Geekbrains

Анализ предсказаний курсов валют с помощью

модели машинного обучения на разный промежуток времени в зависимости от ключевых индикаторов на фондовом рынке

Программа:

Специализация: Искусственный интеллект. Цифровые Профессии

Студент: Эгипти Евгений Викторович

Тольятти

2024

Оглавление

[Введение 1](#_Toc172391428)

[Глава 1 Теоретическая часть 3](#_Toc172391429)

[1.1 Формирование гипотезы 3](#_Toc172391430)

[1.2 Пример 3](#_Toc172391431)

[Глава 2 Практическая часть 4](#_Toc172391432)

[2.1 Среда разработки 4](#_Toc172391433)

[2.2 Выбор датасетов 5](#_Toc172391434)

[2.3 Создание файла Jupiter Notebook 5](#_Toc172391435)

[2.4 Установка необходимых python-пакетов 6](#_Toc172391436)

[2.5 Импортируем нужные модули 6](#_Toc172391437)

[2.6 Загрузка датасетов 7](#_Toc172391438)

[Вывод 10](#_Toc172391439)

[Литература 11](#_Toc172391440)

Введение

Роль валютного курса в экономике переоценить сложно. Изменения валютного курса предопределяют статус той или иной страны в мировом экономическом пространстве, а вместе с ним ориентиры ее будущего развития, как во внешних, так и во внутренних связях.

Курс национальной валюты является важным ориентиром для принятия решений экономическими агентами, как на уровне населения, так и на уровне правительства. При высокой волатильности валюты возрастают издержки экспортно-импортных операций, растут цены на импортируемые товары. Это ведет к повышению цен на продукцию на внутреннем рынке. Чтобы защититься от роста цен и возможной девальвации, потребители увеличивают долю сбережений в иностранной валюте, что не способствует укреплению и стабилизации национальной. Резкие скачки национальной валюты дают повод населению снизить потребление, а предпринимателем сократить инвестиции в национальную экономику. Высокая волатильность препятствует бизнесу в построении долгосрочных стратегий развития, что приводит к снижению экспорта предприятиями и замедлением производства [1].

На данный момент машинное обучение является одной из наиболее развивающихся областей прикладной математики, позволяющих решать большой спектр задач предсказания и распознавания. Многие экономисты используют методы анализа данных для предсказания валютных курсов. Так, например, Martin Evans и Richard Lyons в своей статье «Micro-Based Exchange-Rate Forecasting» используют метод k ближайших соседей и метод опорных векторов для прогнозирования основных мировых валютных пар (EUR/USD, GBP/USD, USD/JPY) [2].

**Темой проекта** является предсказание курса валют с помощью модели машинного обучения.

**Целью** будет являться создание модели машинного обучения, которая будет предсказывать курс валют по отношению к рублю.

**Будет решена проблема** вероятности недополучить прибыль или понести убытки от финансовых, торговых и кредитных операций из-за изменчивости соотношения валют**.** То есть снизятся валютные риски.

**Задачи:**

* Собрать и обработать данные необходимые для обучения модели;
* Обучить модель на собранных данных;
* Поэкспериментировать с разными типами моделей;
* Проверить эффективность модели на тестовых данных;
* Проверить предсказания модели на актуальных данных.

**Инструменты:**

JupyterLab

**Состав команды:** Эгипти Евгений Викторович

**уц**

1. Теоретическая часть
   1. Формирование гипотезы
   2. Пример
2. Практическая часть
   1. Среда разработки

В качестве среды разработки был использован JupyterLab [3], там больше возможностей по сравнению с Jupyter Notebook. Можно одновременно работать с несколькими файлами, использовать отладчик [6], если потребуется.

Данная среда разработки, а также другие инструменты входят в бесплатную программу Anaconda [7] которую можно скачать с сайта <https://www.anaconda.com/>.

Разработку проекта желательно вести в новом окружении, чтобы не возникло конфликта между версиями устанавливаемых пакетов. Для этого, открываем программу Anaconda и выбираем вкладку Environments, в открывшейся вкладке нажимаем кнопку Create, задаём название окружения и версию Python. Делаем активным наше окружение, в данном случае окружение diplom\_2024\_gb, см. рисунок 1.

