

Документ подписан простой электронной подписью
Информация о владельце:
ФИО: Клочков Юрий Сергеевич
Должность: Ректор
Дата подписания: 02.10.2026 11:11:35
Уникальный программный ключ:
3beb265d5d589e7ff4c954946f3ad99a1e70ac12

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра кибернетических систем

Методические рекомендации

по выполнению курсовой работы
по дисциплине «**Технологии интеллектуального анализа BigData**»
для обучающихся по направлению подготовки
09.03.02 – Информационные системы и технологии
профиль Искусственный интеллект и программирование

Составитель:
Лаптева У.В.,
старший преподаватель

Тюмень 2022

Методические рекомендации по выполнению курсовой работы по дисциплине «Технологии интеллектуального анализа BigData» для обучающихся по 09.03.02 – Информационные системы и технологии профиль Искусственный интеллект и программирование [Текст] / сост. У.В. Лаптева. Тюменский индустриальный университет – 1-е изд.– Тюмень: Издательский центр БИК ТИУ – 43с.

Методические рекомендации рассмотрены и рекомендованы к изданию на заседании кафедры Кибернетических систем «29» августа 2022 года, протокол № 1.

Аннотация

Методические рекомендации по выполнению курсовой работы по дисциплине «Технологии интеллектуального анализа BigData» предназначены для обучающихся направления 09.03.02 – Информационные системы и технологии, профиль Искусственный интеллект и программирование.

Цель курсовой работы заключается в закреплении, углублении и систематизации полученных обучающимися теоретических знаний по дисциплине «Технологии интеллектуального анализа BigData», усвоении практических навыков по применению технологий интеллектуального анализа данных.

Методические рекомендации содержат требования к выполнению курсовой работы, критерии оценки работы, структуру пояснительной записки и правила ее оформления, пример выполнения работы и индивидуальные задания.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
Правила выбора темы курсовой работы	5
Типовая структура курсовой работы и требования к содержанию ее разделов. 7	
Рекомендации по выполнению отдельных разделов курсовой работы	8
Примерные нормы времени на выполнение работы	30
Порядок защиты работы, критерии оценки.....	31
Список рекомендуемой литературы.....	33
Требования к оформлению работы	34
Приложение А	40
Приложение Б	41
Приложение В.....	42

ВВЕДЕНИЕ

Курсовая работа по дисциплине «Технологии интеллектуального анализа BigData» является важным элементом учебного процесса, предусмотрена учебным планом подготовки бакалавров по направлению 09.03.02 – Информационные системы и технологии, профиль Искусственный интеллект и программирование.

Курсовая работа является формой самостоятельной работы: студент показывает свое умение работать с образовательными ресурсами, в том числе электронными, для получения требуемых знаний в области интеллектуального анализа данных; собирать, анализировать и грамотно использовать информацию из найденных источников; логически и последовательно излагать свои мысли; делать правильные выводы и предложения.

Цель курсовой работы – формирование у будущих специалистов практических навыков по основам интеллектуального анализа данных, применения современных информационных технологий интеллектуального анализа данных для решения экономических, управленческих и других прикладных задач, развитие умения работы с современными программными средствами, обучение работе с научно–технической литературой и технической документацией по программному обеспечению.

Задачи курсовой работы – реализация требований, установленных в квалификационной характеристике в области анализа, создания, внедрения, сопровождения и применения средств математического обеспечения информационных систем предметной области.

В результате выполнения курсовой работы обучающийся демонстрирует знание методов интеллектуального анализа данных – технологии BI, OLAP, Data Mining; средства визуального анализа данных; технологии Text–Mining, Web–Mining.

Правила выбора темы курсовой работы

Задание на курсовую работу выдается преподавателем из списка предлагаемых тем.

В случае необходимости в задание преподавателем могут быть внесены изменения и дополнения, согласованные с обучающимся. Выбор нового задания при этом производится из предлагаемой тематики курсовых проектов.

Примерная тематика курсовых работ:

1. Прогнозирование возврата персональных ссуд в инвестиционно-кредитной компании Lending Club (дата фрейм loan200 из пакета FNN языка R)

2. Задача кредитного скоринга по данным о 1000 клиентов одного из немецких банков (дата фрейм GermanCredit из пакета caret языка R)

3. Оценивание стоимости домов на основе данных о жилом округе Кинг (Сэтл, шт. Вашингтон) (дата фрейм data.frame house в базовом оснащении языка R)

4. Задача об обнаружении спама по коллекции данных Центра машинного обучения и интеллектуальных систем Калифорнийского университета <http://archive.ics.uci.edu/ml/>

5. Использование результатов гидробиологических исследований обилия водорослей в различных реках (дата фрейм algae из пакета DMwR языка R)

6. Изучение семейного статуса 2000 швейцарцев на протяжении 16 лет их жизни между 15 и 30 годами (дата фрейм biofam из пакета TraMineR языка R)

7. Определение способа изготовления стекла по его химическому составу (дата фрейм fgl из пакета MASS языка R)

8. Определение возраста морского ушка на данных о 4177 особях этого вида. Данные берутся из коллекции данных Центра машинного обучения и интеллектуальных систем Калифорнийского университета <http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data> Гео-ботаника:

9. Анализ зависимости между показателями химического состава и обилием 44 видов травянистых растений на 24 пробных площадках (дата фреймы varechem и varespres из пакета vegan языка R)

10. Задача анализа и моделирования данных в различных предметных областях. На данных, найденных студентами и одобренных преподавателем.

11. Свободная тема, связанная с инициативным исследованием анонимных данных студентов. Предполагает первичную постановку задачи (например, изучить влияние занятий спортом на успеваемость в учебе) включая набор исследовательских гипотез. Самостоятельное определение предикторов и отклика, формирование анкеты для опроса студентов и сбора прямых или косвенных данных для оценки предикторов и отклика, анкетирование (анонимное), обработка данных и формирование дата фрейма. Разведочный анализ данных, использование дата фрейма для обучения и тестирования моделей, интерпретация

результатов разведочного анализа данных и полученных моделей в контексте проверяемых гипотез.

Выбор этой темы является предпочтительным в связи с необходимостью реализации всего конвейера работ исследовательского проекта анализа данных.

Все темы подразумевают широкую вариабельность поставленных и решаемых задач. В процессе выполнения курсовой работы студентам необходимо продемонстрировать различные подходы к решению поставленных задач на основе широкого класса методов и моделей машинного обучения. Особенно высоко оценивается своеобразие постановки задачи и оригинальность ее решения, состоящее в уникальной последовательности операций преобразования и моделирования данных

Типовая структура курсовой работы и требования к содержанию ее разделов

Типовая структура курсовой работы:

- титульный лист (приложение А)
- содержание;
- введение;
- основная часть;
- заключение;
- список литературы;
- приложения.

Структура основной части курсовой работы:

1. Изучение предметной области и постановка задачи (регрессии, классификации, кластеризации).
2. Сбор и обработка данных.
3. Разведочный анализ данных (качественная визуализация данных и осмысление графиков, вычисление описательных статистик, и проч.).
4. Построение статистических моделей и алгоритмов их машинного обучения, включая разработку искусственных нейронных сетей.
5. Интерпретация и использование модели машинного обучения (концепция продукта данных или принятие решения на основе обобщения данных).

Рекомендации по выполнению отдельных разделов курсовой работы

В данном разделе приводятся фрагмент примера выполнения курсовой работы.

Тема: Анализ данных отражающих рейтинг самых счастливых стран.

Содержание расчетно–пояснительной записки: подготовка данных для исследования, разведочный анализ, регрессионный анализ, кластеризация и дерево решений, создание и обучение RNN модели

Подготовка данных для исследования.

Для того чтобы изучить и проанализировать данную тему, мы использовали пять дата сетов за 2015–2019 года, которые были представлены в формате .CSV.

```
df2015 = pd.read_csv("2015.csv") df2016 = pd.read_csv("2016.csv")
df2017 = pd.read_csv("2017.csv") df2018 = pd.read_csv("2018.csv") df2019 =
pd.read_csv("2019.csv")
```

Прежде, чем начать исследование, необходимо было подготовить данные для дальнейшей работы с ними. Для этого были выполнены следующие действия/операции:

Удаление ненужных столбцов в датасетах

```
df2015_figures = df2015.drop(['Country', 'Region', 'Standard Er-
ror', 'Dystopia Residual', 'Happiness Rank'], axis=1) df2016_figures =
df2016.drop(['Country', 'Region', 'Lower Confidence Inter-val', 'Upper Confi-
dence Interval', 'Dystopia Residual', 'Happiness Rank'], axis=1)
df2017_figures =
df2017.drop(['Country', 'Happiness.Rank', 'Whisker.high', 'Whisker.low', 'Dystop
ia.Residual'], axis=1) df2018_figures = df2018.drop(['Country or region',
'Overall rank'], axis=1) df2019_figures = df2019.drop(['Country or region',
'Overall rank'], axis=1)
```

