



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Зав. кафедрой ПОВТиАС  
\_\_\_\_\_  
(подпись) В.В.Долгов  
(И.О.Ф.)  
«\_\_» \_\_\_\_\_ 2020г.

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

к курсовой работе по дисциплине **Объектно-ориентированное программирование**  
на тему: **Прогнозирование макроэкономических показателей методами машинного обучения.**

Автор проекта (работы) \_\_\_\_\_ Борисов Д.В.  
подпись

Направление/специальность, профиль/специализация:  
020303 Математическое обеспечение и администрирование информационных систем

Математическое обеспечение и администрирование информационных систем

Обозначение курсового проекта (работы) КР.250000.000 Группа ВМО32

Руководитель проекта: \_\_\_\_\_ доцент, Габрельян Б.В.  
подпись (должность, И.О.Ф.)

Проект (работа) защищен (а) \_\_\_\_\_  
дата оценка подпись

Ростов-на-Дону

2020



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ДОНСКОЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»  
(ДГТУ)**

Факультет «Информатика и вычислительная техника»

Кафедра «Программное обеспечение вычислительной техники и автоматизированных систем»

Зав. кафедрой ПОВТиАС  
В.В. Долгов  
(подпись) (И.О.Ф.)  
«\_\_» \_\_\_\_\_ 2020г.

### ЗАДАНИЕ

к курсовой работе по дисциплине Объектно-ориентированное программирование

Студент Борисов Д.В.

Код 02.03.03.250000.00 ПЗ

Группа ВМО32

Тема Прогнозирование макроэкономических показателей методами машинного обучения

Срок представления проекта (работы) к защите «\_\_» \_\_\_\_\_ 202\_ г.

Исходные данные для курсового проекта (работы)

задание на выполнение курсовой работы

И. Грэхем - Объектно-ориентированные методы. Принципы и практика

С.Расмуссен - Гауссовы процессы в машинном обучении

---

---

---

---

---

## Содержание пояснительной записки

### ВВЕДЕНИЕ:

В разделе «Введение» рассматриваются актуальность предметной области, необходимость изучения решения вопроса макроэкономического прогнозирования методами машинного обучения, обосновывается потребность в использовании математических методах прогнозирования.

Наименование и содержание разделов:

1 «Постановка задачи». В разделе задаются требования к конечному результату, содержится описание структуры программы для построения прогнозов. Описываются возможности пользователя и поведение программы при некоторых сценариях.

2 «Методы решения задачи». В параграфе описываются методы машинного обучения, наиболее приемлемые для решения задачи, объясняются причины выбора методов.

3 «Программное конструирование». В разделе рассматриваются инструменты разработки, описывается рабочий функционал компонентов программы. Описываются классы получения данных, обработки данных, интерфейса и т.д. и методы их взаимодействия.

4 «Требования к эксплуатации». В разделе описываются требования к устройству, эксплуатирующему программу.

5 «Тестирование». В разделе проверена работоспособность программного средства.

Приведён пример работы программного средства. Протестировано построение графиков, конструкция моделей.

### ЗАКЛЮЧЕНИЕ:

В курсовой работе, согласно заданию, в рамках объектно-ориентированной парадигмы программирования, были реализованы все задачи, создано рабочее приложение.

Перечень графического материала:

1. Рисунок 1 – Изначальный интерфейс пользователя
2. Рисунок 2 – Динамика ВВП выбранных стран
3. Рисунок 3 – Динамика прироста населения выбранных стран
4. Рисунок 4 – Динамика уровня инфляции выбранных стран
5. Рисунок 5 – Результат работы программы по нажатию «Build GPR»
6. Рисунок 6 – Результат работы программы по нажатию «Build VAR»

Руководитель проекта (работы)

\_\_\_\_\_

подпись, дата

\_\_\_\_\_

И.О.Ф.

Задание принял к исполнению

\_\_\_\_\_

подпись, дата

\_\_\_\_\_

И.О.Ф.

# СОДЕРЖАНИЕ

ВВЕДЕНИЕ	5
1 Постановка задачи	6
2 Методы решения задачи	7
2.2 Гауссов процесс	7
2.2 Векторная авторегрессия	8
3 Программное конструирование	9
4 Требования к эксплуатации	10
5 Тестирование	13
ЗАКЛЮЧЕНИЕ	18
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	19
ПРИЛОЖЕНИЕ А UML-диаграмма классов	20
ПРИЛОЖЕНИЕ Б Исходный код программы	21
ПРИЛОЖЕНИЕ Б Исходный код графического интерфейса	27

					КР.250000.00							
Изм.	Лист	№ докум.	Подпись	Дата								
Разраб.		Борисов Д.В.			Прогнозирование макроэкономических показателей методами машинного обучения				Лит.	Лист	Листов	
Провер.		Габрельян Б.В.									4	33
Проверил									ДГТУ кафедра «ПОВТиАС»			
Утверд.												

## ВВЕДЕНИЕ

Вопрос макроэкономического прогнозирования является важной темой в междисциплинарном пространстве экономистов, математиков и программистов. Необходимость в прогнозе фундаментальных макроэкономических показателей обусловлена потребностями мирового экономического сообщества в дальнесрочном прогнозировании, тем самым определяя региональное и мировое бюджетирование, политику социальных программ и т.д. [1]

Математическое обоснование жизнеспособности макроэкономических прогнозов является неотъемлемой частью современной системы прогнозирования. Качественная статистическая выборка, объем данных, развитость математического аппарата и вычислительных мощностей позволяет использовать высокоэффективные способы построения моделей, в том числе и моделей машинного обучения, нейронных сетей и самообучающихся алгоритмов.

Многие современные языки программирования имеют комплекс библиотек, связанных с построением моделей, основанных на машинном обучении. В связи с этим, актуален вопрос о качественной настройке параметров моделей, выборе данных и их предобработке для определения наиболее подходящих в данной области моделей, способной к наиболее точному прогнозированию и производству прогнозов.

Одними из наиболее общепризнанно точных моделей машинного обучения могут считаться векторная авторегрессия в силу своей свободы и возможности прогнозирования при умеренном количестве лагов и гауссовский процесс по причине своей универсальности на множестве данных, подчиняющихся логике экономических данных. [2]

						Лист
						5
Изм.	Лист	№ докум.	Подпись	Дата		

## 1 Постановка задачи

Разработать и реализовать программное средство, удовлетворяющее парадигмам ООП, выполняющее функцию построения моделей машинного обучения по импортируемым в реальном времени данным о следующих макроэкономических показателях:

1. валовый валютный продукт (ВВП) государства за год;
2. уровень безработицы населения государства за год;
3. средний уровень инфляции, наблюдаемый в государстве за год.

Информация о вышеуказанных показателях передается запросами от программного продукта от сервера-ресурса с данными.

