Министерство образования и науки Российской Федерации

ФГАОУ ВО «Уральский Федеральный Университет

имени первого Президента России Б. Н. Ельцина»

Институт радиоэлектроники и информационных технологий – РтФ

Оценка

\_\_\_\_\_\_\_\_\_\_\_\_

ИНФОРМАЦИОННАЯ СИСТЕМА ПРЕДСКАЗАНИЯ ПРОДОЛЖИТЕЛЬНОСТИ ЖИЗНИ В ВЕЛИКОБРИТАНИИ  
пояснительная записка к курсовому проекту по модулю

«Методы анализа Big Data»

Дата Подпись

Преподаватель \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ Мирвода С.Г.

Студент \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ Бабкин В.В.

Студент \_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_ Дзираев А.В

Группа РИ-450004

Оглавление

[ВВЕДЕНИЕ 3](#_Toc535463129)

[1 СОЗДАНИЕ МОДЕЛИ ПРЕДСКАЗАНИЙ 4](#_Toc535463130)

[1.1 Описание выбранного набора данных 4](#_Toc535463131)

[1.2 Подготовка данных для построения модели 5](#_Toc535463132)

[1.3 Краткий исследовательский анализ данных 7](#_Toc535463133)

[1.3.1 Продолжительность жизни в зависимости от пола 7](#_Toc535463134)

[1.3.2 Продолжительность жизни в различных районах 8](#_Toc535463135)

[1.4 Создание модели предсказаний 8](#_Toc535463138)

[2 РАЗРАБОТКА ПРИЛОЖЕНИЯ 11](#_Toc535463139)

[ЗАКЛЮЧЕНИЕ 12](#_Toc535463140)

[ПРИЛОЖЕНИЕ А. Код MODELLEARNING.R 13](#_Toc535463141)

[ПРИЛОЖЕНИЕ Б. КОД MAINWINDOW.XAML.CS 14](#_Toc535463142)

[ПРИЛОЖЕНИЕ В. КОД MAINWINDOW.XAML. 15](#_Toc535463143)

[ПРИЛОЖЕНИЕ Г. КОД RDOTNETCALLET.CS 16](#_Toc535463144)

[БИБЛИОГРАФИЧЕСКИЙ СПИСОК 17](#_Toc535463144)

# ВВЕДЕНИЕ

Целью выполнения курсового проекта является приобретение практических навыков анализа данных с применением статистических методов и методов машинного обучения.

В соответствии с поставленной целью можно выделить следующие задачи:

1. Выбрать набор данных для анализа;
2. Описать выбранный набор данных;
3. Провести краткий исследовательский анализ данных;
4. Сформулировать гипотезу;
5. Подобрать к данным модель;
6. Обучить модель, проверить её работоспособность и доказать её статистическую значимость;
7. Визуализировать модель и качество прогноза;
8. Разработать приложение на любом языке программирования, на любой платформе, подключить разработанную модель к приложению. Приложение по запрашиваемым параметрам выполняет прогноз и возвращает результат пользователю.

# СОЗДАНИЕ МОДЕЛИ ПРЕДСКАЗАНИЙ

## 1.1 Описание выбранного набора данных

Для создания информационной системы выбран набор данных средней продолжительности жизни в Великобритании за 2009-2015 годы по географическому составу, полу и индексу массы тела.

Ссылка на данные – <https://data.gov.uk/dataset/da14c5aa-cc27-48ea-9af5-7bbfda43ee95/life-expectancy-in-leeds-3-year-averages>.

Данные состоят из следующего набора переменных (столбцов):

* Профессия;
* Пол;
* Регион;
* Средняя зарплата за 2005 год;
* Средняя зарплата за 2007 год;
* Средняя зарплата за 2009 год;
* Средняя зарплата за 2011 год;
* Средняя зарплата;

Найдена краткая сводка по составу данных (см. рис. 1).

