

УДК 004.89

https://doi.org/10.33619/2414-2948/115/24

РАЗРАБОТКА И ПРОЕКТИРОВАНИЕ ЭКСПЕРТНОЙ СИСТЕМЫ ДЛЯ АНАЛИЗА КОТИРОВОК АКЦИЙ

©*Окулов М. Д.*, ORCID: 0009-0006-9243-6574, SPIN-код: 8379-2487, Национальный исследовательский университет «МЭИ», г. Москва, Россия, maxim999555@yandex.ru

©*Денисенко В. К.*, Национальный исследовательский университет «МЭИ», г. Москва, Россия, lubaok@bk.ru

DEVELOPMENT AND DESIGN OF AN EXPERT SYSTEM FOR STOCK PRICE ANALYSIS

©*Okulov M.*, ORCID: 0009-0006-9243-6574, SPIN-code: 8379-2487, National Research University Moscow Power Engineering Institute, Moscow, Russia, maxim999555@yandex.ru

©*Denisenko V.*, National Research University Moscow Power Engineering Institute, Moscow, Russia, lubaok@bk.ru

Аннотация. Рассматривается разработка экспертной системы для анализа фондового рынка с использованием технологий искусственного интеллекта. Представлена архитектура системы, включающая базу знаний, механизм логического вывода и пользовательский интерфейс. Описаны ключевые этапы разработки: от сбора и предобработки финансовых данных до реализации алгоритмов анализа на Python с применением библиотек NumPy, Pandas и методов машинного обучения. Особое внимание уделено анализу существующих решений (WallStreetZen, SeekingAlpha) и выбору инструментария (VS Code, QT Designer). Приведены примеры реализации основных компонентов системы, включая обработку временных рядов котировок акций и визуализацию результатов. Разработанная система демонстрирует эффективность в решении задач прогнозирования и может быть полезна участникам финансового рынка.

Abstract. Discusses the development of an expert system for stock market analysis using artificial intelligence technologies. The architecture of the system is presented, which includes a knowledge base, a logical inference mechanism, and a user interface. The key stages of development are described: from the collection and preprocessing of financial data to the implementation of analysis algorithms in Python using NumPy, Pandas libraries and machine learning methods. Special attention is paid to the analysis of existing solutions (WallStreetZen, SeekingAlpha) and the choice of tools (VS Code, QT Designer). Examples of the implementation of the main components of the system are given, including the processing of time series of stock quotes and visualization of results. The developed system demonstrates its effectiveness in solving forecasting problems and can be useful to financial market participants.

Ключевые слова: экспертная система, линейная регрессия, прогноз, акции.

Keywords: expert system, linear regression, forecast, stocks.

Экспертная система — система, основанная на знаниях, где одним из направлений в области исследования является искусственный интеллект, направленный на создание вычислительной системы, который может принимать значение как эксперт. Этапы разработки ЭС включают в себе шесть шагов: идентификация знаний, концептуализация

(знания собираются в базу данных: таблицу, графики), формализация (перевод языка), реализация (создание программного продукта), тестирование (испытания), опытная эксплуатация [1].

Типичная ЭС состоит из следующих основных компонентов: решателя (интерпретатора), рабочей памяти (РП), называемой также базой данных (БД), базы знаний (БЗ), компонентов приобретения знаний, объяснительного и диалогового компонентов (Рисунок 1).



Рисунок 1. Типовая структура экспертной системы

База данных предназначена для хранения исходных и промежуточных данных решаемой в текущий момент задачи. Этот термин совпадает по названию, но не по смыслу с термином, применяемым в информационно-поисковых системах (ИПС) и системах управления базами данных (СУБД) для обозначения всех данных (и в первую очередь - не текущих, а долгосрочных), хранимых в системе. Решатель, используя исходные данные из РП и знания из БЗ, формирует такую последовательность правил, которые, будучи примененными к исходным данным, приводят к решению задачи. Объяснительный компонент объясняет, как система получила решение задачи (или почему она не получила решения) и какие знания она при этом использовала, это облегчает эксперту тестирование системы и повышает доверие пользователя к полученному результату. Диалоговый компонент ориентирован на организацию дружелюбного общения со всеми категориями пользователей как в ходе решения задач, так и приобретения знаний, объяснения результатов работы.

В разработке ЭС участвуют представители следующих специальностей: эксперт в той проблемной области, задачи которой будет решать ЭС; инженер по знаниям — специалист по разработке ЭС; программист — специалист по разработке инструментальных средств.

Экспертная система работает в двух режимах: приобретения знаний и решения задач (называемым также режимом консультации или режимом использования ЭС). Характеристиками ЭС являются цель, проблемная область, глубина анализа проблемной области, используемые методы и типы знаний, класс системы, стадия существования, инструменты. Обычно выделяют следующие типы задач: интерпретация символов или сигналов — составление смыслового описания по входным данным; диагностика — определение неисправностей; предсказание — определение последствий наблюдаемых ситуаций; конструирование — разработка объекта с заданными свойствами при соблюдении установленных ограничений; планирование — определение последовательности действий, приводящих к желаемому состоянию объекта; слежение — наблюдение за изменяющимся

состоянием объекта и сравнение его показателей с установленными или желаемыми; управление — воздействие на объект для достижения желаемого поведения.