Приведение названия столбцов к одному виду

```
df2015_figures = df2015_figures[['Happiness Score', 'Economy (GDP per
Capi-ta)', 'Family', 'Health (Life Expectancy)', 'Freedom', 'Generosity', 'T
rust (Government Corruption)']] df2016_figures = df2016_figures[['Happiness
Score', 'Economy (GDP per Capi-ta)', 'Family', 'Health (Life Expectancy)',
'Freedom', 'Generosity', 'T rust (Government Corruption)']]
df2015_figures.rename(columns={'Happiness Score': 'Score', 'Economy (GDP per
Capita)': 'GDP per capita', 'Family': 'Social support', 'Healt h (Life Ex-
pectancy)': 'Healthy life expectancy', 'Freedom': 'Freedom to make life
choices', 'Trust (Government Corruption)': 'Perceptions of co rruption'},
inplace=True) df2016_figures.rename(columns={'Happiness Score': 'Score',
'Economy (GDP per Capita)': 'GDP per capita', 'Family': 'Social support',
'Healt h (Life Expectancy)': 'Healthy life expectancy', 'Freedom': 'Freedom
to make life choices', 'Trust (Government Corruption)': 'Perceptions of co
rruption'}, inplace=True) df2017_figures.rename(columns={'Happiness.Score':
'Score', 'Economy.
.GDP.per.Capita.': 'GDP per capita', 'Family': 'Social support',
'Healt h..Life.Expectancy.': 'Healthy life expectancy', 'Freedom': 'Freedom
```

```
to make life choices', 'Trust..Government.Corruption.': 'Perceptions of co
rruption'}}, inplace=True)
```

Объединение всех датасетов

```
df_all = df2019_figures.append(df2018_figures, ignore_index=True)
df_all = df_all.append(df2017_figures, ignore_index = True) df_all =
df_all.append(df2016_figures, ignore_index = True) df_all =
df_all.append(df2015_figures, ignore_index = True)
```

Сортировка по столбцу «Score»

```
df_all = df_all.sort_values(by=['Score'], ascending=False, ignore_ind
ex = True)
```

Далее на рисунке 1 представлен готовый для исследования датасет с дан-
ными за 2015–2019 года

	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
0	7.769	1.340000	1.587000	0.986000	0.596000	0.153000	0.393000
1	7.632	1.305000	1.592000	0.874000	0.681000	0.202000	0.393000
2	7.600	1.383000	1.573000	0.996000	0.592000	0.252000	0.410000
3	7.594	1.456000	1.582000	0.861000	0.686000	0.286000	0.340000
4	7.587	1.396510	1.349510	0.941430	0.665570	0.296780	0.419780
...	...	...	...	...	...	...	...
777	2.905	0.015300	0.415870	0.223960	0.118500	0.197270	0.100620
778	2.905	0.091623	0.629794	0.151611	0.059901	0.204435	0.084148
779	2.853	0.306000	0.575000	0.295000	0.010000	0.202000	0.091000
780	2.839	0.208680	0.139950	0.284430	0.364530	0.166810	0.107310
781	2.693	0.000000	0.000000	0.018773	0.270842	0.280876	0.056565

782 rows × 7 columns

Рисунок 1 – Данные 2015–2019 года

Разведочный анализ.

Разведочный анализ – это предварительный анализ данных с целью выяв-
ления наиболее общих зависимостей, закономерностей и тенденций, характера
и свойств анализируемых данных, законов распределения анализируемых вели-
чин. Применяется для нахождения связей между переменными в ситуациях, ко-
гда отсутствуют (или недостаточны) априорные представления о природе этих
связей.

Для проведения анализа были выполнены следующие действия:

Сравнение уровня счастья с 2015 года по 2019 год

Нашли средние значения Score за каждый год

Создали дата сет с колонками «Year» и «value»

```
ScoreYears = pd.DataFrame(['2015', df2015_figures['Score'].mean()],
['2016', df2016_figures['Score'].mean()], ['2017', df2017_figures['Sco
re'].mean()], ['2018', df2018_figures['Score'].mean()], ['2019',
df2019_figures['Score'].mean()]), columns=['Years', 'value'])
ScoreYears.set_index('Years', inplace=True)
```

Созданный датасет:

Years	value
2015	5.375734
2016	5.382185
2017	5.354019
2018	5.375917
2019	5.407096

На рисунке 2 представлен график сравнения уровня счастья за 2015–2019 года.



Рисунок 2 – График уровней счастья за 2015–2019 г

Как мы видим на данном графике, изменение среднего значения Score незначительно. В то же время мы можем сделать вывод, что уровень счастья в 2017-ом году был наименьшим, а в 2019-ом году достиг максимального значения.

Корреляционная матрица.

Одним из основных методов разведочного анализа является корреляционный анализ с целью поиска коэффициентов.

Корреляционная матрица – это квадратная (или прямоугольная) таблица, в которой на пересечении соответствующих строки и столбца находится коэффициент корреляции между соответствующими параметрами.

На рисунке 3 изображена корреляционная матрица исследуемых параметров

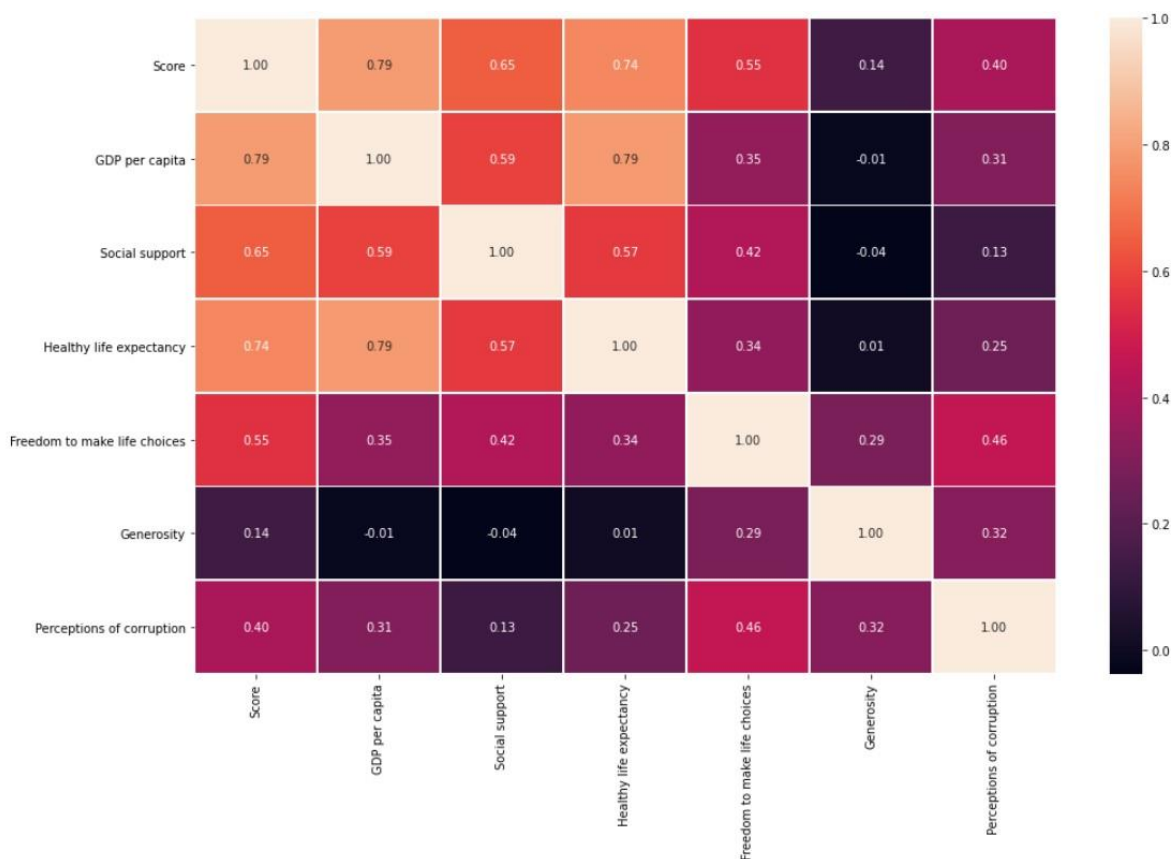


Рисунок 3 – Корреляционная матрица

Соотношение исследуемых параметров

Для того чтобы проанализировать, как влияют некоторые критерии на число Score, мы построили графики соотношений.

График соотношения процента счастья к ВВП
(Score vs GDP per capita)

```
sns.jointplot(x="Score", y="GDP per capita", data=df_all, kind="hex")
```

График соотношения процента счастья к щедрости (Score vs Generosity)

```
sns.jointplot(x="Score", y="Generosity", data=df_all, kind="hex")
```

На рисунках 4 и 5 представлены графики соотношений Score vs Generosity и Score vs GDP per capita

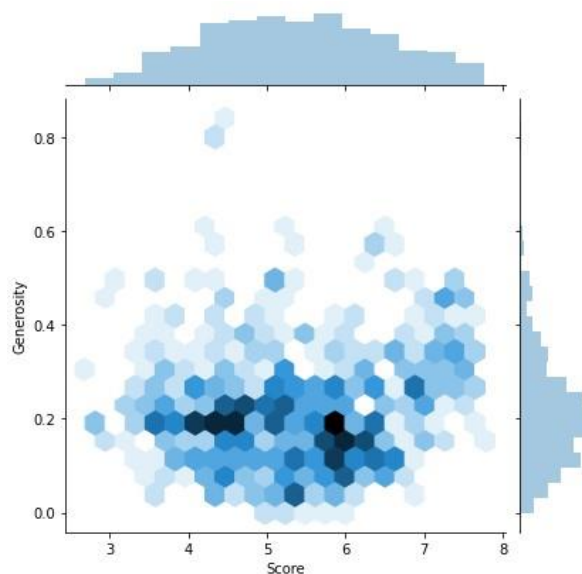


Рисунок 4 – График соотношения Score vs Generosity

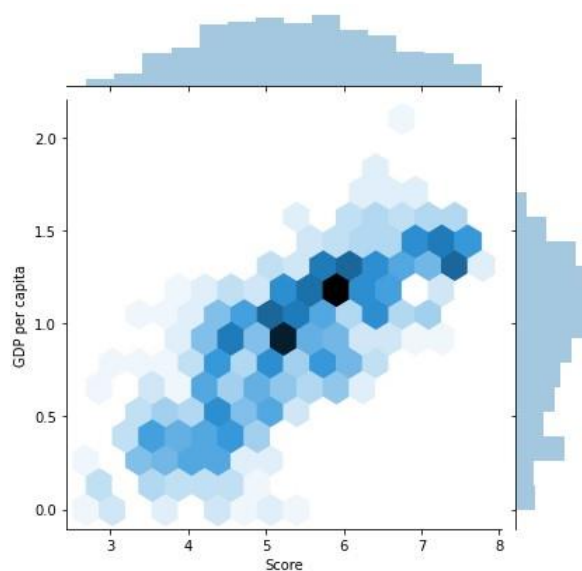


Рисунок 5 – График соотношения Score vs GDP per capita

График соотношения процента счастья к социальной поддержке (Score vs Social support)

```
sns.jointplot(x="Score", y="Social support", data=df_all, kind="hex")
```