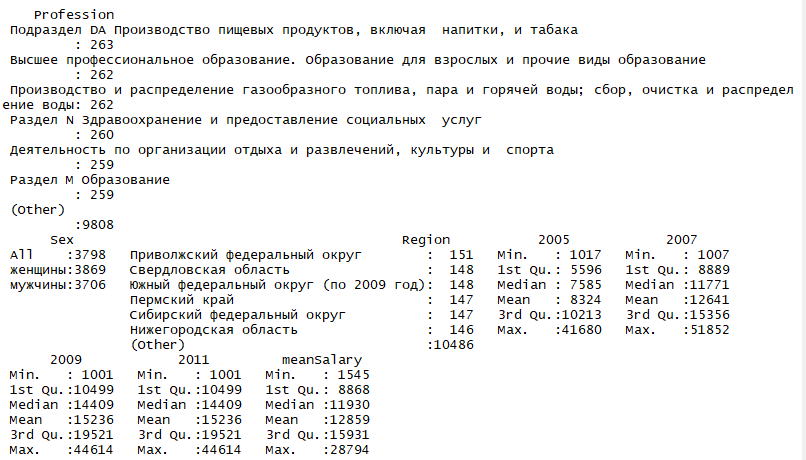


Рисунок 1 – Сводка по выбранному набору данных

## 1.2 Подготовка данных для построения модели

Для построения модели предсказания потребуются 4 переменные: «Профессия», «Пол», «Регион» и «Средняя зарплата». Переменная «Средняя зарплата» получена с помощью переменных “2005”, “2007”, “2009”, “2011” (см. рис. 2).

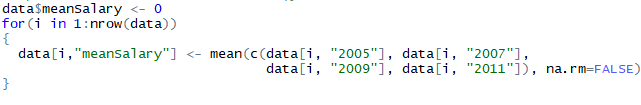


Рисунок 2 – Получение колонки meanSalary

Типы данных в датасете (см. рис. 3).

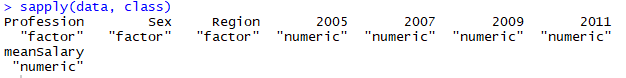


Рисунок 3 – Типы данных переменных

Выполнена проверка данных выбросы и последующее избавление от них (см. рис. 4, 5 и 6).

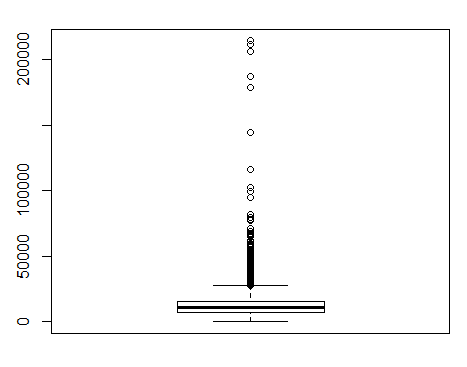


Рисунок 4 – Boxplot(data$meanSalary) до избавления от выбросов

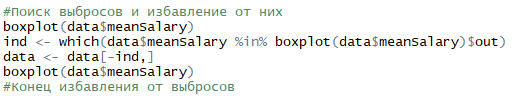


Рисунок 5 – Код избавления от выбросов

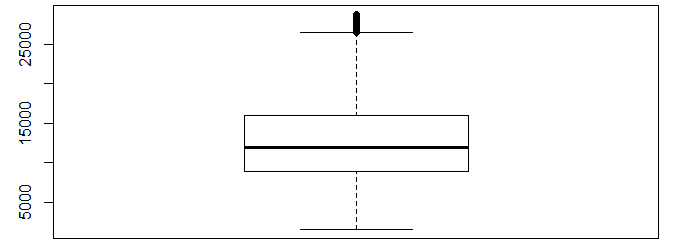


Рисунок 6 – Boxplot(avgLifeExpectancy) после избавления от выбросов

## 1.3 Краткий исследовательский анализ данных

Проведен краткий исследовательский анализ данных:

1.3.1 Средняя зарплата в зависимости от пола:

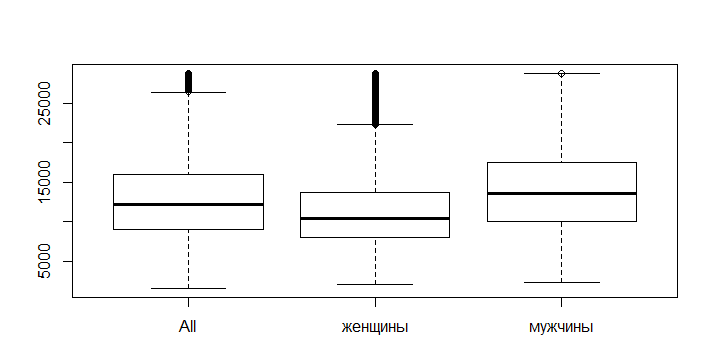


Рисунок 7 – Средняя зарплата в зависимости от пола

Как мы видим, в среднем по России зарплата мужчин выше, чем у женщин.

### 1.3.2 Сравнение зарплат в 2011 и в 2005 годах:

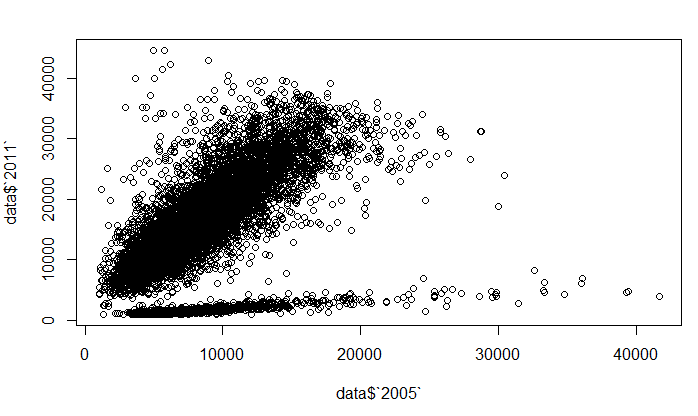


Рисунок 8 – Сравнение зарплат в 2011 и в 2005 годах

Как мы видим, зарплата в 2011 году выросла в сравнении с 2005 годом.