Программное средство должно поддерживать активное взаимодействие с пользователем посредством:

1. графического интерфейса пользователя, имеющего возможность:
  - 1.1. отобразить импортируемые данные на графике;
  - 1.2. настроить выбор стран, данные из которых будут включены в итоговые модели;
  - 1.3. настроить модели машинного обучения;
  - 1.4. запустить построение моделей машинного обучения;
2. окна выводов результатов работы построенных моделей и логов программы;
3. использования функций-декораторов, информирующих пользователя о текущем состоянии процесса.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		6

## 2 Методы решения задачи

### 2.1 Гауссов процесс

Регрессия на основе гауссовских процессов (Кригинг) — это метод машинного обучения, для которого интерполированные значения моделируются гауссовским процессом, определяемым предыдущими ковариациями, в отличие от кусочно-полиномиального сплайна, оптимизирующего гладкость интерполируемых значений. Данный интерполяционный метод назван в честь южноафриканского горного инженера Дэниела Крига, занимавшегося ручным созданием геологических карт по ограниченному набору данных в некоторой области. Это вид обобщенной линейной регрессии, использующий статистические параметры для нахождения оптимальной оценки в смысле минимального среднеквадратического отклонения при построении поверхностей, кубов и карт. В основу метода положен принцип несмещенности среднего; то есть взятые все вместе значения на карте должны иметь правильное среднее значение. Глобальная несмещенность формально обеспечивается за счет повышения низких значений и уменьшения высоких.

При правильных выбранных априорных предположениях кригинг дает наилучшее линейное несмещенное предсказание промежуточных значений. Методы интерполяции, основанные на других критериях, таких как гладкость, не должны давать наиболее вероятных значений в промежуточных точках. [3]

В условиях поставленной задачи данный метод машинного обучения является особенно актуальным в силу качественной работы по малой выборке данных. Мировое научное сообщество активно применяет регрессию на основе гауссовых процессов в области пространственного анализа, что является схожей задачей с задачей макроэкономического прогнозирования.

						Лист
						7
Изм.	Лист	№ докум.	Подпись	Дата		

## 2.2. Векторная авторегрессия

Векторная авторегрессия – это метод машинного обучения, основанный на статистической модели, используемой для определения взаимосвязи между несколькими величинами по мере их изменения во времени. VAR - это разновидность модели стохастического процесса. Модели VAR обобщают модель авторегрессии с одной переменной (одномерной), допуская многомерные временные ряды. VAR-модели часто используются в экономике и естественных науках.

Как и в модели авторегрессии, каждая переменная имеет уравнение, моделирующее ее эволюцию во времени. Это уравнение включает запаздывающие (прошлые) значения переменной, запаздывающие значения других переменных в модели и член ошибки. Модели VAR не требуют столько знаний о силах, влияющих на переменную, как структурные модели с одновременными уравнениями. Единственное необходимое предварительное знание - это список переменных, которые, как можно предположить, будут влиять друг на друга с течением времени.

Традиционно, векторная авторегрессия решает эконометрические задачи, поэтому решение поставленной задачи является приближенным условием к классическим постановкам задач для VAR. Построение данной модели машинного обучения также ценно в следствие отсутствия необходимости излишнего в прогнозирующем вопросе теоретического обоснования полученных коэффициентах.

Исходя из указанных соображений, данный метод машинного обучения может быть применим в данной задаче, наряду с VAR.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		8



### 3 Программное конструирование

Для решения поставленной задачи необходимо соблюдать парадигмы ООП. Объектно-ориентированное программирование (ООП) — это парадигма разработки программных систем, в которой приложения состоят из объектов. В соответствии с этими принципами, необходимо соблюдать классовый характер итоговой программной структуры. [4]

На текущий момент, учитывая многообразие языков программирования, существует ограниченное множество средств, удобных для разработки продукта, основанного на машинном обучении. Поскольку не существует единственно верного языка для работы с машинным обучением, выбрана наиболее эмпирически приемлемая среда, удовлетворяющая условиям выполнения функциональных требований. Таким образом, разработка программного продукта велась на языке Python 3.7.

Python отвечает всем стандартным требованиям к разработке и является лидером среди современных разработок, использующих Machine Learning.

Данное утверждение основано на:

1. Наличии встроенных средств для работы с данными;
2. присутствии пользовательских библиотек, обеспечивающих упрощенное использование методов Machine Learning и Data Mining;
3. интуитивно понятном синтаксисе.

Для максимизации пользы от выбора языка задача построения алгоритмической модели была делегирована библиотеке scikit-learn, имеющей в функционале реализацию GPR, VAR и средства отображения точности, погрешностей и т.д. [5]

Структура полученной программы представляет собой множество классов с определенными методами, а также дополнительный файл интерфейса.

						Лист
						9
Изм.	Лист	№ докум.	Подпись	Дата		

Для инициализации переменных и процессов, для работы с данными имеется класс `data_class`, включающий в себя:

1. Статический метод `set_start()`, задающий начальные данные для создания обращения к API платформы с данными, а также инициализирующий сущность графического интерфейса.
2. Метод `update_data()`, посылающий запрос на загрузку данных при изменении пользователем государств, используемых в построении модели. Переопределяет ранее созданные `data`, `x`, `y` – данные, необходимые для построения моделей.

Класс `decorate`, используемый для реализации функций-декораторов, состоящий из:

1. Метода `decorator_func(func)`, возвращающей пользователю в текстовое окно данные о запущенных методах, предупреждающий об их начале выполнения и завершении.

Класс `import_data`, необходимый для загрузки данных и их первичной обработки. Класс включает в себя:

1. Метод `get_data_from_API()`, осуществляющую запрос к API платформы `worldbank.org`, принадлежащей международной финансовой организации Всемирный банк. Данные, полученные функцией конвертируются в `DataFrame` из изначального JSON файла, где приобретают разметку: информацию о стране, типе макроэкономического показателя и конкретном значении. Возвращает `DataFrame` с размеченной информацией.
2. Метод `set_logarifmical(d)`, принимающий на вход `DataFrame`, переводящий полученные данные в логарифмический вид для улучшения прогнозирования и качества модели, возвращает логарифмический вид данных, очищенный от пустых значений.

Класс `graph()`, необходимый для построения графиков в программе, состоит из:

						Лист
						10
Изм.	Лист	№ докум.	Подпись	Дата		

1. Метода `plotting(data)`, строящего график на основе переданных данных.

Класс `modelling`, состоящий из моделей машинного обучения и их применения:

1. Метод `VAR_modeling(target_data)`, строящий модель векторной авторегрессии и вычисляющий собственную среднеквадратичную ошибку.
2. Метод `gaussian_model(target_data, da, a, r)`, строящий по передаваемым данным и параметрам гауссову модель-регрессор и вычисляющий собственную среднеквадратичную ошибку.

Класс `Ui`, содержащий пользовательский интерфейс, состоящий из:

1. Инициализирующего для класса метода `__init__(self)`, связывающего компоненты импортируемого из файла `forecasting.ui` интерфейса с методами настоящей программы. По итогу работы отображает пользователю готовый интерфейс.
2. Метода `check_boxes()`, проверяющего наличие выбора стран в соответствующих элементах типа `checkBox`.
3. Метода `print_log_result(log, method)`, выводящего результаты вычислений построенной модели в текстовое окно.
4. Метода `print_log`, выводящего в текстовое окно пользователя прочие сообщения программы.