График соотношения процента счастья к продолжительности жизни (Score vs Healthy life expectancy)

```
sns.jointplot(x="Score", y="Healthy life expectancy", data=df_all, kind="hex")
```

На рисунках 6 и 7 представлены графики соотношений Score vs Social support и Score vs Healthy life expectancy

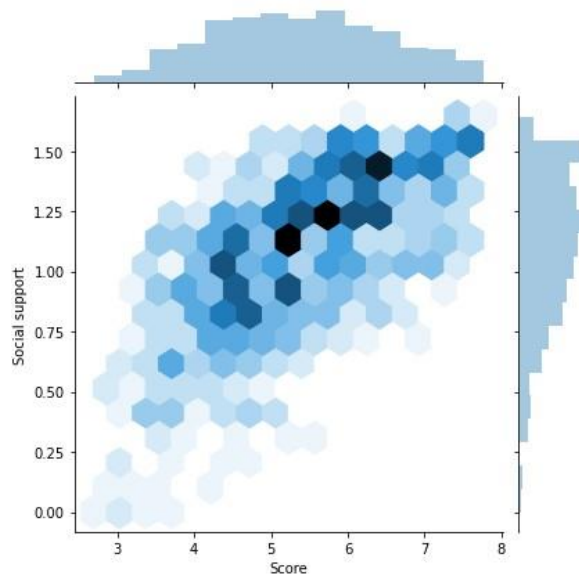


Рисунок 6 – График соотношения Score vs Social support

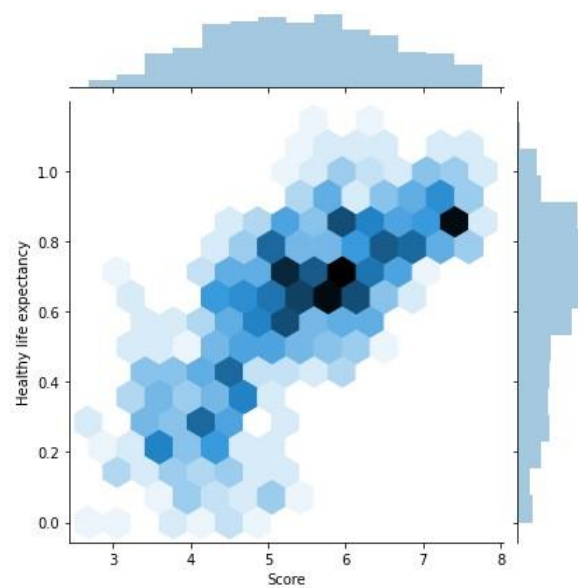


Рисунок 7 – График соотношения Score vs Healthy life expectancy

График соотношения процента счастья к продолжительности жизни (Score vs Freedom to make life choices)

```
sns.jointplot(x="Score", y="Freedom to make life choices", data=df_all, kind="hex")
```

График соотношения процента счастья к продолжительности жизни (Score vs Perceptions of corruption)

```
sns.jointplot(x="Score", y="Perceptions of corruption", data=df_all, kind="hex")
```

На рисунках 8 и 9 представлены график соотношений Score vs Freedom to make life choices и Score vs Perceptions of corruption

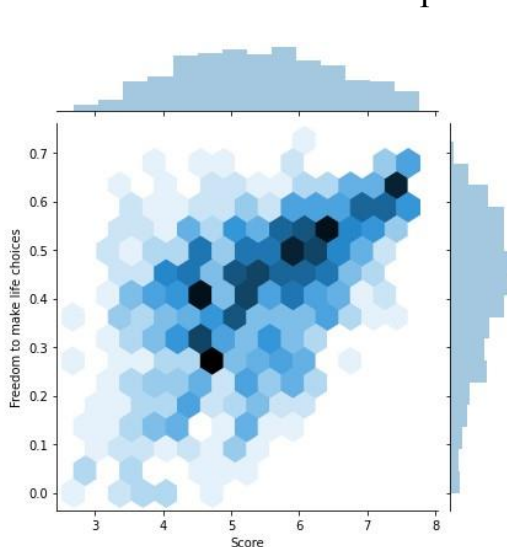


Рисунок 8 – График соотношения Score vs Freedom to make life choices

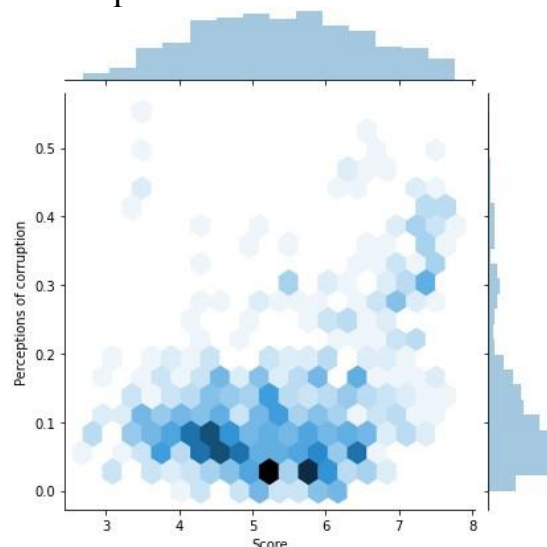


Рисунок 9 – График соотношения Score vs Perceptions of corruption

Выявление наиболее часто появляющихся значений Score

Для того чтобы исследовать частоту появления значений Score, мы построили графики, которые показывают, какое значение «Score» чаще всего встречалось. В 2015 году

```
plt.figure(figsize=(8,5))
plt.title("Уровень счастья 2015", fontsize=30)
sns.distplot(data15['Score'], bins=30, color = "g")
```

В 2016 году

```
plt.figure(figsize=(8,5))
plt.title("Уровень счастья 2016", fontsize=30)
sns.distplot(data16['Score'], bins=30, color = "b")
```

На рисунках 10 и 11 представлены графики уровней счастья с наиболее частыми значениями за 2015 и 2016 года

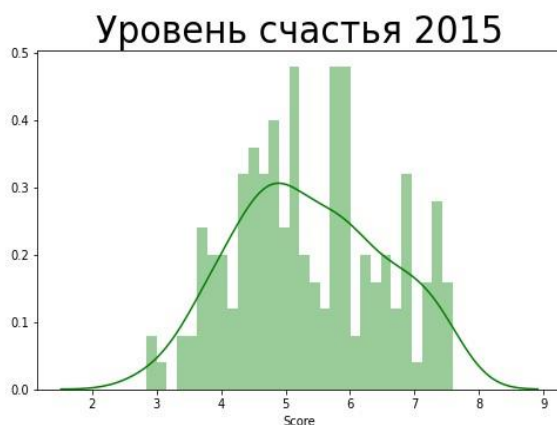


Рисунок 10 – График с наиболее частыми значениями Score за 2015 год

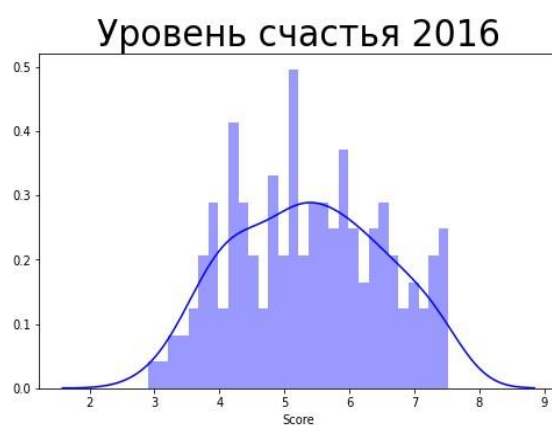


Рисунок 11 – График с наиболее частыми значениями Score за 2016 год

В 2017 году

```
plt.figure(figsize=(8,5))
plt.title("Уровень счастья 2017", fontsize=30)
sns.distplot(data17['Score'], bins=30, color = "r")
```

В 2018 году

```
plt.figure(figsize=(8,5))
plt.title("Уровень счастья 2018", fontsize=30)
sns.distplot(data18['Score'], bins=30, color = "y")
```

На рисунках 12 и 13 представлены графики уровней счастья с наиболее частыми значениями за 2017 и 2018 года

