоритма Байеса

```
[27]: from sklearn.naive_bayes import GaussianNB
nb=GaussianNB()
#training the model
nb.fit(x_train,y_train)
#testing the model
y_pred=nb.predict(x_test)
```

```
[28]: from sklearn import metrics

metrics.accuracy_score(y_test,y_pred)
```

```
[28]: 0.6107784431137725
```

Рисунок 14 – Показатель точности

Модель, обученная недостаточно хорошо, но он может использоваться для прогнозирования выходных данных при других новых входных данных:

```
predicted= model.predict([[0,27,22,23,66,1012]])  
print (predicted)
```

```
[10]
```

```
Classifier=GaussianNB()  
Classifier.fit(x,y)
```

```
GaussianNB()
```

```
Classifier.score(x,y)
```

```
0.6283367556468172
```

Рисунок 15 – Прогнозирование

### 3.3 Реализация Telegram - бота

Пользователь может взаимодействовать с Telegram Bot через мобильный телефон или компьютер.

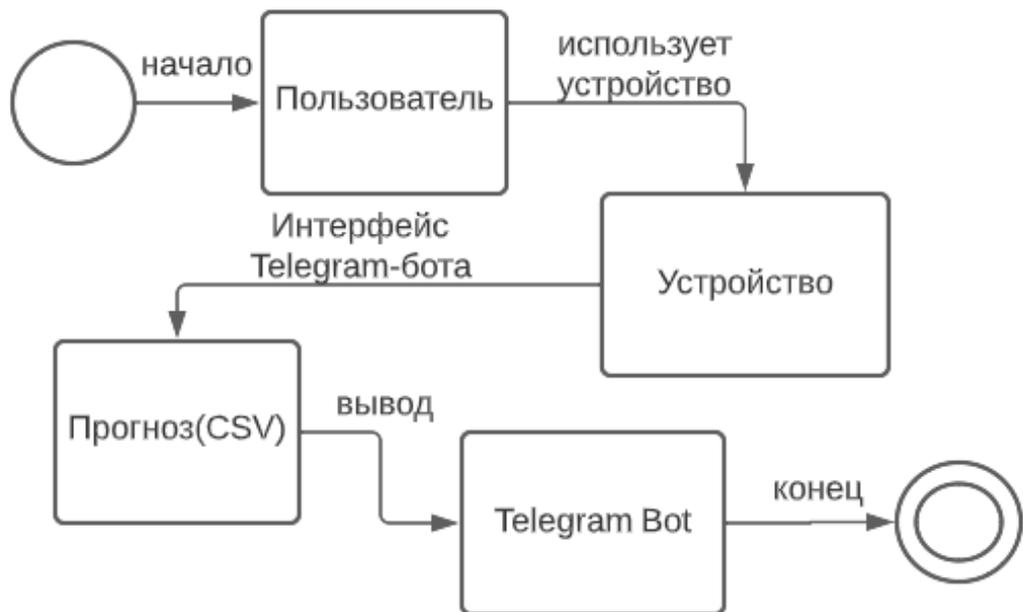


Рисунок 16 – Взаимодействие с системой

Загружаем библиотеку необходимую для Telegram Bot. Вводим API токен бота, который был взят с BotFather. Для каждого бота он выдает свой токен. Запускаем функцию def для температуры. В цикле мы берем сегодняшнюю дату через datetime.now() и сравниваем ее с датой в datasate.

```
[1]: import telebot
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from datetime import datetime
import datetime
import random

bot = telebot.TeleBot("5202215549:AAF-Je6K3FhDRd7c7yiQSSNGVmdcsfOLUi4", parse_mode=None)

def verify_temp():
    now = datetime.datetime.now()
    dateof=str(now)
    finalls=dateof.split(':')
    data=str(finalls[0])+':00:00'
    datalist=[]
    datalist[:0]=data
    data=''.join(datalist[0:])
    dataset = pd.read_csv('result1.csv')
    temp=dataset.iloc[:, 2].values
    y = dataset.iloc[:, 1].values
```