Модуль `__main__` запускает выполнение программы, вызывая `data_class.set_start()`.

### 3 Требования к эксплуатации

Для успешного использования программного средства программистом, необходимо ряда условий:

1. Наличие интерпретатора Python 3.7;
2. Подключение к высокоскоростному Интернет-соединению;
3. Установка библиотек requests, pandas, numpy, matplotlib, statsmodels, sklearn, PyQt5

Установка необходимых библиотек возможна посредством пакетного менеджера python – pip. При наличии pip последней версии необходимо ввести в консоль python команду:

```
pip install requests pandas numpy matplotlib statsmodels sklearn PyQt5
```

Данная команда установит на ПК необходимые для работы программы библиотеки.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		12

## 5 Тестирование

При запуске программы откроется интерфейс пользователя (Рисунок 1).

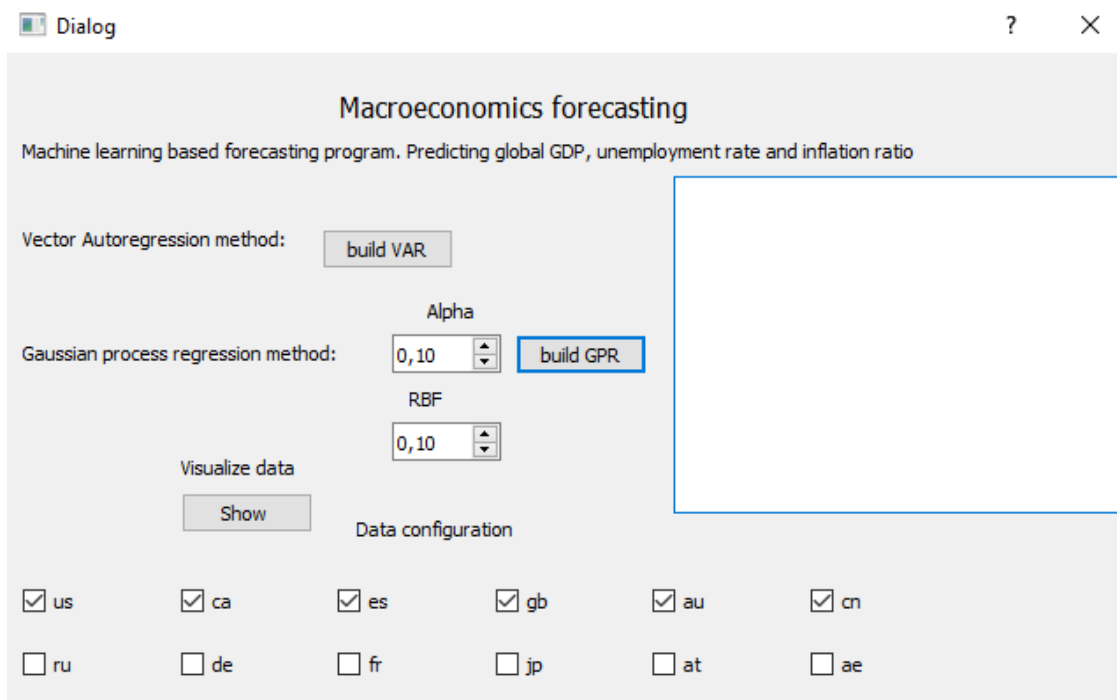


Рисунок 1 - Изначальный интерфейс пользователя

При нажатии кнопки «Show» пользователю будут выведены графики с данными по макроэкономическим показателям выбранных в разделе «Data configuration» стран (Рисунок 2, Рисунок 3, Рисунок 4).