## 1.4 Создание линейной модели предсказания:

### Создадим линейную модель, используя колонки avgLifeExpectancy, AreaType, Sex и BMI. Код на рисунке 9.

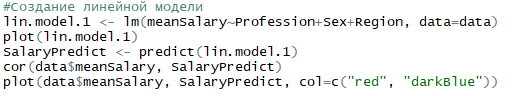


Рисунок 9 – Код создания модели данных

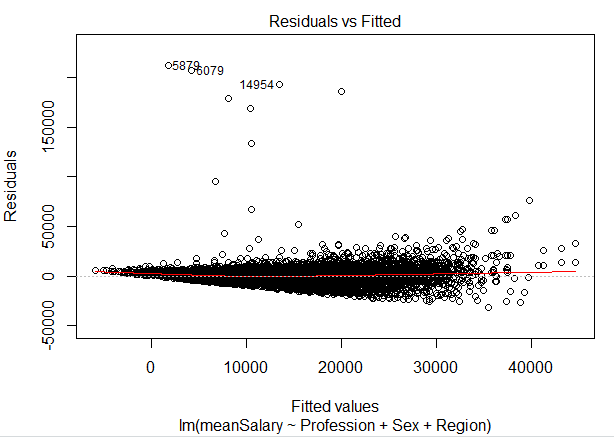


Рисунок 10 – График линейной модели Residuals vs Fitted

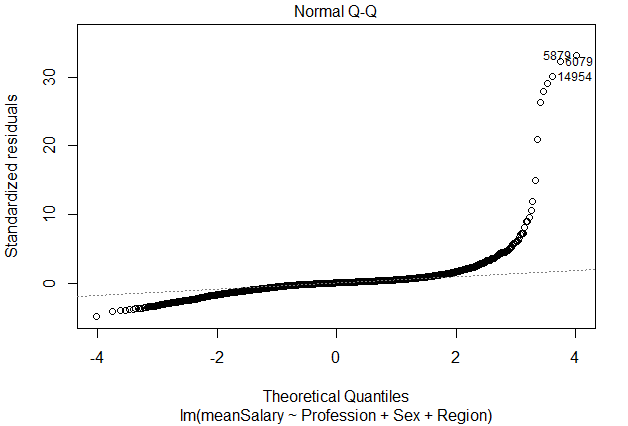


Рисунок 11 – График линейной модели Normal Q-Q

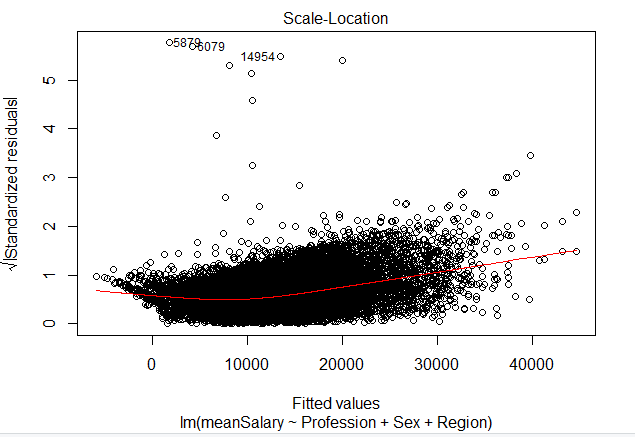


Рисунок 12 – График линейной модели Scale-Location

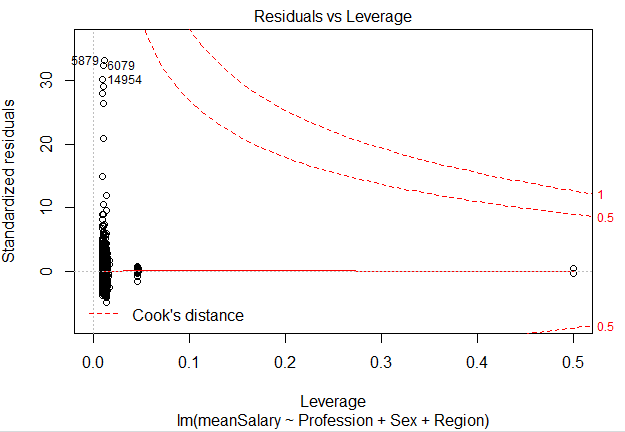


Рисунок 13 – График линейной модели Residuals vs Leverage

Коэффициент корреляции получаем равный 0.6900233 (см. рис. 14).



Рисунок 14 – Коэффициент корреляции

График сравнения предсказанных значений и фактических (см. рис. 15).

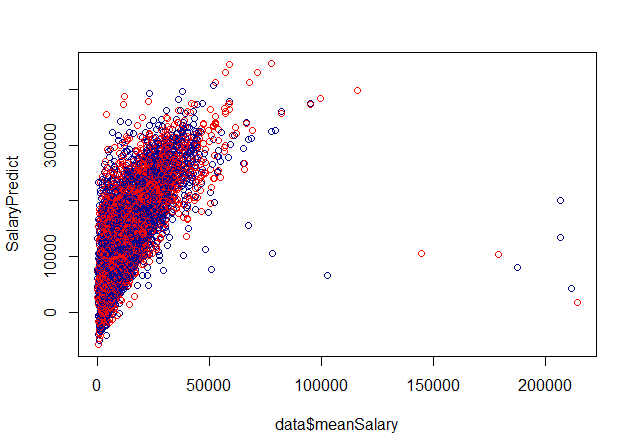


Рисунок 15 – Сравнение предсказанных значений и исходных

# Разработка ПРИЛОЖЕНИЯ

Разработаем приложение, которое будет использовать нашу модель linear.model.1.

Сервис позволяет конечному пользователю вводить данные о месте, где он собирается жить (или уже живет) и о себе – тип зоны, пол, рост и вес.