Анализ существующих аналогов: WallStreetZen, SeekingAlpha. Одной из ведущих платформ для биржевых опросов является WallStreetZen — это программное обеспечение для анализа запасов объединяет последние финансовые данные и автоматически выполняет подробный фундаментальный анализ, а также предоставляет простое описание в 1 строку, чтобы вы могли за считанные минуты узнать точный метод, используемый при расчете любых запасов. WallStreetZen позволяет инвесторам отслеживать свои инвестиции, настраивать уведомления об изменении цен и получать торговые рекомендации. Платформа также предоставляет учебные материалы и руководства по инвестированию, которые помогают пользователям расширить свои финансовые знания и навыки (Рисунок 2).

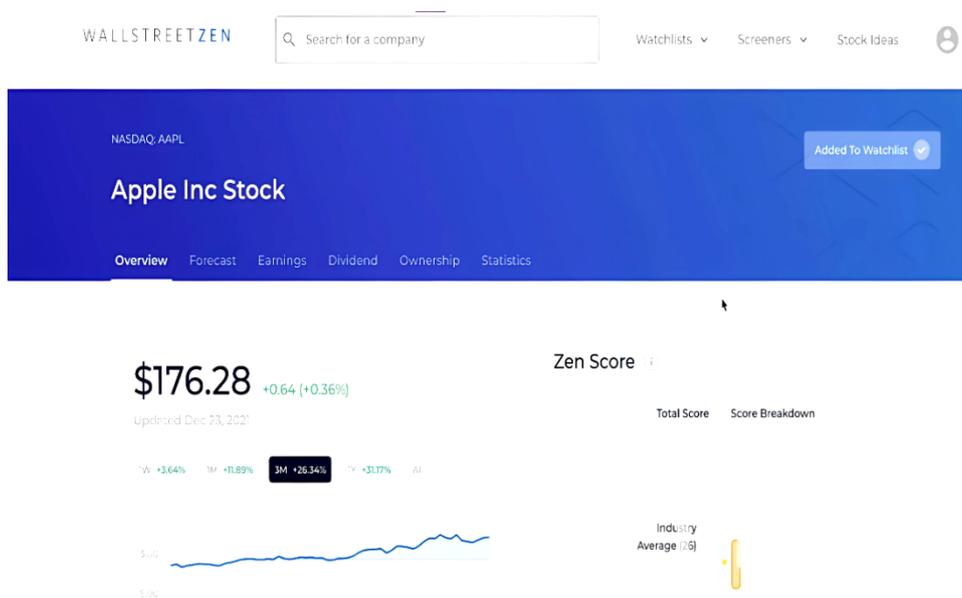


Рисунок 2. Домашняя страница (WallStreetZen)

Seeking Alpha — это онлайн-платформа для финансового анализа, инвестиций и торговли акциями. Сайт предлагает широкий спектр финансовой информации, новостей, аналитических материалов и обзоров компаний для инвесторов и трейдеров (Рисунок 3). Seeking Alpha также предлагает инструменты анализа акций и портфеля ценных бумаг.

Анализ инструментов и методов для проектирования экспертной системы. VS Code (Visual Studio Code) — один из самых популярных текстовых редакторов, разработанный Microsoft. Он предназначен для написания кода, отладки, рефакторинга и работы с различными технологиями и языками программирования. Область применения VS Code: редактор имеет встроенную подсветку синтаксиса, автозавершение, отладчик и множество расширений, что делает его мощным инструментом для создания программного обеспечения. Он часто используется для создания веб-сайтов и веб-приложений. Его возможности включают поддержку HTML, CSS, JavaScript, фреймворков, библиотек для веб-разработки и инструментов для работы с Git и другими системами контроля версий. Редактор можно использовать для создания скриптов на таких языках, как PowerShell, Bash и других, для настройки и автоматизации процессов управления операционной системой и сетью. Он популярен среди специалистов в области анализа данных и машинного обучения. Существуют расширения для работы с такими языками, как R, Python, и инструменты для

визуализации данных. Он также интегрируется с Docker, позволяя разработчикам управлять контейнерами и манипулировать композициями Docker непосредственно из редактора.