Figure 1

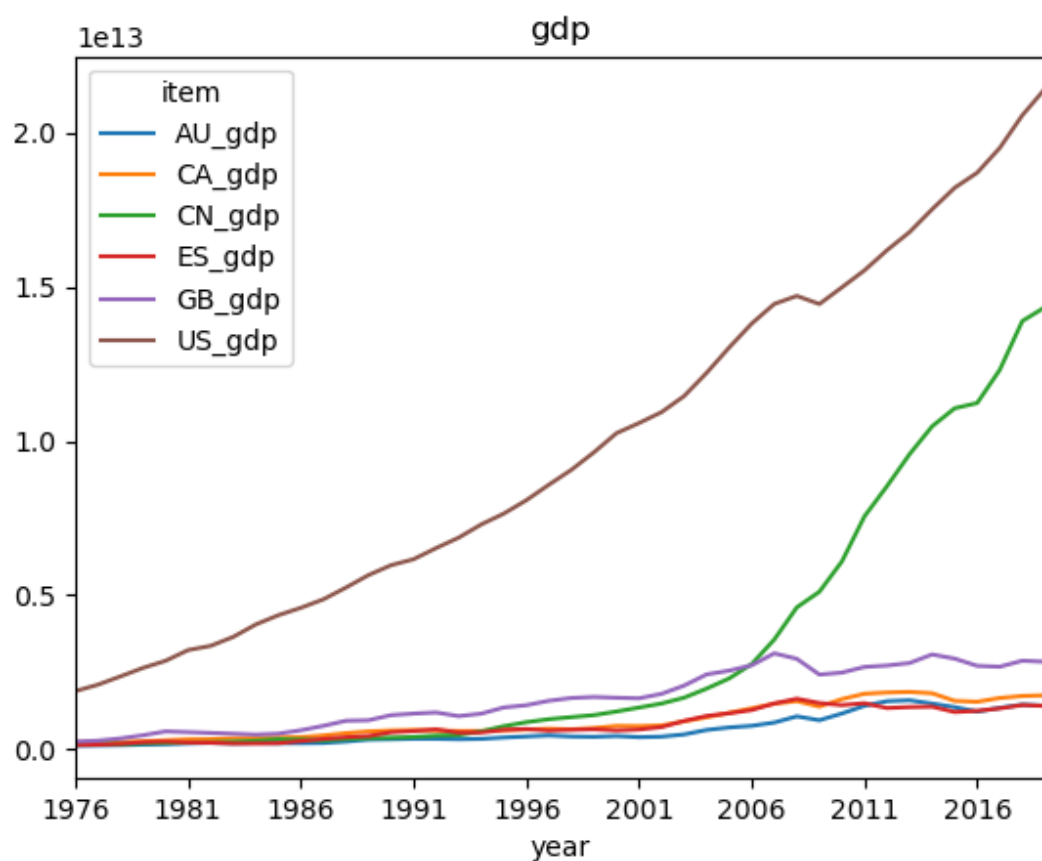


Рисунок 2 – Динамика ВВП выбранных стран

Figure 1

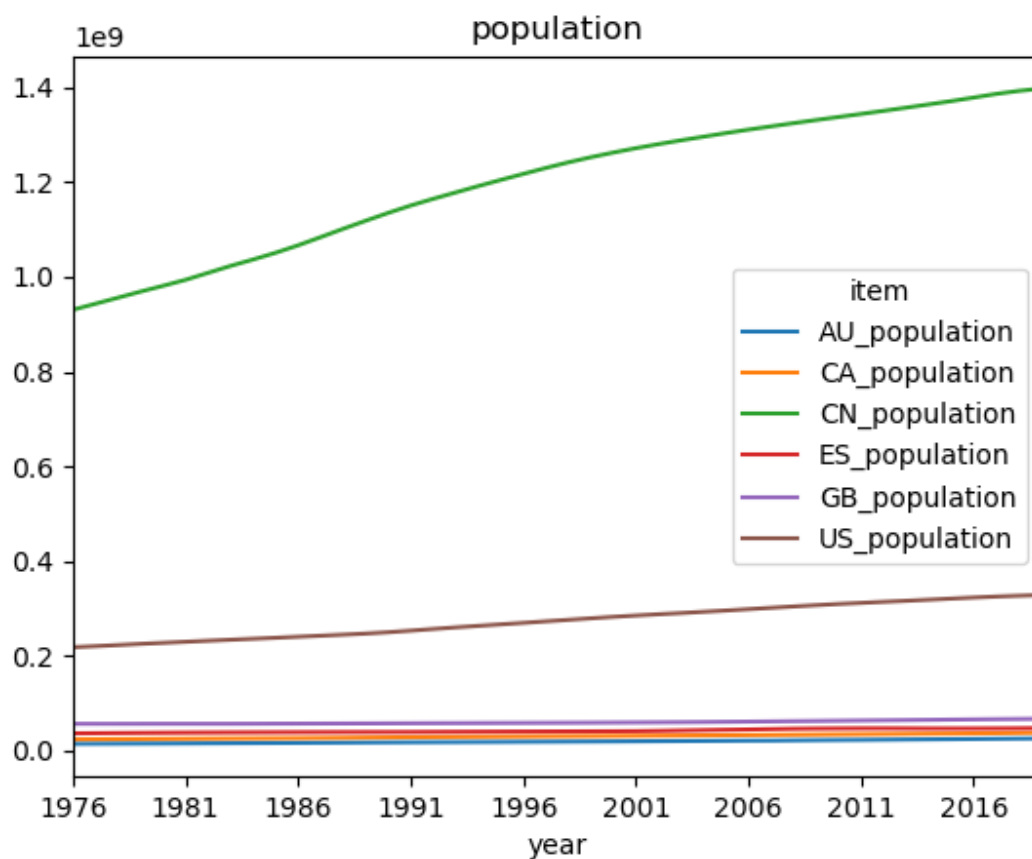


Рисунок 3 – Динамика прироста населения выбранных стран

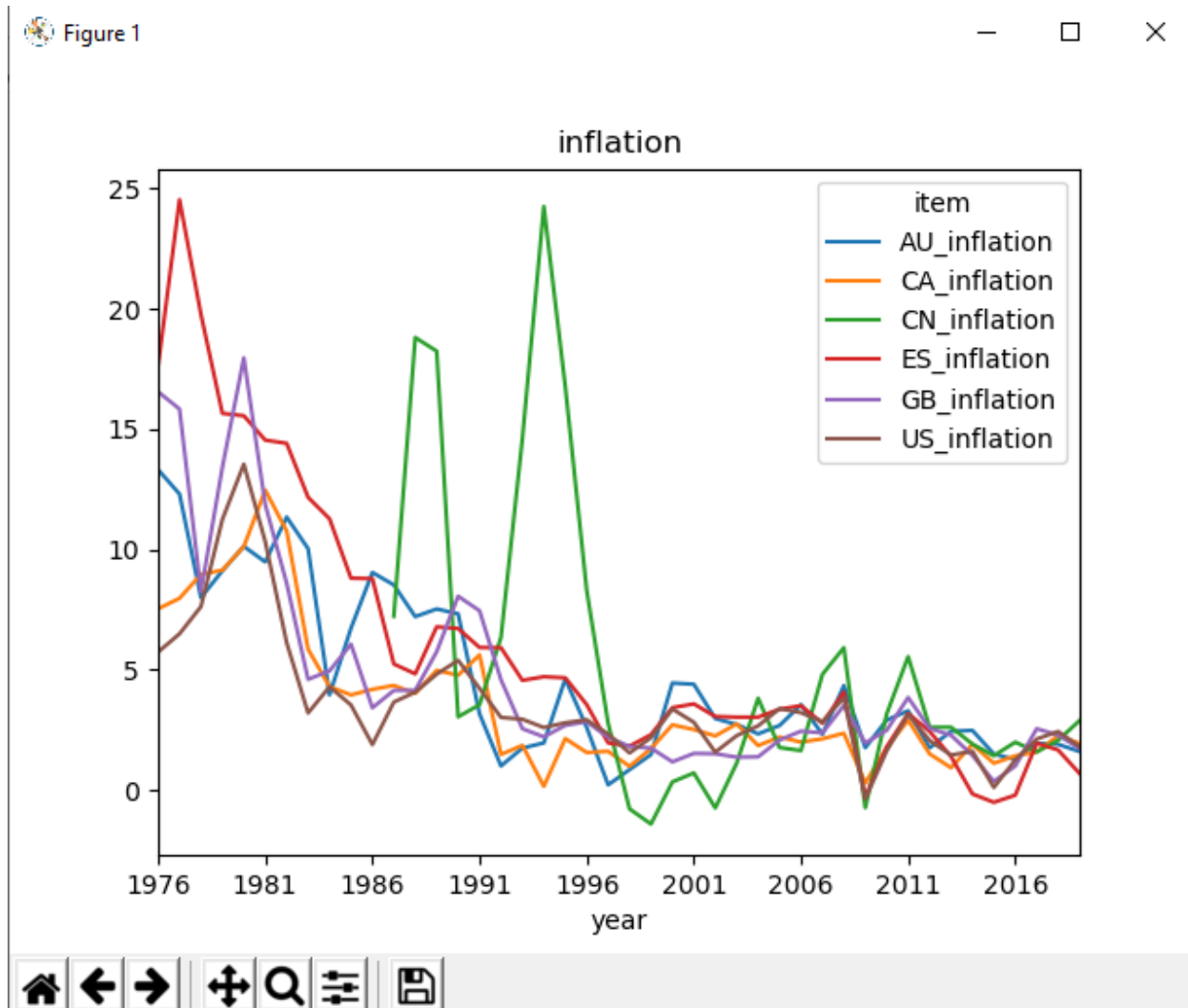


Рисунок 4 – Динамика уровня инфляции выбранных стран

После уточнения параметров «Alpha» и «RBF» пользователь имеет возможность построить GPR модель машинного обучения (Рисунок 5) и получить в качестве результата значение ошибки.