Рисунок 17 – Начало бота

```
def verify_con():
    now = datetime.datetime.today().strftime("%d.%m.%Y")
    dateof=str(now)
    finalls=dateof.split(':')
    data=str(finalls[0])+':00:00'
    datalist=[]
    datalist[:0]=data
    data=''.join(datalist[0:])
    dataset = pd.read_csv('data3.csv')
    con=dataset.iloc[:, 1].values
    y = dataset.iloc[:, 0].values

    for i in range(len(y)):
        #повернуть на сегодняшнюю дату
        change=list(y[i])
        change[8]='2'
        change[9]='2'
        y_result=''.join(change)

        if(dateof==y_result):
            return (con[i])
```

Рисунок 18 – Функция для состояния погоды

### 3.4 Выход/Результат

Message\_handler принимает параметр функции так же туда передается функции для каждого значения, принимает аргумент message (/start; /con; /temp; /humidity; /AQI), а возвращающая данные из csv файла.

Так же в jupyter notebook выводит данные которые запрашивает и выводит телеграм бот. Был проработан весь код и запущен. На следующих рисунках 19 и 20 показаны результаты моих работ.

```
@bot.message_handler(commands=['humidity', 'Humidity'])
def send_hum(message):
    print(verify_hum())
    h = str(verify_hum())+" %\n"
    bot.reply_to(message, h)

@bot.message_handler(commands=['water', 'Water'])
def send_water(message):
    bot.reply_to(message, str(verify_water()[1])+" mg/l in "+ str(verify_water()[0]))

@bot.message_handler(commands=['about'])
def send_about(message):
    bot.reply_to(message, "About us\nPredii Bot is here to help :)")

@bot.message_handler(func=lambda m: True)
def echo_all(message):
    bot.reply_to(message, "Ах, кажется, я не знаком с тем, что вы говорите. Попробуйте эти:\n/temp for Temperat

bot.polling()
```

Облачно  
None  
28.85  
UMIAM MAWPHLANG RIVER AT NONGKREM, EAST KHASI HILLS  
3.15

usy Mode: Command Ln 1, Col 1 tem 21:49 12.05.202

Рисунок 19 – Результат в Jupyter Notebook

Результаты моего прогноза в Telegram Bot:

- /start – запускает бота;
- /temp – выдает температуру;
- /con – выдает состояние погоды;
- /humidity – влажность;
- AQI – качество воздуха.