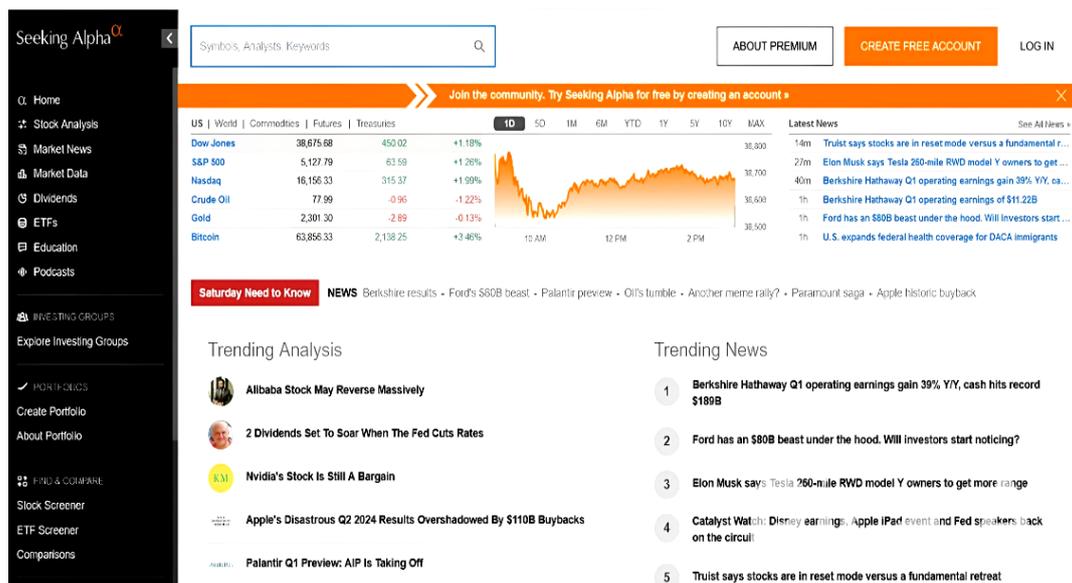


Рисунок 3. Домашняя страница (Seeking Alpha)

Python с использованием Designer. «Конструктор» (например, «Qt Designer») — это инструмент, который позволяет визуально создавать интерфейс для вашего приложения с использованием графических элементов. Обычно он используется в сочетании с библиотекой Qt. Процесс создания графического интерфейса (GUI) на Python с использованием Designer (например, Qt Designer или какого-либо другого инструмента) заключается в следующем. Как только интерфейс будет готов, сохраните его в соответствующем формате, который вы сможете загрузить и использовать на Python. Загрузите сохраненный дизайн интерфейса в скрипт на Python и используйте его для создания функционального приложения. Для этого вам понадобится библиотека для работы с графическим интерфейсом на Python, такая как PyQt или PySide. Используйте Python для определения действий, которые вам необходимо выполнять при взаимодействии с элементами интерфейса. После создания приложения запустите тесты, чтобы убедиться, что все элементы взаимодействуют правильно и функциональность соответствует вашим ожиданиям

Bash — это терминал для операционной системы Linux, который также может быть установлен в Windows в качестве аналога командной строки (cmd). В обоих интерфейсах оболочки пользователь может использовать команды для управления файлами и каталогами в операционной системе Cmd с помощью команд: dir (для отображения списка файлов и каталогов), cd (для изменения текущего каталога), del (в Bash:ls (для отображения списка файлов и каталогов), и так далее. Просмотр), cd (изменить текущий каталог), rm (удалить файлы).

Линейная регрессия — это метод анализа данных, который предсказывает ценность неизвестных данных с помощью другого связанного и известного значения данных. Он математически моделирует неизвестную или зависимую переменную и известную или независимую переменную в виде линейного уравнения [2].

Для регрессии общая прогнозная формула линейной модели выглядит следующим образом:

$$\hat{y} = w[0] * x[0] + w[1] * x[1] + \dots + w[p] * x[p] + b \quad (1)$$

Здесь $x[0]$ по $x[p]$ обозначают признаки (в данном примере число характеристик равно $p+1$) для отдельной точки данных, w и b – параметры модели, оцениваемые в ходе обучения, и \hat{y} — прогноз выдаваемый моделью. Для набора данных с одним признаком эта формула имеет вид:

$$\hat{y} = w[0] * x[0] + b \quad (2)$$

Линейные модели для регрессии можно охарактеризовать как регрессионные модели, в которых прогнозом является прямая линия для одного признака, плоскость, когда используем два признака, или гиперплоскость для большего количества измерений (рис. 4-5).