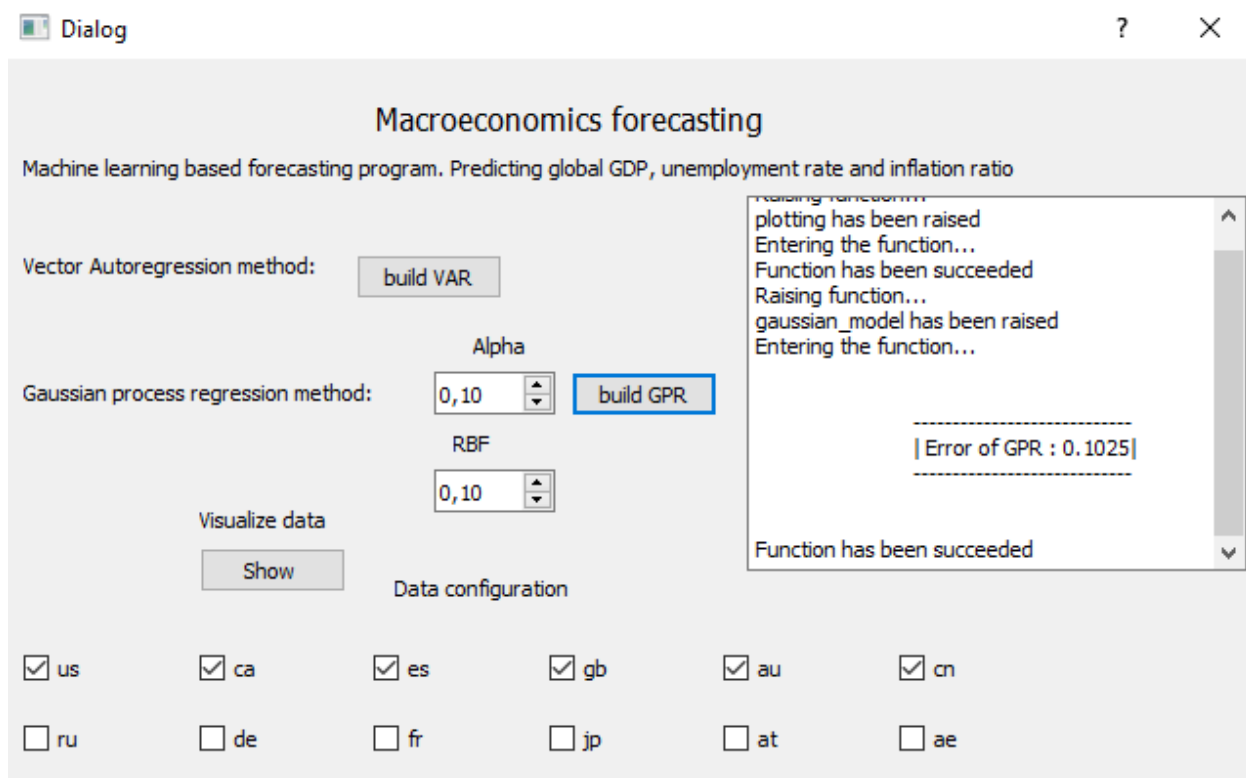


Рисунок 5 – Результат работы программы по нажатию «Build GPR»

Поскольку метод векторной авторегрессии не является настраиваемым, для его вызова необходимо и достаточно нажать «Build VAR» (Рисунок 6).

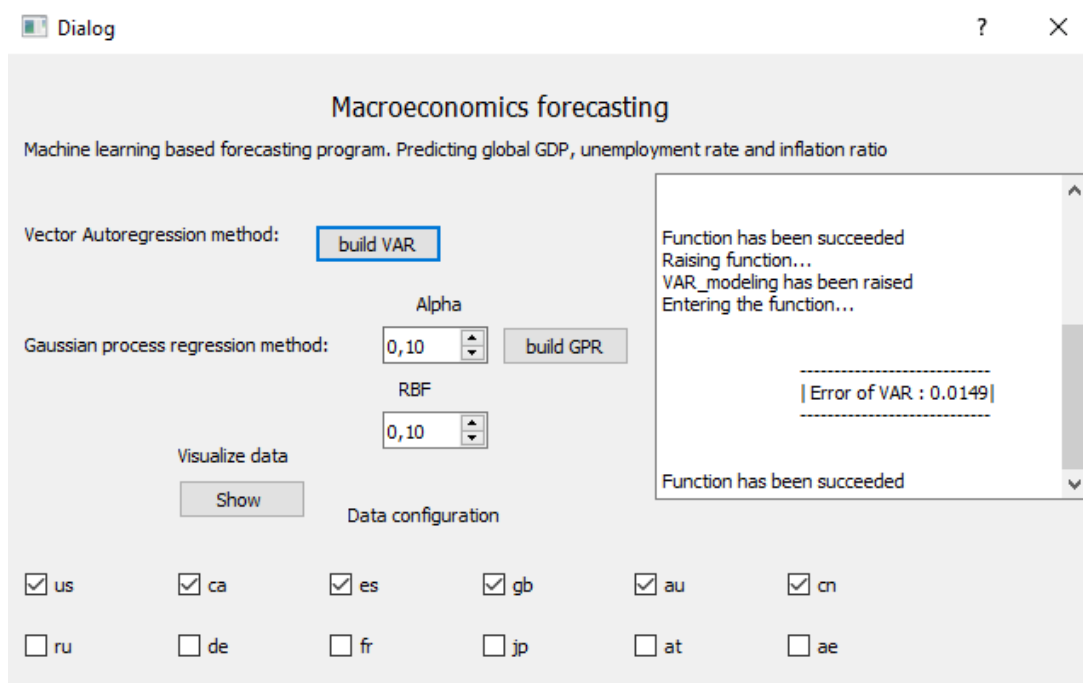


Рисунок 6 – Результат работы программы по нажатию «Build VAR»

Повторный вызов построения методов при изменении стран ведет к изменению показателя ошибки.

## ЗАКЛЮЧЕНИЕ

Результатом проделанной работы стало разработанное и реализованное рамках парадигм ООП программное средство, строящее VAR и GPR модели машинного обучения по импортируемым в реальном времени данным о макроэкономических показателях.

В ходе работы был повышен уровень знаний в области применения ООП, методов машинного обучения и разработки графических интерфейсов пользователя.

Все поставленные ранее задачи по курсовому проекту были успешно выполнены.

Написанная программа соответствует заявленным требованиям задания на курсовую работу и успешно прошла тестирование.