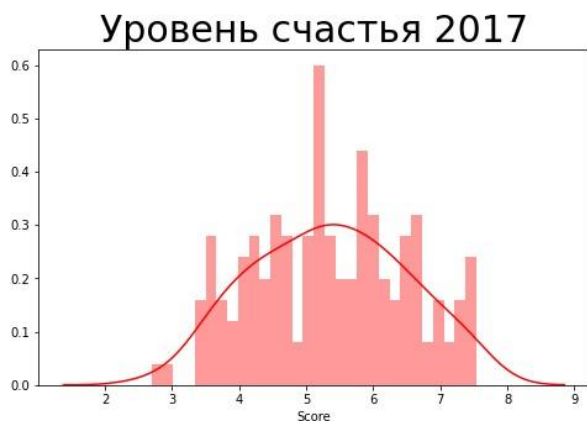


Рисунок 12 – График с наиболее частыми значениями Score за 2017 год

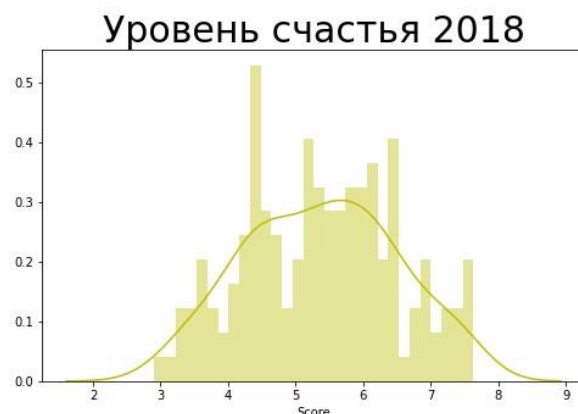


Рисунок 13 – График с наиболее частыми значениями Score за 2018 год

В 2019 году

```
plt.figure(figsize=(8,5))
plt.title("Уровень счастья 2019", fontsize=30)
sns.distplot(data19['Score'], bins=30, color = "#00fff2")
```

На рисунках 14 представлен график уровня счастья с наиболее частыми значениями за 2019 год

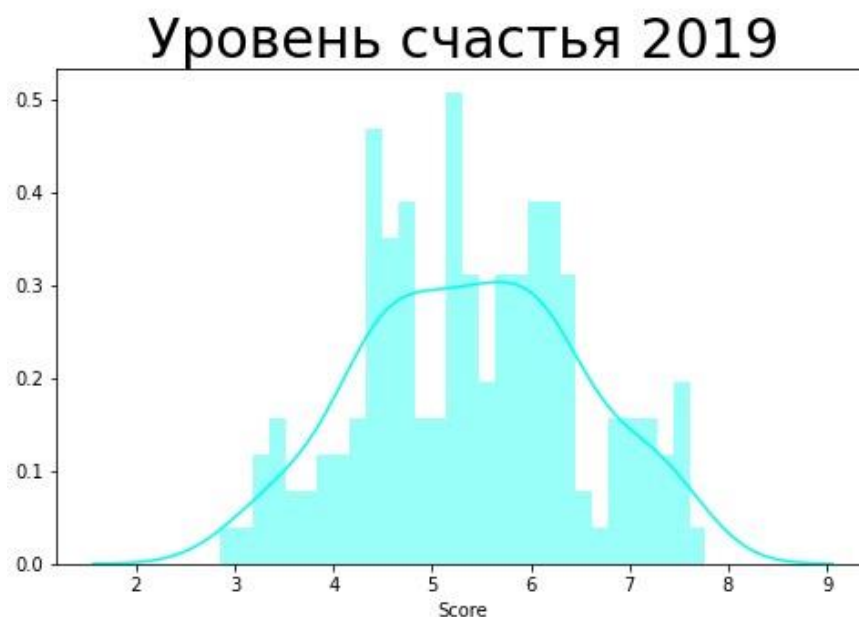


Рисунок 14 – График с наиболее частыми значениями Score за 2019 год

Диаграмма размаха.

Для того чтобы более точно проанализировать как влияет ВВП на уровень счастья, был построена диаграмма размаха. Такой вид диаграммы в удобной форме показывает медиану, нижний и верхний квартили, минимальное и максимальное значение выборки и выбросы.

```
data2 = df_all.copy()
data2['Score_1'] = pd.cut(data2['Score'], [2, 4, 5, 6, 7, 10], labels=['2-4_score', '4-5_score', '5-6_score', '6-7_score', '7+_score'])
plt.figure(figsize=(15, 8))
sns.boxplot(x="Score_1", y="GDP per capita", data=data2)
plt.show()
```

На рисунке 15 представлена диаграмма соотношения уровня Score vs GDP per capita

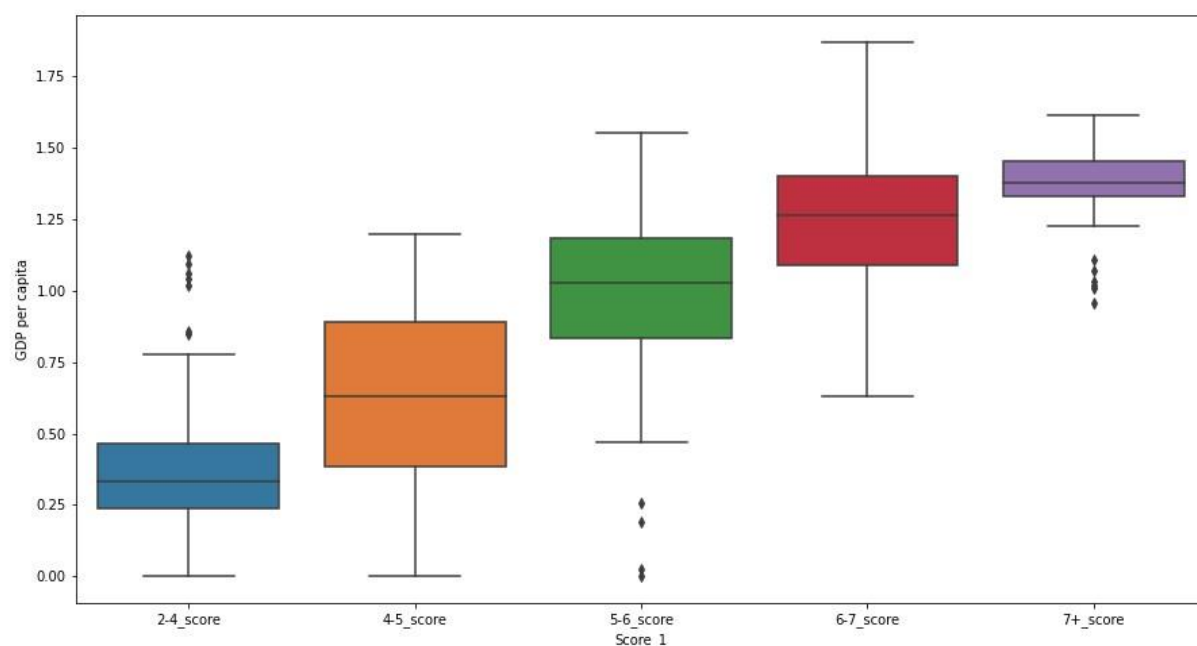


Рисунок 15 – Диаграмма соотношения уровня Score vs GDP per capita

Регрессионный анализ.

Регрессионный анализ – метод моделирования измеряемых данных и исследования их свойств. Данные состоят из пар значений зависимой переменной (переменной отклика) и независимой переменной (объясняющей переменной). Регрессионная модель есть функция независимой переменной и параметров с добавленной случайной переменной. Параметры модели настраиваются таким образом, что модель наилучшим образом приближает данные.

Для проведения регрессионного анализа были выполнены следующие действия:

Выявление величины с наибольшим влиянием на Score путем построения графиков линейной регрессии

Построение графиков Score vs GDP per capita и Score vs Social support

```
plt.figure(figsize=(8,5))
sns.regplot(x="Score", y="GDP per capita", data=df_all)
plt.figure(figsize=(8,5))
sns.regplot(x="Score", y="Social support", data=df_all)
```

На рисунках 16 и 17 представлены графики линейной регрессии Score vs GDP per capita и Score vs Social support

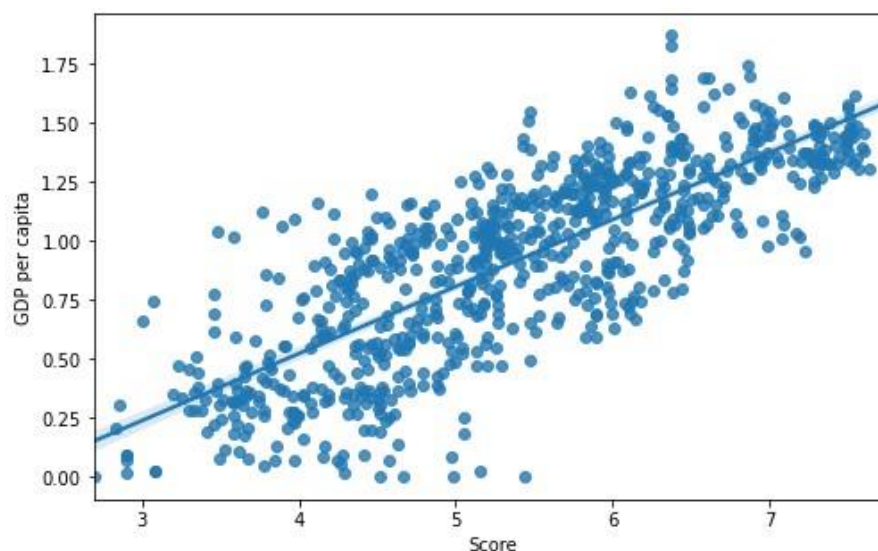


Рисунок 16 – График линейной регрессии Score vs GDP per capita

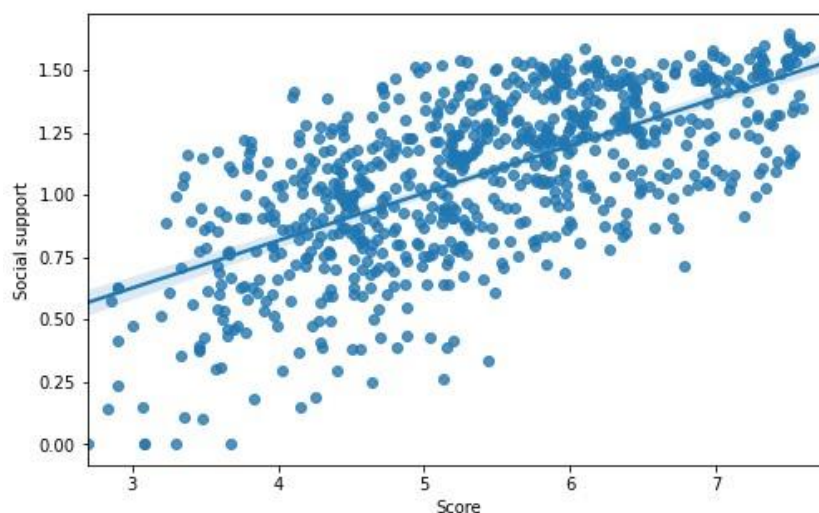


Рисунок 17 – График линейной регрессии Score vs Social support

Построение графиков Score vs Healthy life expectancy и Score vs Freedom to make life choices

```
plt.figure(figsize=(8,5))
sns.regplot(x="Score", y="Healthy life expectancy", data=df_all)
plt.figure(figsize=(8,5))
sns.regplot(x="Score", y="Freedom to make life choices", data=df_all)
```