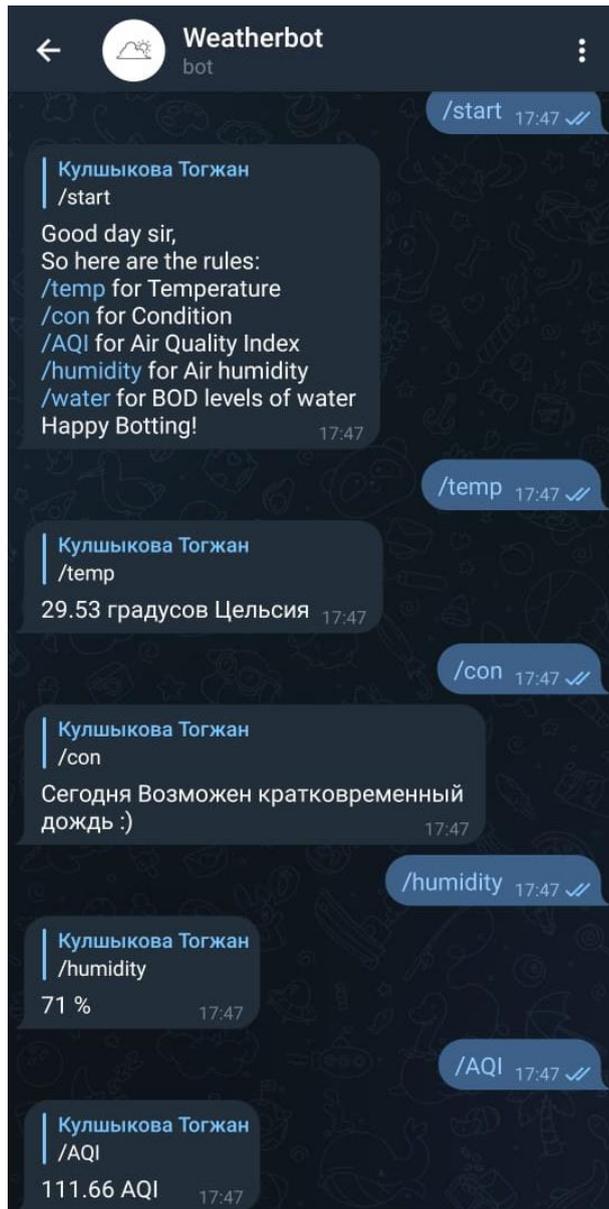


Рисунок 20 - Telegram Bot на 14.05.2022 г

## ЗАКЛЮЧЕНИЕ

В дипломной работе рассмотрены вопросы применения классификационных методов. Работа выбранных классификационных методов показана на примере прогнозирования погоды. В качестве алгоритмов были выбраны метод ближайших соседей и наивный Байес. В результате обучения реальных данных с применением этих методов метод ближайших соседей оказался немного точнее по сравнению со вторым методом. При выполнении работы получены следующие результаты:

- проведен обзор существующих систем прогнозирования;
- изучены классификационные методы, а именно, метод ближайших соседей и наивный Байес;
- проведено обучение данных с использованием указанных методов, оценена точность алгоритмов;
- разработан Telegram Bot для прогнозирования погоды.

С помощью Telegram Bot можно получить среднюю температуру, среднюю влажность, давление, состояние погоды (жаркая, дождливая).

Для реализации системы были использованы: Jupyter Notebook, Telegram Bot, BotFather.

Масштабы этого проекта могут быть расширены за счет включения разработки мобильных приложений, в котором приложение специально разработано для удовлетворения потребностей пользователя в более уединенный способ, в отличие от использования API бота для существующего приложения. Для целей демонстрации этого проекта использовались наборы данных, сравнительно меньше, что приводит к меньшей точности.

## ПЕРЕЧЕНЬ СОКРАЩЕНИЙ

В таблице А.1 сформулированы все термины и сокращения, которые используются в предметной области разрабатываемого проекта, а также специфические термины, связанные с программной реализацией проекта и используемыми технологиями при разработке.

**Таблица А.1 - Термины, сокращения, и их определения**

Сокращение или термин	Определение
Data Mining	Сбор данных
KNN	Алгоритм ближайшего соседа К подпадает под категорию контролируемого обучения и используется для классификации (чаще всего) и регрессии
k-Nearest Neighbours	к Ближайший сосед
Наивный алгоритм Байеса	Вероятностный алгоритм машинного обучения, который можно использовать в самых разных задачах классификации
Яндекс погода	«Яндекс погода» дает прогнозы тем пользователям, которые нуждаются в погодных условиях
°С	Градус Цельсия – единица температуры
Telegram Bot	Учетные записи Telegram, управляемые программным обеспечением
т.д.	– так далее
Temp	Температура
AQI	Качество воздуха
Water	Качество воды
Hum	Влажность воздуха
Con	Состояние погоды
CSV	Файл CSV – это особый вид файла, который позволяет структурировать большие объемы данных

## СПИСОК ЛИТЕРАТУРЫ

- 1 Introduction to Data Mining and Knowledge Discovery. Two Crowds Corporation, [Internet] 2009. Available from:  
<http://www.twocrows.com/introdm.pdf>
- 2 Shu-Hsien L, Pei-Hui C, Pei-Yuan H. Data mining techniques and applications – A decade review from 2000 to 2011. 2012.
- 3 Olaiya F, Barnabas A. Applications of Data Mining Techniques in weather prediction and climate change studies. MECS. 2012.
- 4 Folorunsho O. Application of Data Mining Techniques in Weather Prediction and Climate Change Studies. I.J. Information Engineering and Electronic Business. 2012; 1: 51-59.
- 5 Jayavani K. Statistical classification in machine intellegent. ISR Journals and Publications. 2014.
- 6 Fan C, Xiao F, Wang S. Development of prediction models for next day building energy consumption and peak power demand using data mining techniques. 2014.
- 7 Gupta D, Ghose U. A Comparative Study of Classification Algorithms for Forecasting Rainfall. IEEE. 2015.
- 8 Wang J, Su X. An improved K-Means clustering algorithm. IEEE. 2014. 13 Hand, Manilla, Smyth, Principles of Data Mining. The MIT Press. 2001.
- 9 Dunne S, Ghosh B. Weather adaptive traffic prediction using neuro-wavelet models. WCTR. 2013.
- 10 Witten IH, Frank E. Data Mining Practical Machine Learning Tools, and Techniques. 2015.
- 11 RamgenDeploy, How to Extract Data from ANY Website to CSV, YouTube, 2021.  
<https://youtu.be/-mZgTGkHjLU>.
- 12 Создание Python Telegram бота и его deploy на виртуальную машину // Электронная версия на сайте <https://habr.com/ru/post/549962/>.
- 13 Telegram-бот на Python за полчаса с aiogram // Электронная версия на сайте <https://tproger.ru/articles/telegram-bot-create-and-deploy/>.