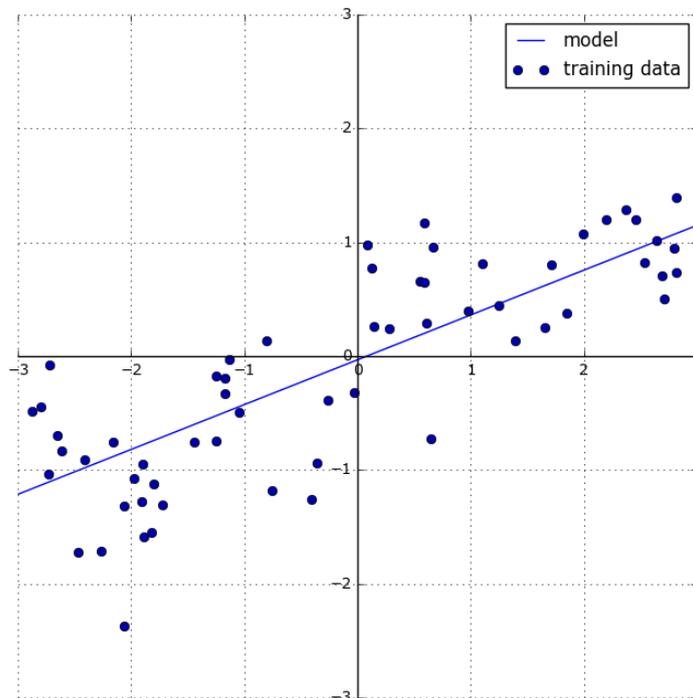


Рисунок 4. Пример-1 Линейная регрессия

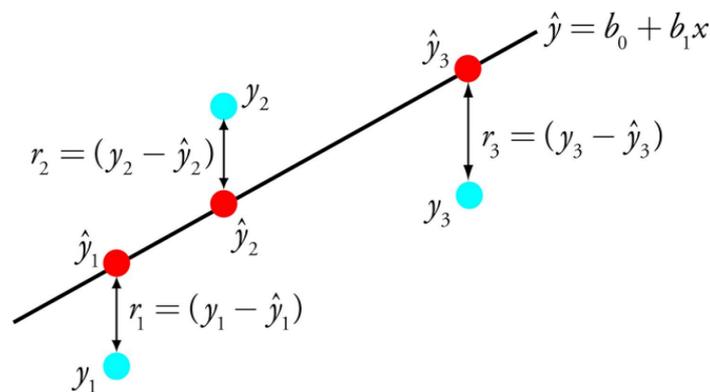


Рисунок 5. Пример-2 Линейная регрессия

NumPy (Numerical Python) — это библиотека для языка программирования Python, предназначенная для работы с многомерными массивами, линейной алгеброй, случайными числами и другими математическими вычислениями. NumPy предоставляет мощные инструменты для работы с массивами и эффективных операций над ними, что делает ее незаменимым инструментом для научных вычислений и анализа данных.

Seaborn — это библиотека для визуализации данных в Python, которая строит на базе библиотеки Matplotlib и предоставляет простой и красивый интерфейс для создания информативных графиков. Seaborn удобно работать с данными в формате pandas DataFrame и позволяет создавать сложные графики с минимальным объемом кода, включая столбчатые диаграммы, гистограммы, ящики с усами, тепловые карты, scatter plot и многое другое.

Pandas — это высокоуровневая библиотека для анализа данных, написанная на языке программирования Python, обладающая структурами данных и инструментами, необходимыми для эффективной работы с данными. Pandas предоставляет удобные и гибкие структуры данных, такие как Series (одномерные массивы) и DataFrame (двумерные таблицы данных), которые позволяют проводить операции на данных, являющиеся стандартными для работы с таблицами данных.

DataFrame — это основная структура данных в Pandas, представляющая двумерную таблицу данных со столбцами, которая может быть легко и эффективно обработана, проанализирована. DataFrame позволяет хранить и манипулировать данными различных типов (числовые, строковые, логические и прочие). Поддерживает множество операций над данными, такие как фильтрация, сортировка, группировка, агрегация, преобразование и многое другое. Это делает анализ и обработку данных более эффективной.

PyKnow — это библиотека на языке Python, которая предоставляет инструменты для разработки экспертных систем, такие как системы диагностики, системы принятия решений, системы рекомендаций и др. Основные компоненты PyKnow включают в себя факты, правила и движок (inference engine). Факты представляются в виде объектов Python, описывающих данные или знания о проблеме. Правила определяют, какие выводы можно сделать на основе определенных фактов. Движок отвечает за выполнение правил и вывод результатов.

Использование инструментов VS Code, Python с Disigner, Bash, библиотек numpy, seaborn, pandas, spyknow и метода линейной регрессии позволяет создать мощную экспертную систему, способную анализировать данные, принимать решения и предсказывать результаты на основе имеющейся информации [3]. Комбинация этих инструментов и методов обеспечивает возможность эффективной работы с данными, автоматизации процессов, визуализации информации и создания правил экспертной системы.

Разработка и проектирование экспертной системы для анализа котировок акций. Сбор финансовых данных. Онлайн-платформа для финансового анализа, инвестиций и торговли акциями предоставляет пользователям доступ к различным инструментам и ресурсам, которые помогают им принимать информированные решения о своих инвестициях. Можно проводить анализ финансовых показателей компаний, изучать отчеты о доходах и убытках, балансы и другие финансовые отчеты для принятия решений об инвестировании. Платформа обычно обеспечивает доступ к торговым рынкам, позволяя пользователям покупать и продавать акции и другие активы онлайн. Данная платформа предоставляет образовательные ресурсы, аналитические отчеты, рекомендации и инструменты для помощи пользователям в развитии их финансовых знаний и навыков (Рисунок 6).