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		18

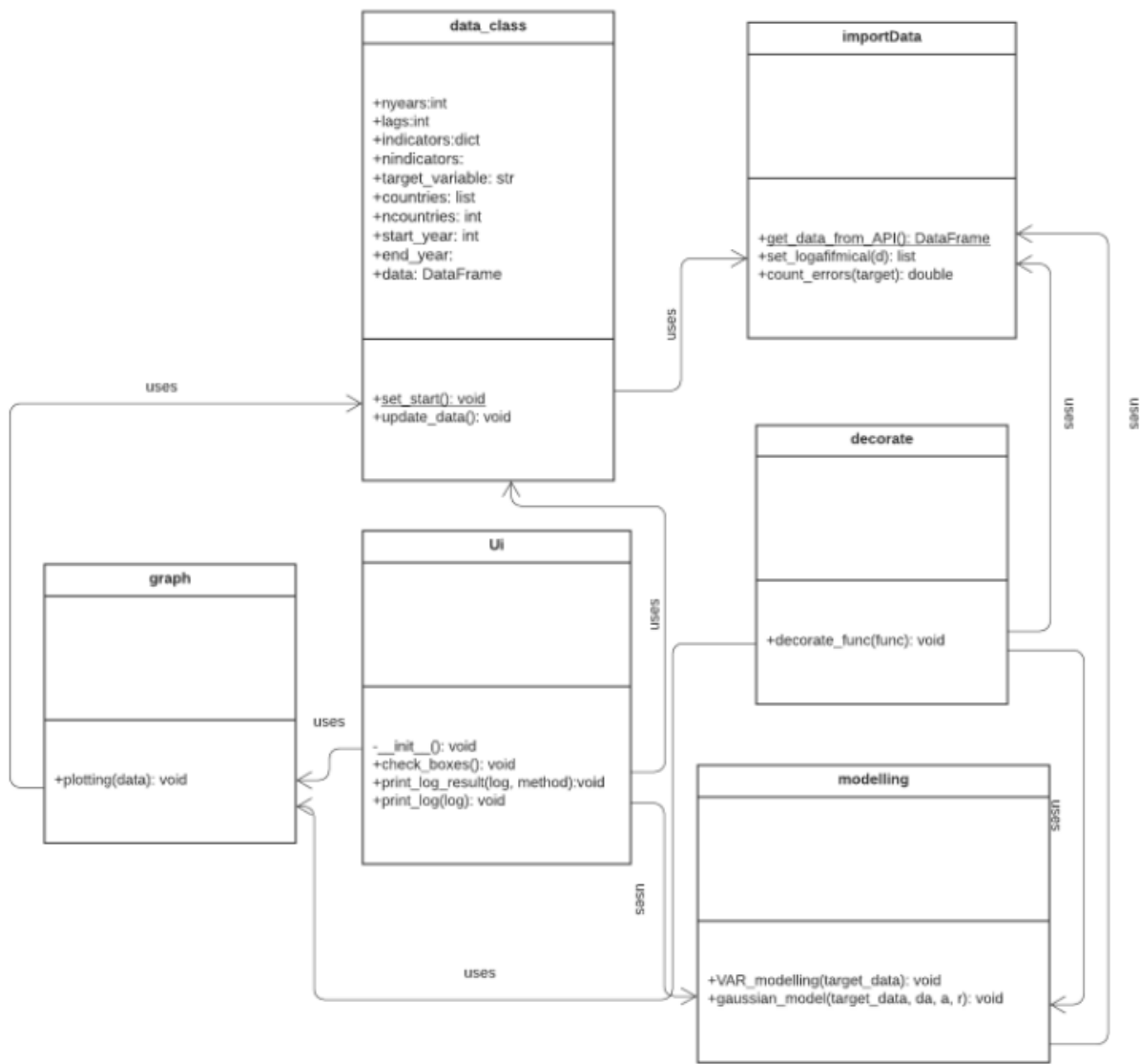
## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Абель Э., Бернанке Б. Макроэкономика / Пер. с англ. Н. Габенова, А. Смольского; научн. ред. Л. Симкина. — 5-е изд.. — СПб.: Питер, 2008. — ISBN 978-5-469-01110-1.
2. С. Е. Rasmussen & С. К. I. Williams, Gaussian Processes for Machine Learning, the MIT Press, 2006, ISBN 026218253X. с 2006 Massachusetts Institute of Technology.
3. Байков В., Бакиров Н., Яковлев А. Математическая геология. Том I.. — 1-е изд. — Ижевск: «Институт компьютерных исследований», 2012. — С. 227. — ISBN 978-5-4344-0053-4.
4. Иан Грэхем. Объектно-ориентированные методы. Принципы и практика = Object-Oriented Methods: Principles & Practice. — 3-е изд. — М.: «Вильямс», 2004. — С. 880. — ISBN 0-201-61913-X.
5. Hands-on Machine Learning with Scikit-Learn, Keras, and Tensorflow, 2nd edition by Aurélien Géron. O' Reilly Media, 2019, 600 pp., ISBN: 978-1-492-03264-9.

						Лист
						19
Изм.	Лист	№ докум.	Подпись	Дата		

ПРИЛОЖЕНИЕ А

UML-диаграмма классов



## ПРИЛОЖЕНИЕ Б

### Исходный код программы

```
import requests
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from statsmodels.tsa.api import VAR
from sklearn.gaussian_process import GaussianProcessRegressor
from sklearn.gaussian_process.kernels import RBF
from PyQt5 import QtWidgets, uic
import sys

class data_class:

    @staticmethod
    def set_start():
        global nyears
        global lags
        global indicators
        global nindicators
        global target_variable
        global countries
        global ncountries
        global start_year
        global end_year
        global data
        global y, x
        global app, window, imp

        nyears = 10
        lags = 5

        indicators = {"gdp": "NY.GDP.MKTP.CD",
                     "population": "SP.POP.TOTL",
                     "inflation": "FP.CPI.TOTL.ZG"}
        nindicators = len(indicators)

        target_variable = "gdp"

        countries = ['us', 'gb']
        ncountries = len(countries)

        # Start and end year for the data set
        start_year = 1976
        end_year = 2019

        app = QtWidgets.QApplication(sys.argv) # Create an instance of
        QtWidgets.QApplication
        window = Ui()
        imp = importData
        data = imp.get_data_from_API()
        y, x = importData.set_logarifmical(data)
```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		21

```

def update_data():
    global data
    global y
    global x
    data = imp.get_data_from_API()
    y, x = importData.set_logarifmical(data)

class decorate:

    def decorator_func(func):
        def wrapper(*args, **kwargs):
            Ui.print_log('Raising function...')
            Ui.print_log('{ } has been raised'.format(func.__name__))
            Ui.print_log('Entering the function...')
            func(*args, **kwargs)
            Ui.print_log('Function has been succeeded')

        return wrapper

class importData(object):

    @staticmethod
    def get_data_from_API():
        template_url = "http://api.worldbank.org/v2/countries/{0}/indi"
        template_url += "cators/{1}?date={2}:{3}&format=json&per_page=999"
        # Countries should be ISO identifiers separated by semi-colon
        raw_data = pd.DataFrame()
        country_str = ';'.join(countries)
        for label, indicator in indicators.items():
            # Fill in the template URL
            url = template_url.format(country_str, indicator,
                                      start_year, end_year)

            # Request the data
            json_data = requests.get(url)

            # Convert the JSON string to a Python object
            json_data = json_data.json()
            json_data = json_data[1]

            for data_point in json_data:
                country = data_point['country']['id']

                # Create a variable for each country and indicator pair
                item = country + '_' + label

                year = data_point['date']

                value = data_point['value']

                # Append to data frame
                new_row = pd.DataFrame([[item, year, value]],
                                       columns=['item', 'year', 'value'])
                raw_data = raw_data.append(new_row)

            upd_data = raw_data.pivot('year', 'item', 'value')

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		22

```

    return upd_data

def set_logarifmical(d):
    # (Runtime warning expected due to NaN)
    d = np.log(d).diff().iloc[1:, :]
    # Set NaN to zero
    d.fillna(0, inplace=True)
    # Subtract the mean from each series
    d = d - d.mean()
    # Convert to date type
    d.index = pd.to_datetime(d.index, format='%Y')
    # Put the target variable into a separate data frame
    target = d[[x for x in data.columns
                  if x.split("_")[-1] == target_variable]]

    return target, d

def count_errors(target):
    ncountries = len(countries)
    errors = target.iloc[-nyears:] - target.shift().iloc[-nyears:]
    # Root mean squared error
    rmse = errors.pow(2).sum().sum() / (nyears * ncountries) ** .5