## Приложение А

### Текст программы

```
/*Реализация кода для получения данных в csv файле*/
import pandas as pd
import requests
import numpy as np
from bs4 import BeautifulSoup as bs
import csv
from datetime import date

def getweather():
    weather = []
    url = "https://yandex.kz/pogoda/almaty?lat=43.238293&lon=76.945465"
    day = date.today().day
    weather.append(day)
    month = date.today().month
    weather.append(month)
    year = date.today().year
    weather.append(year)

    page = requests.get(url)

    soup = bs(page.content, 'html.parser')

    temp = soup.find("div", {"class": "today_nowcard-temp"}).text[:-1]

    time = soup.find("p", {"class": "today_nowcard-timestamp"}).text[6:]
    weather.append(time)
    weather.append(temp)
    condition = soup.find("div", class_="today_nowcard-phrase").text
    weather.append(condition)
    high = soup.find_all("span", class_="deg-hilo-nowcard")[0].text[:-1]

    low = soup.find_all("span", class_="deg-hilo-nowcard")[1].text[:-1]
    weather.append(high)
    weather.append(low)
    rightnow = soup.find("div", class_="today_nowcard-sidecar component
panel").find("table")
```

## Продолжение приложения А

```
wind = rightnow.find_all("tr")[0].find("span").text
weather.append(wind)
Humidity = rightnow.find_all("tr")[1].find("span").text[:-1]
weather.append(Humidity)
dew = rightnow.find_all("tr")[2].find("span").text[:-1]
weather.append(dew)
pressure = rightnow.find_all("tr")[3].find("span").text[:-3]
pressure = pressure.replace(',', '')
weather.append(pressure)
visibility = rightnow.find_all("tr")[4].find("span").text
weather.append(visibility)
if(int(temp)>=30):
    heat = "YES"
else:
    heat = "NO"
weather.append(heat)
if(int(Humidity)>=80):
    wet = "YES"
else:
    wet = "NO"
weather.append(wet)
jobs.append({
    "Day" : day,
    "Month" : month,
    "Year" : year,
    "Time" : time,
    "Temperature" : temp,
    "Condition" : condition,
    "High/Low" : { }/{} .format(high,low),
    "Wind " : wind,
    "Humidity" : Humidity,
    "Dew Point" : dew,
    "Pressure" : pressure,
    "Visibility" : visibility,
    "Heat" : heat,
    "Wet" : wet,
})
return weather
```

## Продолжение приложения А

```
def files_writer(flats):
    with open(r"data1.csv", "a", encoding = 'utf-8') as file:
        a_pen = csv.writer(file)
        a_pen.writerow('Day', 'Month', 'Year', 'Time', 'Temperature', 'Condition',
            'High/Low', 'Wind', 'Humidity', 'Dew Point', 'Pressure', 'Visibility', 'Heat', 'Wet')
        for flat in flats:
            a_pen.writerow((flat['Day'], int(flat['Temperature'][0].replace(" ",
                "")), flat['Wind'], flat['Month'], flat['Year'], flat['Time'], flat['Humidity'], flat['Dew
                Point'], flat['Visibility'], flat['Heat'], flat['Wet']))

    flats = getweather()
    files_writer(flats)

/*Реализация Telegram бота*/
import telebot
import numpy as np
from datetime import datetime
import datetime
import pandas as pd
import matplotlib.pyplot as plt
import random

bot = telebot.TeleBot("5202215549:AAF-
Je6K3FhDRd7c7yiQSSNGVmdcsfOLUi4", parse_mode=None)

def verify_temp():
    now = datetime.datetime.now()
    dateof=str(now)
    finalls=dateof.split(':')
    data=str(finalls[0])+':00:00'
    datalist=[]
    datalist[:0]=data
    data=".join(datalist[0:])
    dataset = pd.read_csv('result1.csv')
    temp=dataset.iloc[:, 2].values
    y = dataset.iloc[:, 1].values
    for i in range(len(y)):
        #повернуть на сегодняшнюю дату
        change=list(y[i])
        change[2]='2'
        change[3]='2'
```