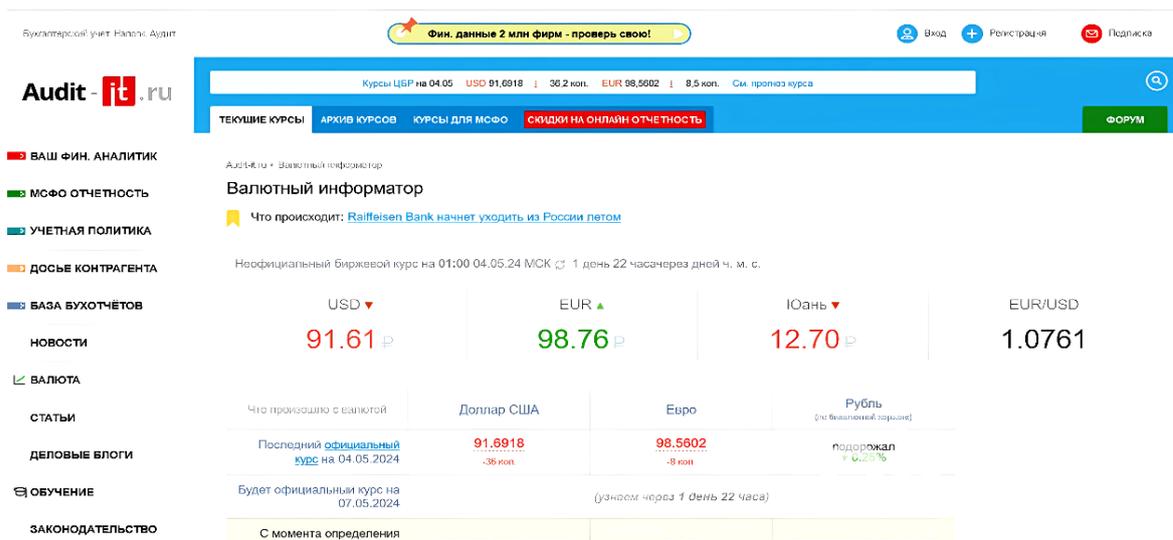


Рисунок 6. Онлайн платформа

```
import time as tp
import numpy as np
import pandas as pd
import seaborn as sns
import datetime as DT
from datetime import datetime
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score
```

Рисунок 7. Используемые библиотеки

NumPy (`import numpy as np`) — это библиотека для языка программирования Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых математических функций для операций с этими массивами (Рисунок 7).

Pandas (`import pandas as pd`) — это библиотека Python, предоставляющая высокоуровневые структуры данных и методы, предназначенные для быстрого и простого анализа данных.

Seaborn (`import seaborn as sns`) — это библиотека визуализации данных Python, основанная на Matplotlib, предоставляющая более высокоуровневый интерфейс для создания красочных и информативных статистических графиков.

Datetime — это библиотека, которая предоставляет функции для получения текущих даты и времени, создания объектов даты и времени, а также для выполнения математических операций с датами и временем.

Matplotlib (`import matplotlib.pyplot as plt`) — это библиотека визуализации данных для Python, которая позволяет создавать графики и диаграммы различных типов.

`train_test_split` — это функция из Scikit-learn (библиотеки для машинного обучения), которая позволяет разделить набор данных на обучающую и тестовую выборки.

`LinearRegression` — класс из Scikit-learn, предоставляющий функционал для линейной регрессии.

`RandomForestClassifier` — класс из Scikit-learn, предоставляющий функционал для случайного леса.

Импорт метрик для оценки качества модели: среднеквадратическая ошибка (MSE), средняя абсолютная ошибка (MAE) и коэффициент детерминации (R^2).

	<TICKER>	<PER>	<DATE>	<TIME>	<OPEN>	<HIGH>	<LOW>	<CLOSE>	<VOL>	<ADD_DATE>	
1	0	AAPL	60	240322	90000	171.73	172.31	170.06	171.03	453477	330322
2	1	AAPL	60	240322	100000	171.07	172.55	170.88	171.89	499830	340322
3	2	AAPL	60	240322	110000	171.91	172.87	171.83	172.6	303823	350322
4	3	AAPL	60	240322	120000	172.59	172.89	172.09	172.65	281765	360322
5	4	AAPL	60	240322	130000	172.64	172.94	172.365	172.66	233644	370322
6	5	AAPL	60	240322	140000	172.66	172.93	172.37	172.74	257026	380322
7	6	AAPL	60	240322	150000	172.75	173.05	172.23	172.25	545453	390322
8	7	AAPL	60	240325	90000	170.54	170.61	169.45	170.17	420156	330325
9	8	AAPL	60	240325	100000	170.18	170.61	169.77	170.545	314146	340325
10	9	AAPL	60	240325	110000	170.54	170.76	170.09	170.62	192314	350325
11	10	AAPL	60	240325	120000	170.62	171.14	170.39	170.99	193530	360325
12	11	AAPL	60	240325	130000	171.0	171.51	170.99	171.46	194252	370325
13	12	AAPL	60	240325	140000	171.47	171.6	171.24	171.45	144019	380325
14	13	AAPL	60	240325	150000	171.45	171.93	170.795	170.85	506663	390325
15	14	AAPL	60	240326	90000	170.0	170.82	169.66	170.74	284312	330326
16	15	AAPL	60	240326	100000	170.74	171.06	170.37	170.65	350746	340326
17	16	AAPL	60	240326	110000	170.65	170.86	170.44	170.57	198265	350326

Рисунок 8. Файлы и формат

На данном этапе `data["<DATE>"]` и `data["<TIME>"]` свернули в один формат, поэтому видна корреляция между временем, котировками акций (Рисунок 9).

```
for i in Name_File_Promotion:
    data = pd.read_csv(directory + Name_File_Promotion[i])
    print(data[:4])
    new_data = data["<DATE>"] + data["<TIME>"]
    data["<ADD_DATE>"] = new_data
    csv_data = data.drop(columns=["Unnamed: 0", "Unnamed: 0.1", "Unnamed: 0.1.1", "Unnamed: 0.1.1.1"])
    csv_data.to_csv(directory + Name_File_Promotion[i])
```

Рисунок 9. Листинг

Паттерное программирование. В разработке экспертной системы использовались такие классы как (Рисунок 10). Class `Select_Graphics ()` — это концепция создания класса (или объекта) в программировании, который позволяет выбирать различные виды графиков для отображения данных. Этот класс может содержать методы и свойства, которые позволяют пользователю выбирать типы графиков, стиль отображения, цвета, масштабы и другие параметры.