    # target_data <=== target Var_modeling(set_logarifmical)

class graph:
    @decorate.decorator_func
    def plotting(data):
        for lab in indicators.keys():
            indicator = data[[x for x in data.columns
                              if x.split("_")[-1] == lab]]
            indicator.plot(title=lab)
            plt.show()

class modelling:
    @decorate.decorator_func
    def VAR_modeling(target_data):
        # Sum of squared errors
        sse = 0
        ncountries = len(countries)
        for t in range(nyears):
            # Create a VAR model
            model = VAR(target_data.iloc[t:-nyears + t], freq='AS')

            # Estimate the model parameters
            results = model.fit(maxlags=1)

            actual_values = target_data.values[-nyears + t + 1]

            forecasts = results.forecast(target_data.values[: -nyears + t], 1)
            forecasts = forecasts[0, :ncountries]
            sse += ((actual_values - forecasts) ** 2).sum()
            # Root mean squared error
            rmse = (sse / (nyears * ncountries)) ** .5

        Ui.print_log_result(rmse, 'VAR')

```

						Лист
						23
Изм.	Лист	№ докум.	Подпись	Дата		

```

@decorate.decorator_func
def gaussian_model(target_data, da, a, r):
    ncountries = len(countries)
    gpr = GaussianProcessRegressor(kernel=RBF(r), alpha=a)
    # Number of data points for estimation/fitting for each forecast
    ndata = target_data.shape[0] - nyyears - lags
    # Sum of squared errors
    sse = 0
    for t in range(nyyears):

        # Observations for the target variables
        y = np.zeros((ndata, ncountries))
        # Observations for the independent variables
        X = np.zeros((ndata, lags * ncountries * nindicators))

        for i in range(ndata):
            y[i] = target_data.iloc[t + i + 1]
            X[i] = da.iloc[t + i + 2:t + i + 2 + lags].values.flatten()

        gpr.fit(X, y)

        x_test = np.expand_dims(da.iloc[t + 1:t + 1 +
lags].values.flatten(), 0)
        forecast = gpr.predict(x_test)

        sse += ((target_data.iloc[t].values - forecast) ** 2).sum()
    rmse = (sse / (nyyears * ncountries)) ** .5
    Ui.print_log_result(rmse, 'GPR')

class Ui(QtWidgets.QDialog):
    def __init__(self):
        super(Ui, self).__init__() # Call the inherited classes __init__
method
        uic.loadUi('forecasting.ui', self) # Load the .ui file
        self.logs = self.findChild(QtWidgets.QTextBrowser, 'textBrowser')

        self.spin_box = self.findChild(QtWidgets.QDoubleSpinBox,
'doubleSpinBox')
        self.spin_box.setMinimum(0)
        self.spin_box.setMaximum(100)
        self.spin_box.setValue(0.1)

        self.spin_box_RBF = self.findChild(QtWidgets.QDoubleSpinBox,
'doubleSpinBox_2')
        self.spin_box_RBF.setMinimum(0)
        self.spin_box_RBF.setMaximum(20)
        self.spin_box_RBF.setValue(0.1)

        self.button_GPR = self.findChild(QtWidgets.QPushButton, 'pushButton')
        self.button_GPR.clicked.connect(lambda: window.check_boxes())
        self.button_GPR.clicked.connect(lambda: modelling.gaussian_model(y,
x, self.spin_box.value(), self.spin_box_RBF.value()))

        self.button_build_VAR = self.findChild(QtWidgets.QPushButton,
'pushButton_2')
        self.button_build_VAR.clicked.connect(lambda: window.check_boxes())
        self.button_build_VAR.clicked.connect(lambda:
modelling.VAR_modeling(y))

```

						Лист
						24
Изм.	Лист	№ докум.	Подпись	Дата		



```

        self.button_show = self.findChild(QtWidgets.QPushButton,
'pushButton_3')
        self.button_show.clicked.connect(lambda: window.check_boxes())
        self.button_show.clicked.connect(lambda: graph.plotting(data))

        self.us = self.findChild(QtWidgets.QCheckBox, 'us')
        self.ru = self.findChild(QtWidgets.QCheckBox, 'ru')
        self.au = self.findChild(QtWidgets.QCheckBox, 'au')
        self.ca = self.findChild(QtWidgets.QCheckBox, 'ca')
        self.de = self.findChild(QtWidgets.QCheckBox, 'de')
        self.es = self.findChild(QtWidgets.QCheckBox, 'es')
        self.fr = self.findChild(QtWidgets.QCheckBox, 'fr')
        self.gb = self.findChild(QtWidgets.QCheckBox, 'gb')
        self.jp = self.findChild(QtWidgets.QCheckBox, 'jp')
        self.at = self.findChild(QtWidgets.QCheckBox, 'at')
        self.cn = self.findChild(QtWidgets.QCheckBox, 'cn')
        self.ae = self.findChild(QtWidgets.QCheckBox, 'ae')

        self.show() # Show the GUI

    @staticmethod
    def check_boxes():
        c = []
        global countries
        if window.us.isChecked():
            c.append('us')
        if window.ru.isChecked():
            c.append('ru')
        if window.au.isChecked():
            c.append('au')
        if window.ca.isChecked():
            c.append('ca')
        if window.de.isChecked():
            c.append('de')
        if window.es.isChecked():
            c.append('es')
        if window.fr.isChecked():
            c.append('fr')
        if window.gb.isChecked():
            c.append('gb')
        if window.jp.isChecked():
            c.append('jp')
        if window.at.isChecked():
            c.append('at')
        if window.cn.isChecked():
            c.append('cn')
        if window.ae.isChecked():
            c.append('ae')
        if len(c) < 2:
            countries = ['us', 'gb']
            Ui.print_log('')
            Ui.print_log('')
            Ui.print_log('not enough countries! Entering US and GB as
default...')
            Ui.print_log('')
            Ui.print_log('')
        else:
            countries = c
            data_class.update_data()

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		25

```

def print_log_result(log, method):
    window.logs.append('')
    window.logs.append(str('\n\t' + '-' * 28))
    window.logs.append(str('\t| Error of ' + method + ' : ' +
str(np.round(log, 4)) + '|'))
    window.logs.append(str('\t' + '-' * 28 + '\n'))
    window.logs.append('')

def print_log(log):
    window.logs.append(log)

if __name__ == '__main__':
    data_class.set_start()
    app.exec_() # Start the application

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		26

## ПРИЛОЖЕНИЕ Б

### Исходный код графического интерфейса