На рисунках 18 и 19 представлены графики линейной регрессии Score vs Healthy life expectancy и Score vs Freedom to make life choices

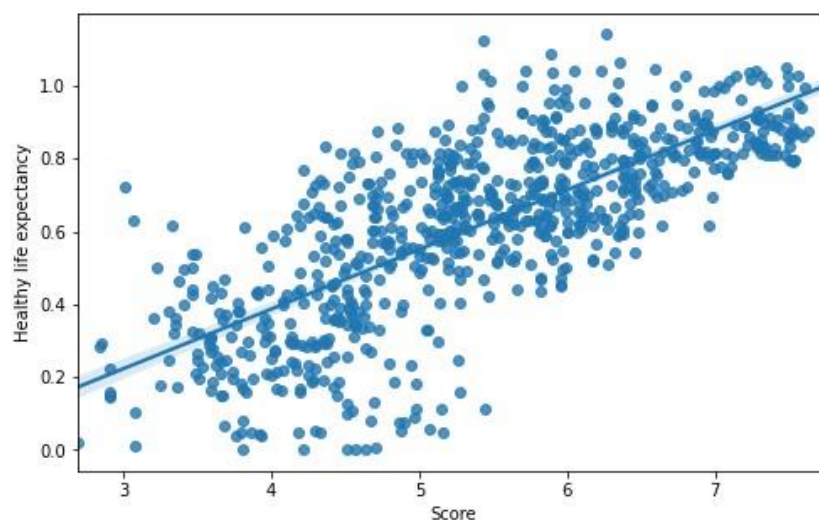


Рисунок 18 – График линейной регрессии Score vs Healthy life expectancy

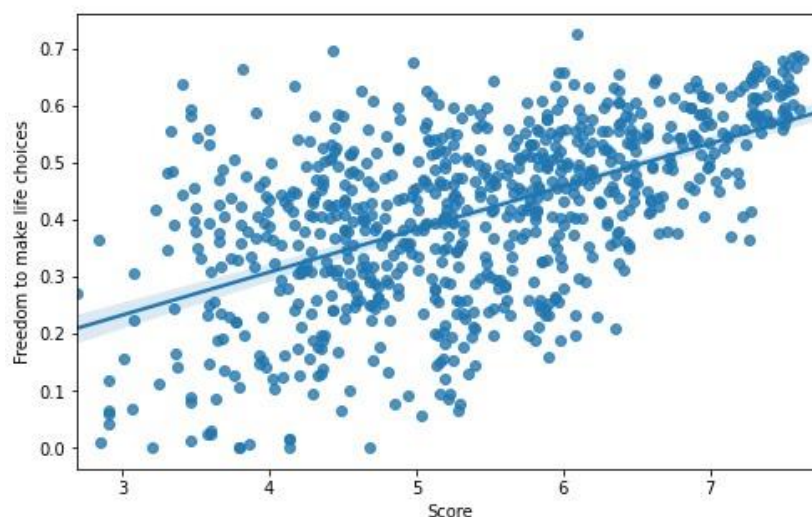


Рисунок 19 – График линейной регрессии Score vs Freedom to make life choices

Построение графиков Score vs Generosity и Score vs Perceptions of corruption

```
plt.figure(figsize=(8,5))
sns.regplot(x="Score", y="Generosity", data=df_all)
plt.figure(figsize=(8,5))
sns.regplot(x="Score", y="Perceptions of corruption", data=df_all)
```

На рисунках 20 и 21 представлены графики линейной регрессии Score vs Generosity и Score vs Perceptions of corruption

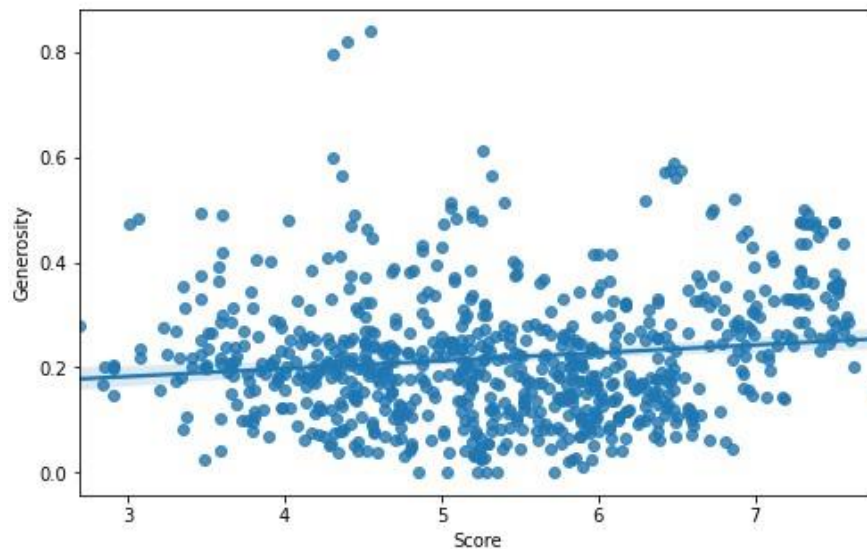


Рисунок 20 – График линейной регрессии Score vs Generosity

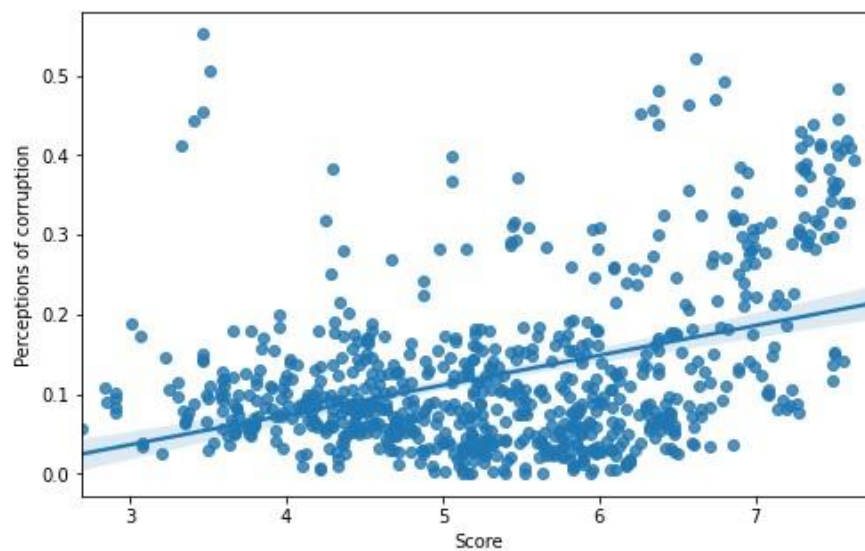


Рисунок 21 – График линейной регрессии Score vs Perceptions of corruption

Нахождение лучшей регрессионной модели, которая описывает массив данных.

Для выявления лучшей регрессионной модели необходимо рассчитать коэффициент R^2 путем следующих действий:

Получим Y

```
trg = df_all[['Score']]
```

Получим зависимые значения

```
trn = df_all.drop(['Score'], axis=1)
```

Объявим все модели для дальнейшего их обучения

```
models = [LinearRegression(), # метод наименьших квадратов Random-
ForestRegressor(n_estimators=100, max_features='sqrt'), # случайный лес
KNeighborsRegressor(n_neighbors=6), # метод ближайших соседей
SVR(kernel='linear'), # метод опорных векторов с линейным ядром]
```

Разобьем данные на тестовую и обучающую выборки

```
Xtrain, Xtest, Ytrain, Ytest = train_test_split(trn, trg,
test_size=0.4)
```

Создаем временные структуры

```
TestModels = DataFrame() tmp = {}
```

Получаем имя модели

```
for model in models:

    m = str(model)
    tmp['Model'] = m[:m.index('(')]
```

Обучаем модель

```
model.fit(Xtrain, Ytrain)
```

Вычисляем коэффициент детерминации

```
tmp['R2_Y'] = r2_score(Ytest, model.predict(Xtest))
```

Записываем данные и итоговый DataFrame

```
TestModels = TestModels.append([tmp])
```

Делаем индекс по названию модели

```
TestModels.set_index('Model', inplace=True) print(TestModels)
```

Созданный датасет с рассчитанным коэффициентом R²

Model	R2_Y
LinearRegression	0.788655
RandomForestRegressor	0.840712
KNeighborsRegressor	0.836159
SVR	0.789132

Построим графики R² коэффициентов

```
TestModels.R2_Y.plot(kind='bar', title='R2_Y')
```