Class `Kategori(Fact)`: — это концепция создания класса (или объекта) в программировании, который представляет собой категорию фактов или данных. Этот класс может содержать свойства и методы для работы с данными, связанными с конкретной категорией информации.

Class `Strategy(KnowledgeEngine)` — это концепция в объектно-ориентированном программировании, которая представляет собой паттерн проектирования, используемый для определения стратегий поведения классов в зависимости от определенных условий или контекста.

Class `SecondWindow(QMainWindow)` — это класс, который является подклассом стандартного виджета Qt под названием `QMainWindow`. `QMainWindow` - это виджет верхнего уровня в приложении Qt, который представляет собой основное окно приложения с панелью инструментов, меню и возможностью управления различными виджетами внутри окна.

Class `MainWindow(QMainWindow)` — это класс, который также является подклассом стандартного виджета Qt под названием `QMainWindow`. Класс `MainWindow(QMainWindow)` обычно используется для создания основного окна приложения Qt, в котором размещаются основные элементы интерфейса, такие как меню, панель инструментов, центральная область и другие виджеты.

```
class Select_Graphics():...
class Kategori(Fact):
    pass
dict = {...}
menu_dict = {...}
class Strategy(KnowledgeEngine):...
class SecondWindow(QMainWindow):...
class MainWindow(QMainWindow):...
if __name__ == '__main__':...
```

Рисунок 10. Классы

Сгенерированный код. Описание интерфейса в формате XML. Этот XML-файл описывает пользовательский интерфейс для программы [4]. Также присутствуют различные теги, свойства и значения, которые определяют размещение и параметры виджетов (элементов интерфейса) в окне приложения. `QMainWindow` (главное окно), `QPushButton` (кнопка), `QLineEdit` (поле для ввода текста), `QLabel` (метка). Также указаны различные свойства для каждого элемента, такие как геометрия (положение и размер), текст на кнопке, заголовок окна. Этот XML-файл используется для создания пользовательского интерфейса в программировании с использованием библиотеки, поддерживающей описание интерфейсов в формате XML, например, Qt для создания графических приложений на Python (Рисунок 11).

`QMainWindow` (главное окно), `QGraphicsView` (виджет для отображения графики).

Программа загружает данные о ценах акций из файлов формата CSV, проводит предобработку данных (удаляет ненужные столбцы, разделяет данные на обучающую и тестовую выборки), обучает модель линейной регрессии на обучающих данных и делает предсказания на тестовых данных [5].

Затем выводится результат предсказаний и осуществляется некоторая обработка данных (Рисунок 12–14).

```
<?xml version="1.0" encoding="UTF-8"?>
<ui_ versian="4.0">
<class>MainWindow</class>
<widget class="QMainWindow" name="MainWindow">
<property name="geometry">
<rect>
<x>0</x>
<y>0</y>
<width>992</width>
<height>241</height>
</rect>
</property>
<property name="minimumSize">
<size>
<width>992</width>
<height>231</height>
</size>
</property>
<property name="maximumSize">
<size>
<width>992</width>
<height>241</height>
</size>
</property>
<property name="windowTitle">
<string>MainWindow</string>
</property>
<widget class="Qwidget" name="centralwidget">
<widget class="QPushButton" name="pushButton">
<property name="geometry">
<rect>
<x>140</x>
<y>150</y>
<width>75</width>
<height>24</height>
</rect>
</property>
<property name="text">
<string>PushButton</string>
</property>
</widget>
<widget class="QLineEdit" name="FildInput_one">
<property name="gecometry">
<TtecD
<x>70</x>
<y>30</y>
<width>241</width>
<height>22</height>
```

Рисунок 11. Код для создания окна для ввода данных проекта

```
GNU nano 7.2
<?xml version "1.0" encoding«UTF-8"?>
<ui versione"4.0">
<class MainWindow</class>
aridget class="QMainWindow" none MainWindow">
<property name="geometry">
<rect>
<0</x>
<y>0</y>
<width>1106</width>
cheight 904</height>
</rect>
</property>
<property name="minimumSize"> <size>
<width>904</width>
<height>0</height>
</size>
</property>
<property name "maximumSize">
<size>
<width>1106</width>
<height>904</height>
</size>
</property>
<property name="windowTitle">
<string MainWindow</string>
</property>
<widget class="Qwidget" name="centralwidget">
aridget class="QGraphicsView" name="graphicsview"> <property name="geometry">
<rect>
<<>10</x>
<y>120</y>
<width>331</width>
<height>231</height>
</rect>
</property>
</widget>
widget class "QGraphicsView" name="graphicsview_2">
<property name=" geometry">
<rect>
<<x-10</x>
<y>370</y>
aridth>331</width>
<height 231</height>
</rect>
</property>
</widget>
```

Рисунок 12. Код для создания окна для отображения данных по выбору стратегий

Программа загружает данные о ценах акций из файлов формата CSV, объединяет столбцы «<DATE>» и «<TIME>» в новый столбец «<ADD DATE>», после чего сохраняет обновленные данные в новые CSV-файлы в другой директории.