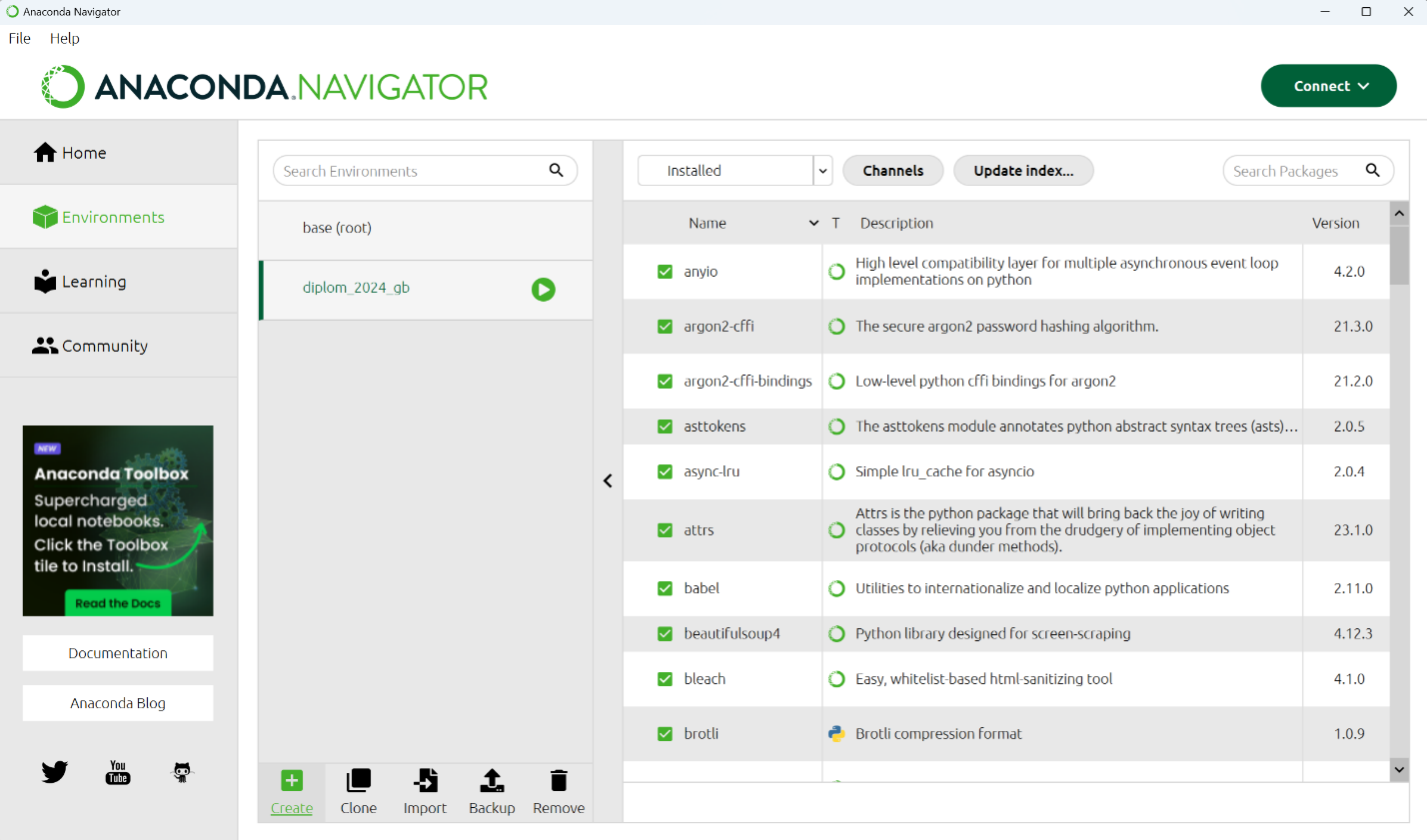


Рисунок 1. Выбор активного окружения diplom\_2024\_gb

После выбора окружения, нажимаем вкладку Home и выбираем среду разработки JupyterLab.

* 1. Выбор датасетов

Датасеты были взяты с сайта InvestFunds (<https://investfunds.ru/>), их можно бесплатно скачать в формате Excel, за выбранный промежуток времени.