## Продолжение приложения А

```
y_result=".join(change)
```

```
if(data==y_result):  
    return round(temp[i],2)
```

```
def verify_con():  
    now = datetime.datetime.today().strftime("%d.%m.%Y")  
    dateof=str(now)  
    dataset = pd.read_csv('data3.csv')  
    con=dataset.iloc[:, 1].values  
    y = dataset.iloc[:, 0].values
```

```
for i in range(len(y)):  
    #повернуть на сегодняшнюю дату  
    change=list(y[i])  
    change[8]='2'  
    change[9]='2'  
    y_result=".join(change)
```

```
if(dateof==y_result):  
    return (con[i])
```

```
def verify_hum():  
    now = datetime.datetime.today().strftime("%d.%m.%Y")  
    dateof=str(now)  
    dataset = pd.read_csv('data3.csv')  
    hum=dataset.iloc[:, 10].values  
    y = dataset.iloc[:, 0].values
```

```
for i in range(len(y)):  
    #повернуть на сегодняшнюю дату  
    change=list(y[i])  
    change[8]='2'  
    change[9]='2'  
    y_result=".join(change)
```

```
if(dateof==y_result):  
    return round(hum[i])
```

## Продолжение приложения А

```
def verify_aqi():
    now = datetime.datetime.now()
    dateof=str(now)
    finalls=dateof.split(':')
    data=str(finalls[0])+':00:00'
    datalist=[]
    datalist[0]=data
    data="".join(datalist[0:])
    dataset = pd.read_csv('air_result.csv')
    aqi=dataset.iloc[:, 2].values
    y = dataset.iloc[:, 1].values

    for i in range(len(y)):
        #повернуть на сегодняшнюю дату
        change=list(y[i])
        change[2]='2'
        change[3]='2'
        y_result="".join(change)

        if(data==y_result):
            return round(aqi[i], 2)

@bot.message_handler(commands=['start','help'])
def send_welcome(message):
    bot.reply_to(message,"Good day sir,\nSo here are the rules: \n/n/temp for
Temperature\n/con for Condition\n/AQI for Air Quality Index\n/humidity for Air
humidity\nHappy Botting!")

@bot.message_handler(commands=['temp','Temp'])
def send_temp(message):
    bot.reply_to(message, str(verify_temp())+" градусов Цельсия")

@bot.message_handler(commands=['aqi','AQI'])
def send_aqi(message):
    print(verify_aqi())
    m = str(verify_aqi())+" AQI\n"
    bot.reply_to(message, m)
```

## Продолжение приложения А

```
@bot.message_handler(commands=['con','Condition'])
def send_hum(message):
    print(verify_con())
    c = "Сегодня " + str(verify_con())+" :)\n"
    bot.reply_to(message, c)
```

```
@bot.message_handler(commands=['humidity','Humidity'])
def send_hum(message):
    print(verify_hum())
    h = str(verify_hum())+" %\n"
    bot.reply_to(message, h)
```

```
@bot.message_handler(func=lambda m: True)
def echo_all(message):
    bot.reply_to(message, "Ах, кажется, я не знаком с тем, что вы говорите.
    Попробуйте эти:\n/temp for Temperature\n/con for Condition\n/AQI for Air Quality
    Index\n/humidity for Air humidity\nHappy Botting!")
```

```
bot.polling()
```

```
@bot.message_handler(commands=['temp','Temp'])
def send_temp(message):
    bot.reply_to(message, (verify_temp()," degrees celcius"))
```

```
@bot.message_handler(commands=['aqi','AQI'])
def send_aqi(message):
    verify_aqi()
    m = (verify_aqi()," AQI\n")
    bot.reply_to(message, m)
```

```
@bot.message_handler(commands=['about'])
def send_about(message):
    bot.reply_to(message, " О нас\nPredii Bot всегда готов помочь:")
```

```
bot.polling()
```

**ОТЗЫВ**  
на дипломную работу

Обучающийся:  
Кудшыкова Тогжан Абылайкызы

Тема дипломной работы: «Применение классификационных методов при прогнозировании»

Дипломная работа посвящена вопросам применения классификационных методов на примере прогнозирования погоды.