Далее на рисунке 22 представлен график R² коэффициентов

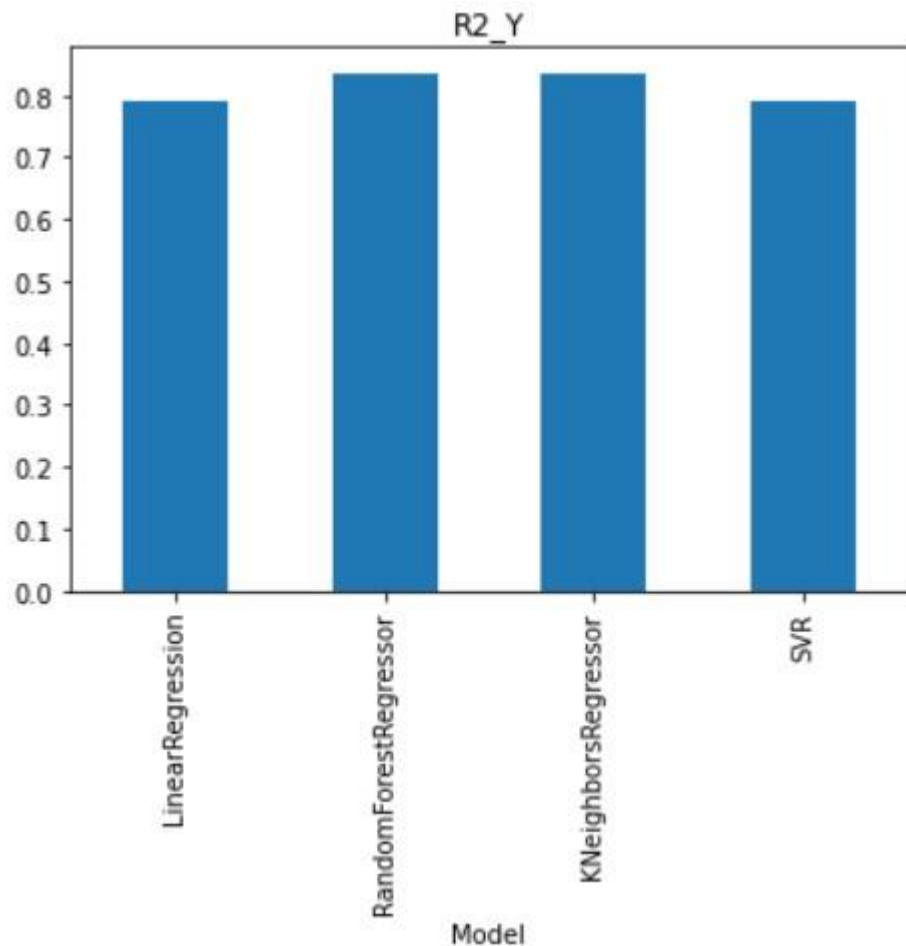


Рисунок 22 – График R² коэффициентов

Исходя из графика, можно сделать вывод, что модель, построенная методом случайных деревьев, самая точная ($R^2 = 0.840712$)

По новой обучим модель

```
model = models[1] model.fit(Xtrain, Ytrain)
```

Посмотрим, как каждый фактор влияет на прогнозное значение Y

```
model.feature_importances_
```

Вывод:

```
array([0.33196456, 0.1460898 , 0.28349085, 0.11946502, 0.04653737,
0.07245241])
```

Как мы можем увидеть ВВП на душу населения и продолжительность жизни наиболее сильно влияют на оценку счастья (около 60%), социальная поддержка и свобода жизненного выбора почти равнозначны (около 14%), а самое незначительное влияние у щедрости и коррупции (около 5% и 9% с ответственно)

Кластеризация и дерево решений.

Кластеризация (кластерный анализ) – задача группировки множества объектов на подмножества (кластеры) таким образом, чтобы объекты из одного кластера были более похожи друг на друга, чем на объекты из других кластеров по какому-либо критерию.

Кластеризация проводилась методами: k-means, Mean shift, иерархическая кластеризация.

Метод k-mean

Алгоритм разбивает множество элементов векторного пространства на заранее известное число кластеров k. Основная идея заключается в том, что на каждой итерации переисчисляется центр масс для каждого кластера, полученного на предыдущем шаге, затем векторы разбиваются на кластеры вновь в соответствии с тем, какой из новых центров оказался ближе по выбранной метрике.

```
X = df_all.values[:,1:]
X = np.nan_to_num(X)
Clus_dataSet = StandardScaler().fit_transform(X)
k_means = KMeans(init = "k-means++", n_clusters = 3, n_init = 12)
k_means.fit(X) labels = k_means.labels_

df_all["Class"] = labels df_all.groupby("Class").mean()
```

На 23 изображен результат выполнения метода k-mean

	Score	GDP per capita	Social support	Healthy life expectancy	Freedom to make life choices	Generosity	Perceptions of corruption
Class							
0	6.488727	1.329645	1.348938	0.836728	0.498882	0.238157	0.183698
1	5.304936	0.931416	1.083568	0.632216	0.388437	0.184902	0.085470
2	4.141788	0.388352	0.745482	0.312042	0.339124	0.244801	0.114086

Рисунок 23 – результат выполнения метода k-mean

Как мы можем видеть, алгоритм K-means разделил наши данные на 3 кластера. Попробуем разобраться, по каким критериям производилось деление.

К классу 0 присвоены страны, в которых наивысшие показатели ВВП на душу населения, социальная поддержка, предполагаемая продолжительность жизни, свобода жизненного выбора, но в то же время средний показатель щедрости и наивысший показатель коррумпированности

К классу 1 присвоены страны, в которых средние показатели ВВП на душу населения, социальная поддержка, предполагаемая продолжительность жизни, свобода жизненного выбора, но в то же время низший показатель щедрости и низший показатель коррумпированности

К классу 2 присвоены страны в которых низшие показатели ВВП на душу населения, социальная поддержка, предполагаемая продолжительность жизни, свобода жизненного выбора, но в то же время высший показатель щедрости и средний показатель коррумпированности.

Из чего можно сделать вывод, что для 2019 года ключевыми показателями являются:

- ВВП на душу населения
- социальная поддержка
- предполагаемая продолжительность жизни
- свобода жизненного выбора

Метод Mean shift

Алгоритм Mean shift назначает точки данным кластерам итеративно, сдвигая точки в направлении наивысшей плотности точек данных, то есть центроида кластера. В отличие от кластеризации k-means, он не делает

никаких предположений; следовательно, это непараметрический алгоритм.

```
X = df_all.values[:,1:]
X = np.nan_to_num(X)
Clus_dataSet = StandardScaler().fit_transform(X)

ms = MeanShift() ms.fit(Clus_dataSet) labels = ms.labels_
cluster_centers = ms.cluster_centers_ print(cluster_centers)
n_clusters_ = len(np.unique(labels)) print("Estimated clusters:",
n_clusters_) colors = 10*['r.', 'g.', 'b.', 'c.', 'k.', 'y.', 'm.']
for i in range(len(X)):
    plt.plot(X[i][0], X[i][1], colors[labels[i]], markersize = 5)
    plt.scatter(cluster_centers[:,0], cluster_centers[:,1]) plt.show()
```

На 24 изображен результат выполнения метода Mean shift

```
[[ 0.10726012  0.17384108  0.14108375 -0.03910207 -0.34458138 -0.34705862]]
Estimated clusters: 1
```

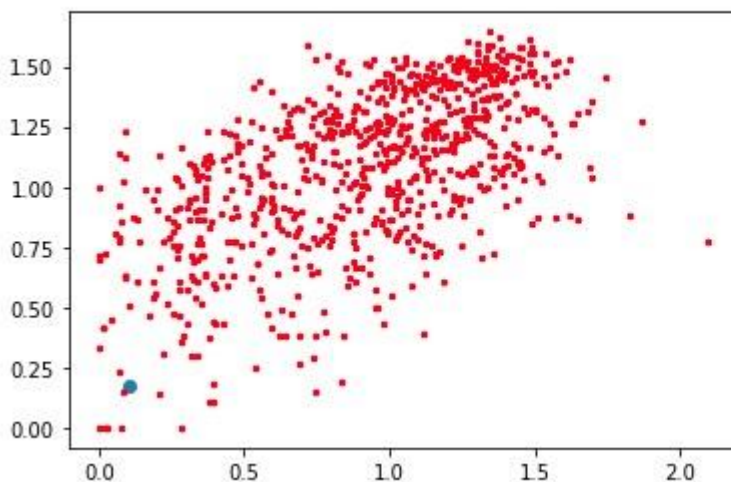


Рисунок 24 – Результат выполнения метода k-mean

Иерархическая кластеризация.