Смысл программы заключается в обработке и изменении данных о ценах акций компаний перед их сохранением в новые файлы для дальнейшего использования или анализа (Рисунок 14). Визуализация проекта (Рисунок 15–18).

```

GNU nano 7.2
import time as tp
import numpy as np
import pandas as pd
import seaborn as sns
import dncerne as OT
from datetine fuport dacotice
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

directory_graphics = "Stock_File/"
Name_File_Promotion = [
1: "AFL 240322_240422.csv",
2: "AFLT_240322_240422.csv",
3: "GHN 240322_240422.csv",
4: "CSCO 240322_240422.csv",
5: "DG 240322_240422.csv",
6: "LNON-240322_240422.csv",
7: "MGNT 240322_240422.csv",
8: "PIKK 240322_240422.csv",
9: "SBER 240322-240422.csv",
10: "SIBN 240322-240422.csv",
11: "SNGS 240222_240422.csv",
12: "SONY 240322_240422.csv",
13: "TMOS 240322_240422.csv",
14: "TSLA 240522_240422.csv",
15: "UBER 240322_240422.csv",
16: "GAZP 240322-240422.csv",
17: "VTBR 240522_240422.csv",
18: "VNDX 240222_240422.csv"
]

for i in Name_File_Promotion:
    data = pd.read_csv(directory_graphics + i)
    print(data[4])
    new_data = data[["DATE", "TIME"]]
    data["ADD_DATE"] = new_data
    csv_data = data.drop(columns=["Unnamed: 0"])
    csv_data.to_csv(directory_graphics + i)

```

```

GNU nano 7.2 import pandas as pd
directory "Stock File/" directory_2 "Stock_File_test/"
Name_File_Promotion = {
1: "AAPL 240322_240422.csv",
2: "AFLT_240322_240422.csv",
3: "CHMF 240322_240422.csv",
4: "CSCO 240322_240422.csv",
5: "DG 240322_240422.csv",
6: "LKOH 240322_240422.csv",
7: "MGNT 240322_240422.csv",
8: "PIKK 240322_240422.csv",
9: "SBER 240322_240422.csv",
10: "SIBN 240322_240422.csv",
11: "SNGS_240322_240422.csv",
12: "SONY 240322_240422.csv",
13: "TMOS 240322_240422.csv",
14: "TSLA 240322_240422.csv",
15: "UBER_240322_240422.csv",
16: "GAZP 240322_240422.csv",
17: "VTBR 240322_240422.csv",
18: "VNDX 240322_240422.csv"
}

for i in Name_File_Promotion:
    data = pd.read_csv(directory + i)
    print(data[4])
    new_data = data[["DATE", "TIME"]]
    data["ADD_DATE"] = new_data
    csv_data = data.drop(columns=["Unnamed: 0"])
    csv_data.to_csv(directory_2 + i)

```

Рисунок 13. Код программы Make_Graphics.py

Рисунок 14. Код Greate_Data_File.py

Элемент	Описание элемента
1	Сумма
2	Цель
3	Риск
4	Ввод
5	Выбор дохода
6	Выбор показателя
7	Данные успешно получены
8	Кнопка