В ходе выполнения работы был проведен обзор существующих систем, с помощью которых можно получить сведения о погоде. Были изучены методы классификации, на основе которых будет разрабатываться программное приложение. Для этого рассмотрены 2 метода, было проведено сравнение их прогнозирующей способности. Кроме того были разработаны:

- архитектура программного приложения;
- интерфейс программы;
- телеграм бот для прогнозирования погоды.

Набор данных, задействованный в программе был собран обучающимся самостоятельно и является реальным. Для реализации системы были использованы: Jupyter Notebook, Telegram Bot, BotFather.

В ходе выполнения работы Кудшыкова Т. А. показала хорошую теоретическую подготовку, способность работать современными программными средствами. Все разделы дипломной работы выполнены автором самостоятельно. Результатом является разработка вполне функционирующего мобильного приложения.

Считаю, что дипломная работа Кудшыкова Тогжан Абылайкызы может быть допущена к защите.

Научный руководитель,  
сеньор-лектор, к.т.н.



Байматаева Ш.М.



## Метаданные

Название

диплом\_кулшыкова\_тогжан 1.docx

Автор

Кулшыкова Тогжан Научный руководитель

Подразделение

ИАИИТ

## Список возможных попыток манипуляций с текстом

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв		21
Интервалы		0
Микропробелы		0
Белые знаки		0
Парафразы (SmartMarks)		32

## Объем найденных подоби

Обратите внимание! Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



25

Длина фразы для коэффициента подобия 2



5192

Количество слов



39425

Количество символов

## Подобия по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и обратите внимание на то, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

### 10 самых длинных фраз

Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	<a href="https://xn--80ad5aze.xn--p1ai/raznoe/telegramm-boty-dlya-biznesa-telegram-boty-dlya-biznesa-10-besplatnyh-pomoshnikov.html">https://xn--80ad5aze.xn--p1ai/raznoe/telegramm-boty-dlya-biznesa-telegram-boty-dlya-biznesa-10-besplatnyh-pomoshnikov.html</a>	81	1.56 %
2	Разработка системы Автоматизированное рабочее место – «Тренер» спортивной информационной системы 4/27/2018 Satbayev University (ИКиИТ)	36	0.69 %
3	<a href="https://ichi.pro/ru/intuitivnoe-rukovodstvo-po-naivnomu-bajesovskomu-klassifikatoru-39424879690124">https://ichi.pro/ru/intuitivnoe-rukovodstvo-po-naivnomu-bajesovskomu-klassifikatoru-39424879690124</a>	29	0.56 %

4	Разработка алгоритма машинного обучения для современных систем управления процессами КАП 12/12/2019 Satbayev University (ИКиИТ)	28	0.54 %
5	диплом Ким Александра КНУ 5 мая.docx 5/6/2022 Kazakh-Deutsch University (Факультет инжиниринга и информационных технологий)	25	0.48 %
6	<a href="http://www.statproject.ru/publ/analiz_i_obrabotka_dannykh_osnovnye_zadachi/dobycha_dannykh_data_mining/novye_dostizhenija_statistica_data_miner_prognozirovanie_pogody/26-1-0-45">http://www.statproject.ru/publ/analiz_i_obrabotka_dannykh_osnovnye_zadachi/dobycha_dannykh_data_mining/novye_dostizhenija_statistica_data_miner_prognozirovanie_pogody/26-1-0-45</a>	22	0.42 %
7	<a href="https://studwork.org/shop/27648-kalkulyator">https://studwork.org/shop/27648-kalkulyator</a>	21	0.40 %
8	<a href="https://161.ru/text/gorod/2018/04/25/58232771/">https://161.ru/text/gorod/2018/04/25/58232771/</a>	19	0.37 %
9	<a href="https://russianblogs.com/article/892597912/">https://russianblogs.com/article/892597912/</a>	17	0.33 %
10	<a href="http://www.statproject.ru/publ/analiz_i_obrabotka_dannykh_osnovnye_zadachi/dobycha_dannykh_data_mining/novye_dostizhenija_statistica_data_miner_prognozirovanie_pogody/26-1-0-45">http://www.statproject.ru/publ/analiz_i_obrabotka_dannykh_osnovnye_zadachi/dobycha_dannykh_data_mining/novye_dostizhenija_statistica_data_miner_prognozirovanie_pogody/26-1-0-45</a>	16	0.31 %