Суть иерархической кластеризации состоит в последовательном объединении меньших кластеров в большие или разделении больших кластеров на меньшие.

```
seeds_df = df_all
varieties = list(seeds_df.pop('Score'))
samples = seeds_df.values
mergings = linkage(samples, method='complete')
plt.figure(figsize=(80, 30))
dendrogram(mergings, labels=varieties,
leaf_rotation=90, leaf_font_size=6)
plt.show()
```

На рисунке 25 изображен результат выполнения иерархической кластеризации

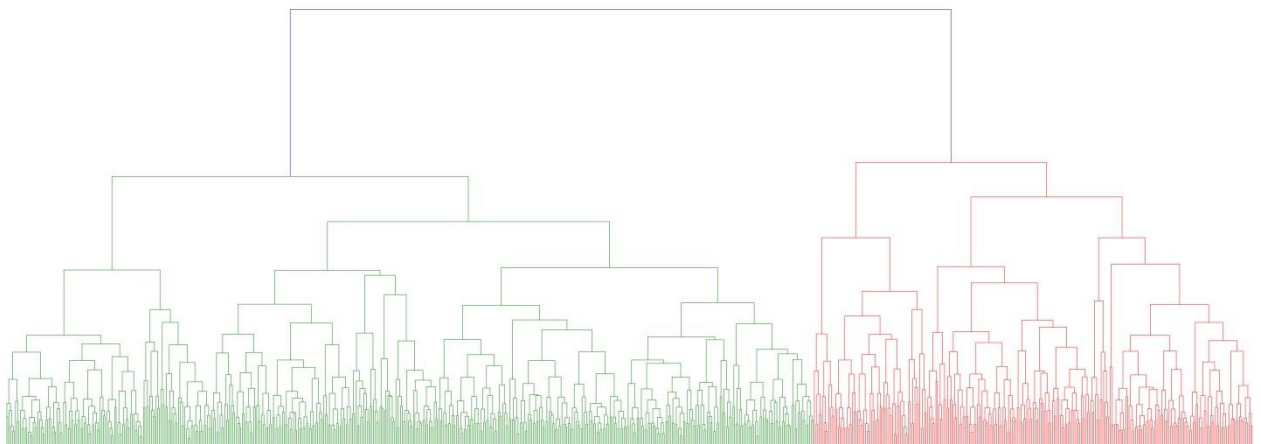


Рисунок 25 – Иерархическая кластеризация

Было построено дерево решений для классов кластеризованных методом k-means

```
df_all.head()
age_sal_tree = DecisionTreeClassifier()
age_sal_tree.fit(df_all[['Score', 'GDP per capita', 'Social support', 'Healthy life expectancy', 'Freedom to make life choices', 'Generosity', 'Perceptions of corruption']].values,
df_all['Class'].values);
dot_data = StringIO()
export_graphviz(age_sal_tree, out_file=dot_data, feature_names=['Score', 'GDP per capita', 'Social support', 'Healthy life expectancy', 'Freedom to make life choices', 'Generosity', 'Perceptions of corruption'])
(graph, ) = graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())
```

На рисунке 26 изображено дерево решений

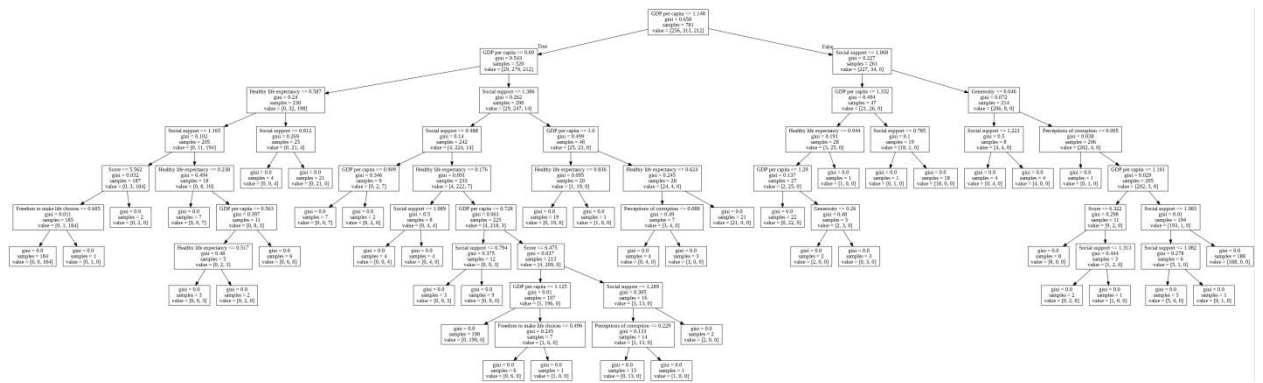


Рисунок 26 – Дерево решений

Создание и обучение RNN модели.

Рекуррентные нейронные сети (RNN) – вид нейронных сетей, где связи между элементами образуют направленную последовательность. Благодаря этому появляется возможность обрабатывать серии событий во времени или последовательные пространственные цепочки.

При реализации нейронной сети мы использовали следующие настройки:

- метод оптимизации – adam,
- метрика расчёта потерь – MAE
- обучение производится на 20 эпохах

На рисунке 27 представлен график истории обучения нейронной сети на каждой эпохе

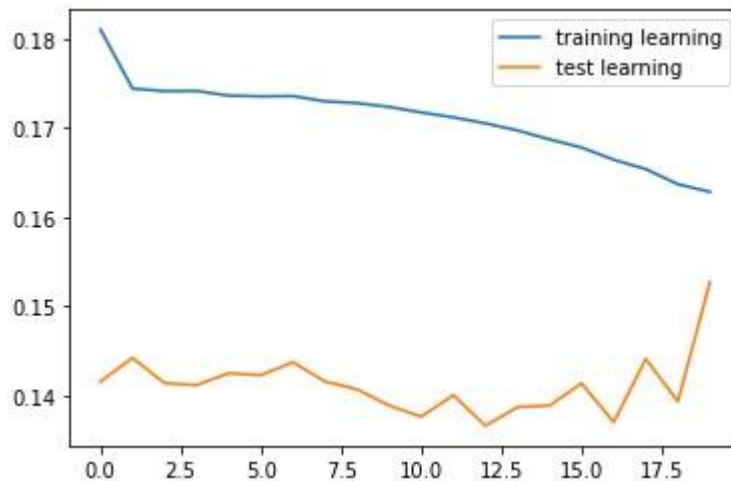


Рисунок 27 – График истории обучения

На рисунке 28 представлены графики предсказанных и реальных критерийных значений

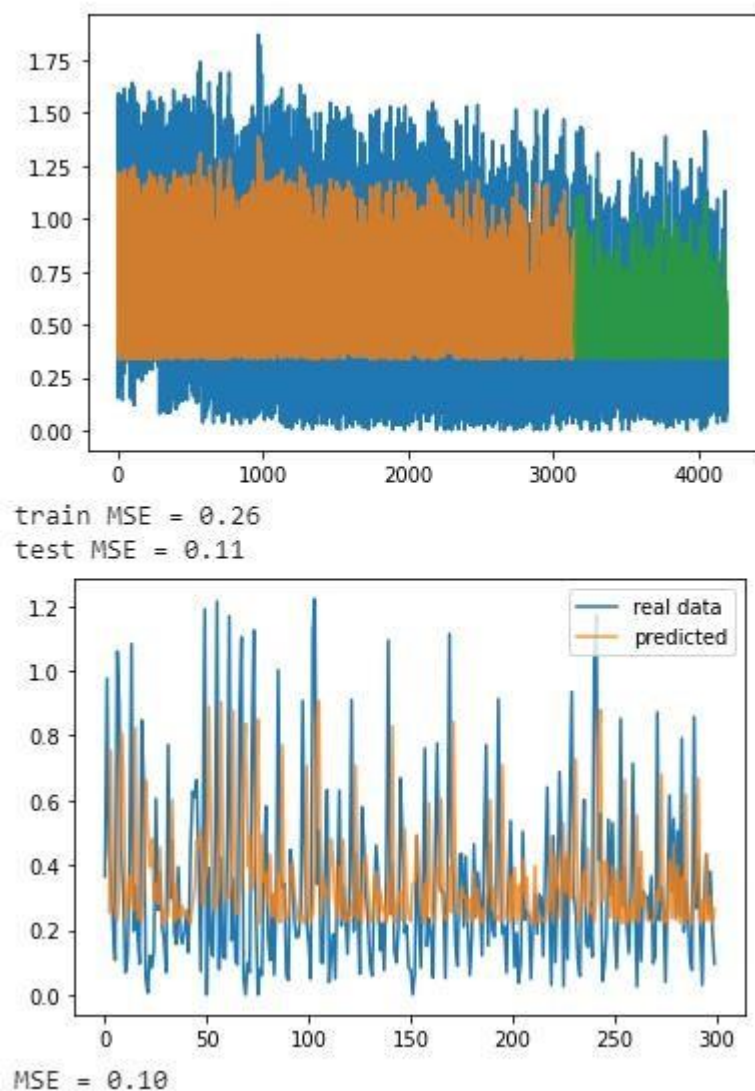


Рисунок 28 – Графики предсказанных и реальных критериальных значений

Как можно увидеть, нейросеть хоть и имеет низкий объем потерь, но ее точность не самая высокая, объясняется это тем, что наш исходный массив данных имеет небольшой объем.