В качестве ключевых индикаторов [4] были приняты следующие показатели:

* индекс Мосбиржи;
* индекс РТС;
* индекс S&P 500;
* платина (Банк России);
* нефть Brent;
* золото (Банк России);
* серебро (Банк России);
* палладий (Банк России).

В качестве таргетов (целевой переменной) были взяты следующие курсы валют [5]:

* доллар США / Российский рубль;
* Евро / Российский рубль;
* бивалютная корзина / Российский рубль;
* Британский фунт стерлингов / Российский рубль;
* Швейцарский франк / Российский рубль;
* Китайский юань / Российский рубль;
* Японская иена / Российский рубль.
  1. Создание файла Jupiter Notebook

После того как мы установили JupyterLab, выбрали рабочее окружение и открыли среду разработки (пункт 2.1), создадим файл Jupiter Notebook. Для этого нужно перейти меню File → New → Notebook.

Файл Jupiter Notebook состоит из ячеек, в которых может находиться код программы или пояснительный текст программы, написанный на языке Markdown. Код каждой ячейки может выполняться отдельно, а также есть возможность запустить выполнение всех ячеек. Результат выполнения ячейки может выводиться непосредственно после самой ячейки.

* 1. Установка необходимых python-пакетов

Для того чтобы использовать необходимые нам библиотеки (Pandas, NumPy, Matplotlib и т.д.) нужно установить соответствующие пакеты с помощью системы управления пакетами pip. Для этого в ячейки ноутбука запустим следующий код:

ячейка Jupiter Notebook

# Для каждого проекта я использую новое окружение,

# думаю это помогает избежать ошибок связанных с версиями пакетов

!pip install pandas

!pip install matplotlib

!pip install seaborn

!pip install numpy

!pip install scikit-learn

!pip install openpyxl

* 1. Импортируем нужные модули

ячейка Jupiter Notebook

# Импортируем необходимые библиотеки

import numpy as np

import pandas as pd

from matplotlib import pyplot as plt

import seaborn as sns

# Следующая магическая команда Jupyter Notebook нужна для того, чтобы графики

# отображались прямо в ноутбуке, а не в отдельном окне

%matplotlib inline

# Настройка более четкого отображения графиков

%config InlineBackend.figure\_format = 'svg'

# Модуль для зазбиения выборки на тренировочнию и тестовую

from sklearn.model\_selection import train\_test\_split

# Уберем warnings

import warnings

warnings.filterwarnings('ignore')

# Настройка формата вывода чисел float

pd.set\_option('display.float\_format', '{:.2f}'.format)

# Вычисление Z-score

from scipy import stats

from sklearn.preprocessing import LabelEncoder

from scipy.stats.mstats import winsorize

# Дата и время

from datetime import datetime

# Случайные числа

import random as rnd

# Алгоритм машинного обучения 'Метод случайного леса'

from sklearn.ensemble import RandomForestRegressor

# Кросс-валидация

from sklearn.model\_selection import cross\_validate

# Разбиение

from sklearn.model\_selection import KFold

# Метрика r2

from sklearn.metrics import r2\_score

# Пути файловой системы

from pathlib import Path

# Регулярные выражения

import re

* 1. Загрузка датасетов

Загрузка файлов в дата фрейм происходит в автоматическом режиме. Программа читает все файлы с расширением xlsx в директории заданной константой PATH\_DATASET. Все файлы соединяются по полю Дата. Если число строк в файле меньше, чем MIN\_ROW, то такие данные не будут включены в итоговый дата фрейм. В дата фрейме название столбца будет соответствовать названию файла. Информация о ходе процесса выводится под ячейкой.

ячейка Jupiter Notebook

# Расположение данных

PATH\_DATASET = './Dataset'

# Минимальное разрешённое количество строк для загрузки

MIN\_ROW = 6000

# Итоговый датафрейм со всеми данными

data\_loaded = pd.DataFrame()

# Расположение файлов датасета

p = Path(Path.cwd() / PATH\_DATASET)

for obj in p.iterdir():

    if obj.is\_file():

        \*\_, name\_file\_all = str(obj).split('\\')

        name\_file, type\_file = name\_file\_all.split('.')

        if type\_file == 'xlsx':

            df\_temp = pd.read\_excel(f'{PATH\_DATASET}/{name\_file\_all}')