```
<?xml version="1.0" encoding="UTF-8"?>
<ui version="4.0">
  <class>Dialog</class>
  <widget class="QDialog" name="Dialog">
    <property name="geometry">
      <rect>
        <x>0</x>
        <y>0</y>
        <width>640</width>
        <height>374</height>
      </rect>
    </property>
    <property name="windowTitle">
      <string>Dialog</string>
    </property>
    <widget class="QLabel" name="label">
      <property name="geometry">
        <rect>
          <x>190</x>
          <y>10</y>
          <width>331</width>
          <height>41</height>
        </rect>
      </property>
      <property name="font">
        <font>
          <pointsize>12</pointsize>
        </font>
      </property>
      <property name="layoutDirection">
        <enum>Qt::LeftToRight</enum>
      </property>
      <property name="text">
        <string>Macroeconomics forecasting</string>
      </property>
    </widget>
    <widget class="QLabel" name="label_2">
      <property name="geometry">
        <rect>
          <x>10</x>
          <y>90</y>
          <width>241</width>
          <height>31</height>
```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		27

```

</rect>
</property>
<property name="text">
  <string>Vector Autoregression method:</string>
</property>
</widget>
<widget class="QTextBrowser" name="textBrowser">
  <property name="geometry">
    <rect>
      <x>380</x>
      <y>70</y>
      <width>256</width>
      <height>192</height>
    </rect>
  </property>
</widget>
<widget class="QPushButton" name="pushButton">
  <property name="geometry">
    <rect>
      <x>290</x>
      <y>160</y>
      <width>75</width>
      <height>23</height>
    </rect>
  </property>
  <property name="text">
    <string>build GPR</string>
  </property>
</widget>
<widget class="QPushButton" name="pushButton_2">
  <property name="geometry">
    <rect>
      <x>180</x>
      <y>100</y>
      <width>75</width>
      <height>23</height>
    </rect>
  </property>
  <property name="text">
    <string>build VAR</string>
  </property>
  <property name="autoExclusive">
    <bool>>false</bool>
  </property>
  <property name="flat">
    <bool>>false</bool>
  </property>
</widget>
<widget class="QLabel" name="label_3">
  <property name="geometry">

```

```

<rect>
  <x>10</x>
  <y>160</y>
  <width>181</width>
  <height>21</height>
</rect>
</property>
<property name="text">
  <string>Gaussian process regression method:</string>
</property>
</widget>
<widget class="QPushButton" name="pushButton_3">
  <property name="geometry">
    <rect>
      <x>100</x>
      <y>250</y>
      <width>75</width>
      <height>23</height>
    </rect>
  </property>
  <property name="text">
    <string>Show </string>
  </property>
</widget>
<widget class="QLabel" name="label_4">
  <property name="geometry">
    <rect>
      <x>100</x>
      <y>220</y>
      <width>91</width>
      <height>31</height>
    </rect>
  </property>
  <property name="text">
    <string>Visualize data</string>
  </property>
</widget>
<widget class="QLabel" name="label_5">
  <property name="geometry">
    <rect>
      <x>10</x>
      <y>40</y>
      <width>591</width>
      <height>31</height>
    </rect>
  </property>
  <property name="text">
    <string>Machine learning based forecasting program. Predicting global GDP, unemployment
rate and inflation ratio</string>
  </property>

```

						Лист
Изм.	Лист	№ докум.	Подпись	Дата		29

```

</widget>
<widget class="QLabel" name="label_6">
  <property name="geometry">
    <rect>
      <x>240</x>
      <y>140</y>
      <width>47</width>
      <height>13</height>
    </rect>
  </property>
  <property name="text">
    <string>Alpha</string>
  </property>
</widget>
<widget class="QDoubleSpinBox" name="doubleSpinBox">
  <property name="geometry">
    <rect>
      <x>220</x>
      <y>160</y>
      <width>62</width>
      <height>22</height>
    </rect>
  </property>
</widget>
<widget class="QLabel" name="label_7">
  <property name="geometry">
    <rect>
      <x>230</x>
      <y>190</y>
      <width>47</width>
      <height>13</height>
    </rect>
  </property>
  <property name="text">
    <string>RBF</string>
  </property>
</widget>
<widget class="QDoubleSpinBox" name="doubleSpinBox_2">
  <property name="geometry">
    <rect>
      <x>220</x>
      <y>210</y>
      <width>62</width>
      <height>22</height>
    </rect>
  </property>
</widget>
<widget class="QWidget" name="gridLayoutWidget">
  <property name="geometry">
    <rect>

```

```

<x>10</x>
<y>290</y>
<width>531</width>
<height>80</height>
</rect>
</property>
<layout class="QGridLayout" name="gridLayout">
  <item row="1" column="3">
    <widget class="QCheckBox" name="jp">
      <property name="text">
        <string>jp</string>
      </property>
    </widget>
  </item>
  <item row="1" column="2">
    <widget class="QCheckBox" name="fr">
      <property name="text">
        <string>fr</string>
      </property>
    </widget>
  </item>
  <item row="1" column="1">
    <widget class="QCheckBox" name="de">
      <property name="text">
        <string>de</string>
      </property>
    </widget>
  </item>
  <item row="0" column="5">
    <widget class="QCheckBox" name="cn">
      <property name="text">
        <string>cn</string>
      </property>
      <property name="checked">
        <bool>true</bool>
      </property>
    </widget>
  </item>
  <item row="1" column="4">
    <widget class="QCheckBox" name="at">
      <property name="text">
        <string>at</string>
      </property>
    </widget>
  </item>
  <item row="1" column="5">
    <widget class="QCheckBox" name="ae">
      <property name="text">
        <string>ae</string>
      </property>
    </widget>
  </item>

```

```

</widget>
</item>
<item row="1" column="0">
  <widget class="QCheckBox" name="ru">
    <property name="text">
      <string>ru</string>
    </property>
  </widget>
</item>
<item row="0" column="0">
  <widget class="QCheckBox" name="us">
    <property name="acceptDrops">
      <bool>>false</bool>
    </property>
    <property name="text">
      <string>us</string>
    </property>
    <property name="checked">
      <bool>>true</bool>
    </property>
  </widget>
</item>
<item row="0" column="1">
  <widget class="QCheckBox" name="ca">
    <property name="text">
      <string>ca</string>
    </property>
    <property name="checked">
      <bool>>true</bool>
    </property>
  </widget>
</item>
<item row="0" column="2">
  <widget class="QCheckBox" name="es">
    <property name="text">
      <string>es</string>
    </property>
    <property name="checked">
      <bool>>true</bool>
    </property>
  </widget>
</item>
<item row="0" column="3">
  <widget class="QCheckBox" name="gb">
    <property name="text">
      <string>gb</string>
    </property>
    <property name="checked">
      <bool>>true</bool>
    </property>
  </widget>
</item>

```



```

</widget>
</item>
<item row="0" column="4">
  <widget class="QCheckBox" name="au">
    <property name="text">
      <string>au</string>
    </property>
    <property name="checked">
      <bool>true</bool>
    </property>
  </widget>
</item>
</layout>
</widget>
<widget class="QLabel" name="label_8">
  <property name="geometry">
    <rect>
      <x>200</x>
      <y>260</y>
      <width>101</width>
      <height>21</height>
    </rect>
  </property>
  <property name="text">
    <string>Data configuration</string>
  </property>
</widget>
</widget>
<resources/>
<connections/>
</ui>

```