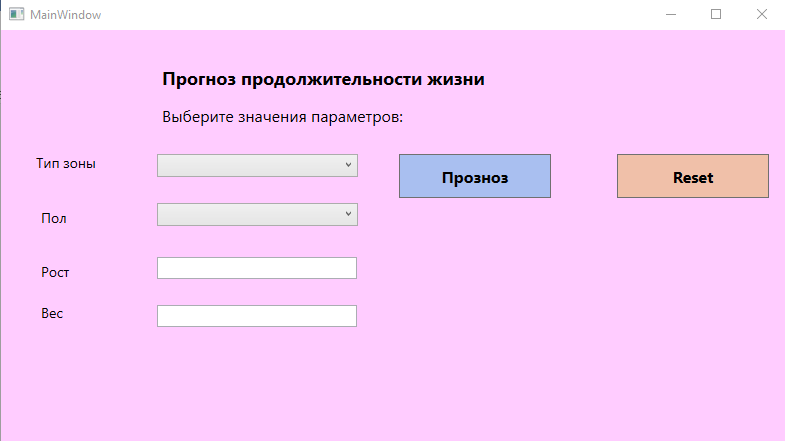


Рисунок 16 – Экран выбора данных для модели

Рост и вес, которые вводит пользователь используются для расчета индекса массы тела (BMI), который будет передан модели данных (см. рис. 17).



Рисунок 17 – Код расчет BMI

Когда нужно получить результат приложение обращается к модели и получает предсказание о продолжительности жизни, основываясь на данных о типе зоне, поле и индексе массы тела пользователя.

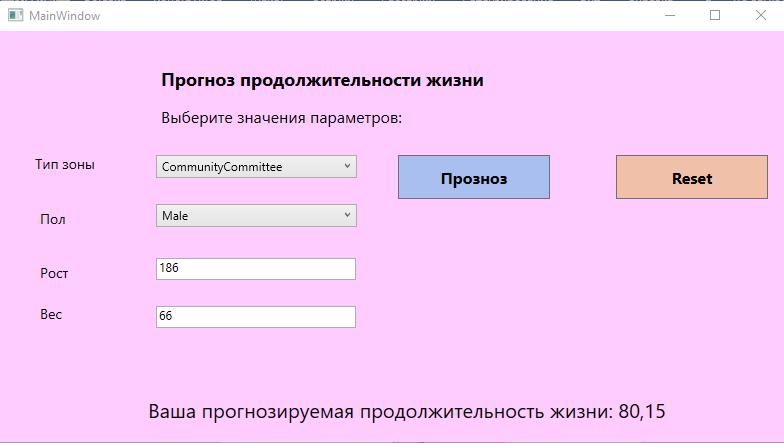


Рисунок 17 – Экран результата выполнения модели

# ЗАКЛЮЧЕНИЕ

В результате выполнения данного курсового проекта была создана модель данных о продолжительности жизни в Великобритании в зависимости от типа зоны проживания, пола и индекса массы тела.

В ходе выполнения работы было проведено краткое описание набора данных, проведен краткий исследовательский анализ данных, была подобрана линейная модель. Получен коэффициент корреляции и построено сравнение прогнозируемых данных и фактических. Затем было разработано приложение с помощью Visual Studio WPF и к нему была подключена разработанная модель, с помощью которой пользователь приложения может получать прогноз.

# Приложение А. Код LearnMODEL.r

data <- read.csv('DataSet.csv', header=TRUE, sep=',')

#Обработка данных

for(i in 1:nrow(data))

{

data[i, "avgLifeExpectancy"] <- mean(data[i, "X2009\_LifeExpectancy"], dta[i,"X2010\_LifeExpectancy"],

data[i,"X2011\_LifeExpectancy"], data[i,"X2012\_LifeExpectancy"],

data[i,"X2013\_LifeExpectancy"], data[i,"X2014\_LifeExpectancy"],

data[i,"X2015\_LifeExpectancy"])

}

#Конец обработки данных

#boxplot до избавления от выбросов

boxplot(data$avgLifeExpectancy)