            # Переименуем стобцы

            df\_temp.rename(columns={'Значение': f'{name\_file}'}, inplace=True)

            # Нормализуем дату

            df\_temp['Дата'] = df\_temp['Дата'].dt.normalize()

            # Проверяем условие первой загрузки

            if data\_loaded.shape[1]:

                if df\_temp.shape[0] >= MIN\_ROW:

                    print(f'Обработан: {name\_file\_all} {df\_temp.shape}')

                    data\_loaded = pd.merge(data\_loaded, df\_temp, on='Дата', how='inner')

                    print(f'Размерность после merge {data\_loaded.shape}\n')

                else:

                    print(f'Отклонён, мало строк: {name\_file\_all}\n')

            else:

                data\_loaded = df\_temp

                print(f'Обработан: {name\_file\_all} {df\_temp.shape}\n')

результат выполнения

Обработан: chf\_rub-(банк-россии).xlsx (7303, 2)

Обработан: eur\_rub-(банк-россии).xlsx (7303, 2)

Размерность после merge (7303, 3)

Обработан: gbp\_rub-(банк-россии).xlsx (7303, 2)

Размерность после merge (7303, 4)

Обработан: jpy\_rub-(банк-россии).xlsx (7303, 2)

Размерность после merge (7303, 5)

Обработан: s-p-500.xlsx (6178, 2)

Размерность после merge (5817, 6)

Обработан: usd\_rub-(банк-россии).xlsx (7303, 2)

Размерность после merge (5817, 7)

Отклонён, мало строк: бивалютная-корзина\_rub.xlsx

Обработан: золото-(банк-россии).xlsx (6034, 2)

Размерность после merge (4951, 8)

Обработан: индекс-мосбиржи.xlsx (6130, 2)

Размерность после merge (4849, 9)

...

Обработан: серебро-(банк-россии).xlsx (6034, 2)

Размерность после merge (4842, 14)

Выведем первые пять строк дата фрейма.

ячейка Jupiter Notebook

# Посмотрим, что загрузилось

data\_loaded.head()

Проверим размер загруженных данных.

ячейка Jupiter Notebook

# Посмотрим размер

data\_loaded.shape

результат выполнения

(4842, 14)

* 1. Разведочный анализ данных (EDA)

# Вывод

Выполнив данное ДЗ, я научился писать скрипты для Linux, а также прокачал свои навыки в Word при оформлении данного ДЗ.

1. Литература

1. ФАКТОРЫ, ВЛИЯЮЩИЕ НА ФОРМИРОВАНИЕ КУРСА РУБЛЯ Е. Я. Волков. – URL: https://view.officeapps.live.com/op/view.aspx?src=https%3A%2F%2Fwww.kubsu.ru%2Fsites%2Fdefault%2Ffiles%2Fusers%2F21365%2Fportfolio%2Fmvko\_i\_fo\_kursovaya.docx&wdOrigin=BROWSELINK (дата обращения: 11.07.2024) – Текст: электронный.

2. Особенности методов машинного обучения для прогнозирования курса валют Е. В. Сангаджиева, Г. А. Оргдаева. – URL: https://files.scienceforum.ru/pdf/2020/5e04936d95941.pdf (дата обращения: 11.07.2024) – Текст: электронный.

3. Основы работы в Jupyter/Jupiter Notebook и JupyterLab — Python Tutorial И. Шамаев. – URL: https://python.ivan-shamaev.ru/jupiterlab-jupyter-notebook-install-python-tutorial/ (дата обращения: 19.07.2024) – Текст: электронный.

4. Ключевые индикаторы InvestFunds. – URL: https://investfunds.ru/indicators/key-indexes/ (дата обращения: 20.07.2024) – Текст: электронный.

5. Курсы валют InvestFunds. – URL: https://investfunds.ru/indicators/currency/ (дата обращения: 20.07.2024) – Текст: электронный.

6. Debugger JupiterLab. – URL: https://jupyterlab.readthedocs.io/en/4.1.x/user/debugger.html (дата обращения: 19.07.2024) – Текст: электронный.

7. Anaconda. – URL: https://www.anaconda.com/ (дата обращения: 19.07.2024) – Текст: электронный.