#### из базы данных RefBooks (0.00 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	----------	---

#### из домашней базы данных (1.83 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	Разработка алгоритма машинного обучения для современных систем управления процессами КАП 12/12/2019 Satbayev University (ИКиИТ)	59 (4)	1.14 %
2	Разработка системы Автоматизированное рабочее место – «Тренер» спортивной информационной системы 4/27/2018 Satbayev University (ИКиИТ)	36 (1)	0.69 %

#### из программы обмена базами данных (1.75 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	Разработка Telegram Bot для кафе.docx 5/21/2021 Turan University (TU) (Б - Информационные системы)	46 (4)	0.89 %
2	диплом Ким Александра КНУ 5 мая.docx 5/6/2022 Kazakh-Deutsch University (Факультет инжиниринга и информационных технологий)	45 (3)	0.87 %

#### из интернета (6.01 %)

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	--------------	---

1	<a href="https://xn--80ad5aze.xn--p1ai/raznoe/telegram-boty-dlya-biznesa-telegram-boty-dlya-biznesa-10-besplatnyh-pomoshnikov.html">https://xn--80ad5aze.xn--p1ai/raznoe/telegram-boty-dlya-biznesa-telegram-boty-dlya-biznesa-10-besplatnyh-pomoshnikov.html</a>	81 (1)	1.56 %
2	<a href="https://ichi.pro/ru/intuitivnoe-rukovodstvo-po-naivnomu-bajesovskomu-klassifikatoru-39424879690124">https://ichi.pro/ru/intuitivnoe-rukovodstvo-po-naivnomu-bajesovskomu-klassifikatoru-39424879690124</a>	46 (3)	0.89 %
3	<a href="http://www.statproject.ru/publ/analiz_i_obrabotka_dannykh_osnovnye_zadachi/dobycha_dannykh_data_mining/novye_dostizhenija_statistika_data_miner_prognozirovanie_pogody/26-1-0-45">http://www.statproject.ru/publ/analiz_i_obrabotka_dannykh_osnovnye_zadachi/dobycha_dannykh_data_mining/novye_dostizhenija_statistika_data_miner_prognozirovanie_pogody/26-1-0-45</a>	38 (2)	0.73 %
4	<a href="https://studwork.org/shop/27648-kalkulyator">https://studwork.org/shop/27648-kalkulyator</a>	30 (2)	0.58 %
5	<a href="https://github.com/kokovych/bohdan_weather_bot/blob/master/weather_channel.py">https://github.com/kokovych/bohdan_weather_bot/blob/master/weather_channel.py</a>	25 (3)	0.48 %
6	<a href="https://161.ru/text/gorod/2018/04/25/58232771/">https://161.ru/text/gorod/2018/04/25/58232771/</a>	19 (1)	0.37 %
7	<a href="https://russianblogs.com/article/892597912/">https://russianblogs.com/article/892597912/</a>	17 (1)	0.33 %
8	<a href="https://coderlessons.com/tutorials/python-technologies/uznaite-mashinnoe-obuchenie-s-python/mashinnoe-obuchenie-s-python-kratkoe-rukovodstvo">https://coderlessons.com/tutorials/python-technologies/uznaite-mashinnoe-obuchenie-s-python/mashinnoe-obuchenie-s-python-kratkoe-rukovodstvo</a>	16 (3)	0.31 %
9	<a href="https://sourceforge.net/p/pytelegrambotapi/wiki/Home/">https://sourceforge.net/p/pytelegrambotapi/wiki/Home/</a>	15 (1)	0.29 %
10	<a href="http://elib.timacad.ru/dl/full/sbagro-2021-122.pdf/download?lang=en">http://elib.timacad.ru/dl/full/sbagro-2021-122.pdf/download?lang=en</a>	14 (1)	0.27 %
11	<a href="https://gist.github.com/ash17791/823d96343daa11e4399ba7d2788334e6">https://gist.github.com/ash17791/823d96343daa11e4399ba7d2788334e6</a>	11 (1)	0.21 %

### Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)
------------------	------------	---

## РЕЦЕНЗИЯ

на дипломную работу студента 4 курса Казахского национального  
исследовательского технического университета имени К.И. Сатпаева  
специальности 5B070300 - «Информационные системы»

Кулшыковой Тогжан Абылайкызы

на тему: «Применение классификационных методов при прогнозировании»

Рецензируемая дипломная работа Кулшыковой Т.А. посвящена применению классификационных методов на примере прогнозирования погоды.

Структура дипломной работы включает в себя: введение, три раздела, заключение, перечень сокращений, список литературы и приложение.

Во введении определяется актуальность выбранной темы, описательный образ телеграмм – бота.

В дипломной работе приведен анализ классификационных методов прогнозирования. В качестве алгоритмов были выбраны метод ближайших соседей и наивный Байес. В результате обучения реальных данных с применением этих методов метод ближайших соседей оказался немного точнее по сравнению со вторым методом. Кроме того, выбраны средства разработки приложения; разработана архитектура работы бота; разработан интерфейс и Telegram Bot для прогнозирования погоды. С помощью Telegram Bot можно получить информацию по средней температуре, средней влажности, давлению, состоянию погоды (жаркая, дождливая). Для реализации системы были использованы Jupyter Notebook, Telegram Bot, BotFather.

Прогноз погоды всегда был и будет актуальным для людей, которые имеют доступ в телеграм, в особенности сейчас, когда он становится популярным, так же особенностью этой работы является то, что бот имеет прямую связь с пользователем. Масштабы этого проекта могут быть расширены за счет включения разработки мобильных приложений.

Цель и задачи дипломной работы полностью соответствуют теме, и описаны в ходе реализации дипломной работы.

Учитывая актуальность и практическую ценность работы, считаю, что работа Кулшыковой Т. заслуживает оценки « 90% - 277 », а автор при успешной защите достойна присвоения академической степени бакалавра по специальности 5B070300 - «Информационные системы».

Рецензент  
канд. техн. наук,  
ассоц. профессор АТУ



Балгабаева Л. Ш.

" 16 " май 2022 г.

**Протокол анализа Отчета подобия  
заведующего кафедрой / начальника структурного  
подразделения**

Заведующий кафедрой / начальник структурного подразделения заявляет, что ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Құлшықова Тогжан Абылайқызы  
**Название:** Применение классификационных методов при прогнозировании  
**Координатор:** Шолпан Байматаева  
**Коэффициент подобия 1:** 9,59  
**Коэффициент подобия 2:** 3,83  
**Замена букв:** 21  
**Интервалы:** 0  
**Микропробелы:** 0  
**Белые знаки:** 0  
**Парафразы:** 32

После анализа отчета подобия заведующий кафедрой / начальник структурного подразделения констатирует следующее:

- обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, работа признается самостоятельной и допускается к защите;
- обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;
- обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, работа не допускается к защите.

Обоснование:

- обнаруженные в работе заимствования являются добросовестными

---

---

Заведующий кафедрой КоиХИ  
канд. техн. наук, ассоц.  
профессор

  
Р.Ж. Сатыбалдиева  
Подпись Заведующего кафедрой  
«    »    2022 г.  
Дата

## Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

**Автор:** Құлшықова Тоғжан Абылайқызы

**Название:** Применение классификационных методов при прогнозировании

**Координатор:** Шолпан Байматаева

**Коэффициент подобия 1:** 9,59

**Коэффициент подобия 2:** 3,83

**Замена букв:** 21

**Интервалы:** 0

**Микропробелы:** 0

**Белые знаки:** 0

**Парафразы:** 32

### После анализа Отчета подобия констатирую следующее:

обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

- обнаруженные в работе заимствования являются добросовестными

Научный руководитель  
сениор-лектор, к.т.н.



Байматаева Ш.М.  
Подпись Научного руководителя  
« 16 » мая 2022 г.  
Дата