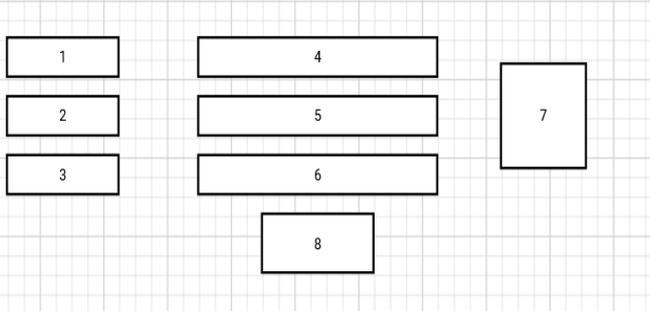


Рисунок 15. Макет



Рисунок 16. Начальное окно

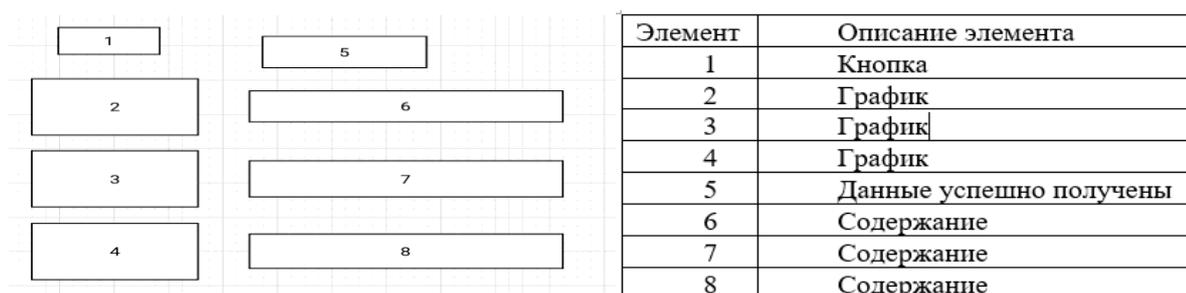


Рисунок 17. Макет

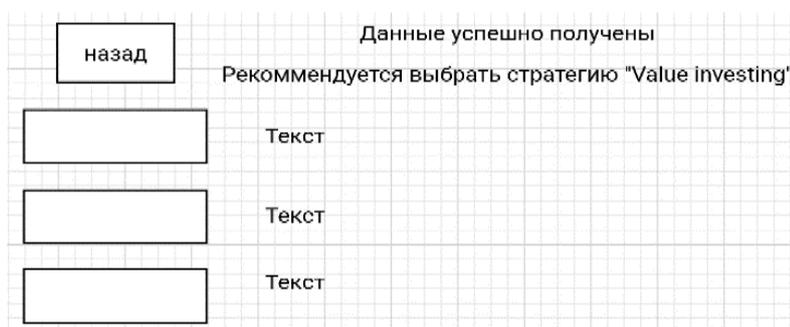


Рисунок 18. Описание работы

Заключение

В ходе исследования была разработана экспертная система для анализа котировок акций, включающая все ключевые компоненты: базу знаний, решатель, механизмы объяснения и диалоговый интерфейс. Система реализована на Python с использованием современных библиотек (NumPy, Pandas, Seaborn) и методов машинного обучения (линейная регрессия). Основные этапы разработки включали сбор и предобработку финансовых данных, создание алгоритмов анализа, проектирование пользовательского интерфейса и тестирование системы. В результате создано решение, способное обрабатывать рыночные данные, выявлять закономерности и формировать прогнозы. Экспертная система демонстрирует эффективность при анализе котировок акций и может быть полезна как частным инвесторам, так и финансовым аналитикам. Дальнейшее развитие проекта может включать расширение функционала за счет дополнительных методов анализа и интеграции с другими источниками данных.

Список литературы:

1. Речинский А. В., Станкевич Л. А., Черненькая Л. В., Черненький А. В. Анализ особенностей построения экспертных систем // Известия Тульского государственного университета. Технические науки. 2023. №4. С. 199-205.
2. Антонов Г. В., Иванов С. И. Линейная регрессия как один из методов статистического исследования // Известия Великолукской государственной сельскохозяйственной академии. 2021. №2. С. 64-75.
3. Бабилова Н. Н. Применение библиотеки numpy для векторизации кода Python // Вестник Сыктывкарского университета. Серия 1. Математика. Механика. Информатика. 2023. №1 (46). С. 14-29.
4. Алипатова Д. Г., Свищёв А. В. Сравнительный анализ XML и Jetpack Compose в Android разработке // Международный журнал гуманитарных и естественных наук. 2024. №11-2. С. 142-146.
5. Ларкина А. А., Портнов К. В., Аникин Д. В. Алгоритм формирования обучающей выборки на основе метода кластеризации // Journal of Monetary Economics and Management. 2024. №6. С. 38-42.

References:

1. Rechinskii, A. V., Stankevich, L. A., Chernen'kaya, L. V., & Chernen'kii, A. V. (2023). Analiz osobennostei postroeniya ekspertnykh sistem. *Izvestiya Tul'skogo gosudarstvennogo universiteta. Tekhnicheskie nauki*, (4), 199-205. (in Russian).
2. Antonov, G. V., & Ivanov, S. I. (2021). Lineinaya regressiya kak odin iz metodov statisticheskogo issledovaniya. *Izvestiya Velikolukskoi gosudarstvennoi sel'skokhozyaistvennoi akademii*, (2), 64-75. (in Russian).
3. Babikova, N. N. (2023). Primenenie biblioteki numpy dlya vektorizatsii koda Python. *Vestnik Syktyvkarskogo universiteta. Seriya 1. Matematika. Mekhanika. Informatika*, (1 (46)), 14-29. (in Russian).
4. Alipatova, D. G., & Svishev, A. V. (2024). Sravnitel'nyi analiz XML i Jetpack Compose v Android razrabotke. *Mezhdunarodnyi zhurnal gumanitarnykh i estestvennykh nauk*, (11-2), 142-146. (in Russian).
5. Larkina, A. A., Portnov, K. V., & Anikin, D. V. (2024). Algoritm formirovaniya obuchayushchei vyborki na osnove metoda klasterizatsii. *Journal of Monetary Economics and Management*, (6), 38-42. (in Russian).

*Работа поступила
в редакцию 20.04.2025 г.*

*Принята к публикации
27.04.2025 г.*

Ссылка для цитирования:

Окулов М. Д., Денисенко В. К. Разработка и проектирование экспертной системы для анализа котировок акций // Бюллетень науки и практики. 2025. Т. 11. №6. С. 169-181. <https://doi.org/10.33619/2414-2948/115/24>

Cite as (APA):

Okulov, M., & Denisenko, V. (2025). Development and Design of an Expert System for Stock Price Analysis. *Bulletin of Science and Practice*, 11(6), 169-181. (in Russian). <https://doi.org/10.33619/2414-2948/115/24>