#Избавимся от выбросов

ind <- which(data$avgLifeExpectancy %in% boxplot(data$avgLifeExpectancy)$out)

data <- data[-ind,]

data <- data[data$avgLifeExpectancy > 75, ]

#boxplot после избавления от выбросов

boxplot(data$avgLifeExpectancy)

#Создание линейной модели

linear.model.1 <- lm(avgLifeExpectancy ~ AreaType+Sex+BMI, data=data)

plot(linear.model.1)

lifePredict <- predict(linear.model.1)

cor(data$avgLifeExpectancy, lifePredict)

plot(data$avgLifeExpectancy, lifePredict, col=c("red", "darkBlue"))

# ПРИЛОЖЕНИЕ Б. КОД MainWindow.xaml.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using System.Collections.ObjectModel;

using RDotNet;

using System.IO;

namespace LifeExpectancy

{

public partial class MainWindow : Window

{

public MainWindow()

{

InitializeComponent();

}

public bool CheckTextBoxes()

{

return Height.Text.Length != 0 && AreaType.Text.Length != 0 &&

ComboBoxSex.Text.Length != 0 && Weight.Text.Length != 0;

}

private void Clear\_Click(object sender, RoutedEventArgs e)

{

Weight.Text = string.Empty;

Height.Text = string.Empty;

AreaType.Text = string.Empty;

ComboBoxSex.Text = string.Empty;

labelResult.Content = string.Empty;

}

private void Button\_Click(object sender, RoutedEventArgs e)

{

StartupParameter rinit = new StartupParameter();

RdotNetCaller rdotNetCaller = new RdotNetCaller("Model.r");

var BMI = (Convert.ToDouble(Weight.Text) / (((Convert.ToDouble(Height.Text)\*0.01) \* (Convert.ToDouble(Height.Text)\*0.01)))).ToString();

if (!CheckTextBoxes())

MessageBox.Show("Введены не все значения параметров!!!");

else

{

var result = rdotNetCaller.CallMyModel(AreaType.Text, ComboBoxSex.Text, BMI).Replace('.', ',');

var convertedResult = Convert.ToDouble(result);

var roundedResult = Math.Round(convertedResult, 2);

result = roundedResult.ToString();

labelResult.Content = string.Format("Ваша прогнозируемая продолжительность жизни: {0}", result);

}

}

}

}

# ПРИЛОЖЕНИЕ В. КОД MainWindow.xaml

<Window x:Class="LifeExpectancy.MainWindow"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"

xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

xmlns:d="http://schemas.microsoft.com/expression/blend/2008"

xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"

xmlns:local="clr-namespace:LifeExpectancy"

mc:Ignorable="d"

Title="MainWindow" Height="450" Width="800">

<Grid Background="#FFCCFF">

<Button x:Name="Clear" Content="Reset" HorizontalAlignment="Left" Margin="616,124,0,0" VerticalAlignment="Top" Width="152" RenderTransformOrigin="1.547,8.222" Click="Clear\_Click" FontSize="16" FontWeight="Bold" Background="#FFF0C0A9" Height="44"/>

<Label x:Name="label1" Content="Прогноз продолжительности жизни" HorizontalAlignment="Left" Margin="156,31,0,0" VerticalAlignment="Top" RenderTransformOrigin="5.867,0.684" FontSize="18" Background="{x:Null}" FontWeight="Bold"/>

<Label Content="Тип зоны" HorizontalAlignment="Left" Margin="30,118,0,0" VerticalAlignment="Top" FontSize="14"/>

<Label Content="Рост" HorizontalAlignment="Left" Margin="35,227,0,0" VerticalAlignment="Top" RenderTransformOrigin="1.096,1.23" FontSize="14"/>

<Label Content="Пол" HorizontalAlignment="Left" Margin="35,173,0,0" VerticalAlignment="Top" FontSize="14" RenderTransformOrigin="0.472,-0.276"/>

<Label Content="Выберите значения параметров:" HorizontalAlignment="Left" Margin="156,70,0,0" VerticalAlignment="Top" FontSize="16"/>

<ComboBox x:Name="AreaType" HorizontalAlignment="Left" Height="23" Margin="156,124,0,0" VerticalAlignment="Top" Width="201">

<TextBlock>CommunityCommittee</TextBlock>

<TextBlock>Deprivation</TextBlock>

<TextBlock>Leeds Resident</TextBlock>

<TextBlock>Ward</TextBlock>

</ComboBox>

<TextBox x:Name="Height" HorizontalAlignment="Left" Height="22" Margin="156,227,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="200" RenderTransformOrigin="0.495,-4.022"/>

<ComboBox x:Name="ComboBoxSex" HorizontalAlignment="Left" Height="23" Margin="156,173,0,0" VerticalAlignment="Top" Width="201">

<TextBlock>Female</TextBlock>

<TextBlock>Male</TextBlock>

<TextBlock>Person</TextBlock>

</ComboBox>

<Button x:Name="Evaluate" Content="Прозноз" HorizontalAlignment="Left" Margin="398,124,0,0" VerticalAlignment="Top" Width="152" RenderTransformOrigin="1.547,8.222" Click="Button\_Click" FontSize="16" Background="#FFA9BFF0" FontWeight="Bold" Height="44"/>

<Label x:Name="labelResult" Content="" HorizontalAlignment="Center" Margin="30,360,0,0" VerticalAlignment="Top" FontSize="20"/>

<Label Content="Вес" HorizontalAlignment="Left" Margin="35,268,0,0" VerticalAlignment="Top" RenderTransformOrigin="1.096,1.23" FontSize="14"/>

<TextBox x:Name="Weight" HorizontalAlignment="Left" Height="22" Margin="156,275,0,0" TextWrapping="Wrap" VerticalAlignment="Top" Width="200" RenderTransformOrigin="0.495,-4.022"/>

</Grid>

</Window>

# Приложение Г. код RdotNetCaller.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using RDotNet;

using System.IO;

using System.Collections.ObjectModel;

namespace LifeExpectancy

{

class RdotNetCaller

{

private REngine engine;

public RdotNetCaller(String scriptPath)

{

var oldPath = Environment.GetEnvironmentVariable("PATH");

var rPath1 = @"C:\Program Files\Microsoft\R Client\R\_SERVER\bin\x64";

var rPath2 = @"C:\Program Files\Microsoft\R Client\R\_SERVER\library\caret\R";

if (!Directory.Exists(rPath1))

throw new DirectoryNotFoundException(string.Format(" R.dll not found in : {0}", rPath1));

var newPath = string.Format("{0}{1}{2}", rPath1, Path.PathSeparator, oldPath);

newPath = string.Format("{0}{1}{2}", rPath2, Path.PathSeparator, newPath);

Environment.SetEnvironmentVariable("PATH", newPath);

REngine.SetEnvironmentVariables(

"C:\\Program Files\\R\\R-3.4.4\\bin\\i386",

"C:\\Program Files\\R\\R-3.4.4"

);

engine = REngine.GetInstance();

engine.Initialize();

engine.Evaluate($"source('{scriptPath}')");

}

public string CallMyModel(params string[] parameters)

{

engine.Evaluate(string.Format($"parameters <- data.frame('{parameters[0].ToString()}', '{parameters[1].ToString()}', {parameters[2].ToString()})"));

engine.Evaluate("names(parameters) <- c('AreaType','Sex','BMI')");

engine.Evaluate("predictions <- predict(linear.model.1, parameters)");

return engine.Evaluate("as.character(predictions)").AsCharacter().ToArray()[0];

}

}

}

# БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Оpen data platform [Электронный ресурс]. Доступен по ссылке – https://data.gov.uk/.
2. Big Data от А до Я. Часть 1: Принципы работы с большими данными, парадигма MapReduce [Электронный ресурс]. Доступен по ссылке – <https://helpiks.org/6-56149.html>.
3. Создание модели машинного обучения [Электронный ресурс]. Доступен по ссылке – https://aws.amazon.com/ru/getting-started/projects/build-machine-learning-model/