Код построения нейросети:

```
def rearrange(ds, lb=2):
    X, Y = [], []
    for i in range(len(ds) - lb - 1):
        a = ds[i:(i + lb), 0]
        X.append(a)
        Y.append(ds[i + lb, 0])
    X = np.array(X)
    X = np.reshape(X, (X.shape[0], 1, 1))
    return X, np.array(Y)

def create_rnn():
    # создаём модель Sequential
    model = Sequential()
    # добавляем входной слой из LSTM
    model.add(LSTM(8, input_shape=(1, 1)))
    # и на выходе Dense слой
    model.add(Dense(1))
    # метрика расчёта потерь -- MAE (Mean Absolute Error)
```

```

        # метод оптимизации -- adam        model.compile(loss='mae', optimiz-
er='adam')        return model

    # путь до файла с моделью model_file = Path('model_BTC.h5')

    # загружаем наши данные df = df_all.copy() df = df.drop(['Score'], ax-
is=1) take_n = 700
    # делаем выборку данных для обучения и проверки ds =
np.array(df[:take_n]).reshape(-1, 1)
    # в определенном виде print(ds[:10])

    # перенормируем данные
    scl = MinMaxScaler(feature_range=(0, 1))    ds_scl =
scl.fit_transform(ds)

    # разделяем на две выборки: обучающую и для проверки d_train, d_test =
train_test_split(ds_scl, shuffle=False)    print(f'train_shape =
{d_train.shape}, test_shape = {d_test.shape}')

    # количество элементов по которым делается прогноз lb = 1
    # количество элементов для пропуска skip_count = lb * 2
    # приводим данные к виду который понимает нейронка
    X_train, Y_train = rearrange(d_train, lb) X_test, Y_test = rear-
range(d_test, lb)
    print(f'X_train_size = {X_train.shape}, Y_train_shape = {Y_train.sha-
pe}')

    # проверка на существование файла if not model_file.exists():
        # создаём model = create_rnn()
        # обучаем fit-
ted = model.fit(X_train, Y_train, epochs=20, batch_size=1, valida-
tion_data=(X_test, Y_test), verbose=2)
        # сохраняем чтобы каждый раз не обучать model.save(model_file)
        # смотрим как она обучалась на каждой эпохе
        plt.plot(fitted.history['loss'], label='training learning')
    plt.plot(fitted.history['val_loss'], label='test learning')    plt.legend()
    plt.show() else:
        # загружаем если есть обученная модель    model =
load_model(model_file)

    # посмотрим как она справляется с прогнозом на тех же данных x_train =
X_train.reshape(X_train.shape[0]) x_test = X_test.reshape(X_test.shape[0])
    y_pred_train = model.predict(X_train) y_pred_test = model.predict(X_test)

    # возвращаем данным исходные границы изменения y_pred_train =
scl.inverse_transform(y_pred_train)    y_pred_test =
scl.inverse_transform(y_pred_test)

    # реальные значения x
    x_real = np.arange(0, ds.shape[0], 1)
    x_v_train, x_v_test = train_test_split(x_real, shuffle=False)
    # и графики
    plt.plot(x_real, ds, label='real data')
    plt.plot(x_v_train[skip_count:], y_pred_train, label='train predict')

```

```

, alpha=0.8) plt.plot(x_v_test[skip_count:], y_pred_test, label='test
predict', a
lpha=0.8) plt.show()

# делаем выборку исключая первые элементы
y_real_first = ds[skip_count:y_pred_train.size + skip_count]
y_real_second = ds[y_real_first.shape[0] + 2 * skip_count:]
# считаем ошибку прогнозирования используя MSE
print('train MSE = {:.2f}'.format(mean_squared_error(y_real_first, y
_pred_train))) print('test MSE =
{:.2f}'.format(mean_squared_error(y_real_second, y _pred_test))) take_m =
50

# подготовим новые данные
ds = np.array(df[take_n:take_n + take_m]).reshape(-1, 1) ds_scl =
scl.fit_transform(ds)
X_val, _ = rearrange(ds_scl, lb)
# произведём прогноз на новом блоке данных
Y_pred = model.predict(X_val) # вернём нормальным вид данным y_pred =
scl.inverse_transform(Y_pred) x_real = np.arange(0, ds.shape[0], 1)

# выводим графики
plt.plot(x_real, ds, label='real data')
plt.plot(x_real[skip_count:], y_pred, label='predicted', alpha=0.8)
plt.legend()
plt.show()
print('MSE = {:.2f}'.format(mean_squared_error(ds[skip_count:], y_pr
ed)))

```

Заключение

Проведя анализ дата сета, команде удалось установить корреляционный зависимости, построить регрессивную модель, кластеризировать данные несколькими способами, на основе проведенной кластеризации построить деревья решений, а также создать рекуррентную нейронную сеть, которая предсказывает величину оценочных критериев.

Примерные нормы времени на выполнение работы

В процессе выполнения курсовой работы, обучающийся самостоятельно планирует свою работу в соответствии с графиком текущих аттестаций.

Примерные нормы времени на выполнение работы по разделам приведены в таблице.

Таблица 5 – Нормы времени

№ аттест.	Виды деятельности по выполнению курсовой работы	Объём в %	Количество баллов
1	Сбор информации Изучение предметной области и постановка задачи (регрессии, классификации, кластеризации). Сбор и обработка данных.	20	0–15
2	Разведочный анализ данных (качественная визуализация данных и осмысление графиков, вычисление описательных статистик, и проч.). Построение статистических моделей и алгоритмов их машинного обучения, включая разработку искусственных нейронных сетей.	50	0–60
3	Интерпретация и использование модели машинного обучения (концепция продукта данных или принятие решения на основе обобщения данных). Оформление пояснительной записки.	30	0–25
	ИТОГО		0–100

Порядок защиты работы, критерии оценки

Критерии оценки курсовой работы:

- степень усвоения студентом понятий и категорий по теме исследования;
- умение работать с документальными и литературными источниками;
- умение формулировать основные выводы по результатам анализа конкретного материала;
- грамотность и стиль изложения;
- самостоятельность работы, оригинальность в осмыслении материала;
- правильность и аккуратность оформления;
- соответствие оформления курсовой работы установленным требованиям.

Критерии, при наличии хотя бы одного из которых работа оценивается только на "неудовлетворительно". К ним относятся:

- содержание работы не относится к предмету дисциплины;
- работа перепечатана из Интернета, CD–ROM или других носителей информации;
- объем работы менее 15 листов машинописного текста;
- в работе отсутствует приложение (копии документов, логические схемы, таблицы, иллюстрации и т.д.).

Оформление курсовой работы не соответствует требованиям (отсутствует нумерация страниц, неверное или неполное оформление библиографии и т.д.).

При оценке письменных курсовых проектов преподаватель обращает также внимание на следующие распространенные ошибки в работах студентов:

- отсутствие четкости в определении основного содержания курсовой работы, убедительных доказательств, обоснований, выводов и рекомендаций;
- нарушение последовательности изложения, частые повторения, нечеткие формулировки, оговорки, грамматические ошибки.

В процессе выполнения курсовой работы обучающийся самостоятельно планирует свою работу в соответствии с графиком текущих аттестаций:

Оценка курсовой работы предусматривает использование рейтинговой системы. За каждый этап выполнения курсовой работы обучающемуся выставляются баллы (таблица 1)

Оценка **«отлично»** (91–100 баллов) выставляется обучающемуся, если он выполнил данный этап в срок, представил отчет о выполнении данного этапа, соответствующий требованиям, защитил соответствующую часть курсовой работы и при этом продемонстрировал глубокие и исчерпывающие знания всего теоретического и практического материала.

Оценка **«хорошо»** (76–90 баллов) выставляется обучающемуся, если он своевременно выполнил данный этап, представил отчет о выполнении данного этапа, соответствующий требованиям, защитил соответствующую часть курсовой работы и при этом продемонстрировал твердые и достаточно полные зна-

ния всего теоретического и практического материала.

Оценка **«удовлетворительно»** (61–75 баллов) выставляется обучающемуся, если он либо несвоевременно выполнил данный этап, либо представил отчет о выполнении данного этапа, не соответствующий требованиям, либо защитил соответствующую часть курсовой работы и при этом продемонстрировал не всегда полные и конкретные, но без грубых ошибок ответы на поставленные вопросы.

Оценка **«неудовлетворительно»** (менее 61 балла) выставляется обучающемуся, если он либо не выполнил данный этап, либо не представил отчет о выполнении данного этапа, либо при защите соответствующей части курсовой работы допустил грубые ошибки в ответе, продемонстрировал непонимание сущности излагаемых вопросов.

Список рекомендуемой литературы

1. Сапрыкин, О. Н. Интеллектуальный анализ данных : учебное пособие / О. Н. Сапрыкин. – Самара : Самарский университет, 2020. – 80 с. – ISBN 978–5–7883–1563–8. – Текст : электронный // Лань : электронно–библиотечная система. – URL: <https://e.lanbook.com/book/188906>. – Режим доступа: для авториз. пользователей.
2. Макшанов, А. В. Технологии интеллектуального анализа данных : учебное пособие / А. В. Макшанов, А. Е. Журавлев. – 2–е изд., стер. – Санкт–Петербург : Лань, 2022. – 212 с. – ISBN 978–5–8114–4493–9. – Текст : электронный // Лань : электронно–библиотечная система. – URL: <https://e.lanbook.com/book/206711>. – Режим доступа: для авториз. пользователей.
3. Бонцанини, М. Анализ социальных медиа на Python. Извлекайте и анализируйте данные из всех уголков социальной паутины на Python / М. Бонцанини ; перевод с английского А. В. Логунова. – Москва : ДМК Пресс, 2018. – 288 с. – ISBN 978–5–97060–574–5. – Текст : электронный // Лань : электронно–библиотечная система. – URL: <https://e.lanbook.com/book/108129>. – Режим доступа: для авториз. пользователей.
4. Сьоре, Э. Проектирование и реализация систем управления базами данных / Э. Сьоре ; перевод с английского А. Н. Киселева. – Москва : ДМК Пресс, 2021. – 466 с. – ISBN 978–5–97060–488–5. – Текст : электронный // Лань : электронно–библиотечная система. – URL: <https://e.lanbook.com/book/190718>. – Режим доступа: для авториз. пользователей.
5. Целых, А. Н. Современные методы прикладной информатики в задачах анализа данных : учебное пособие / А. Н. Целых, А. А. Целых, Э. М. Котов. – Ростов–на–Дону : ЮФУ, 2021. – 130 с. – ISBN 978–5–9275–3783–9. – Текст : электронный // Лань : электронно–библиотечная система. – URL: <https://e.lanbook.com/book/195357>. – Режим доступа: для авториз. пользователей.
6. Мартин, О. Байесовский анализ на Python : руководство / О. Мартин ; перевод с английского А. В. Снастина. – Москва : ДМК Пресс, 2020. – 340 с. – ISBN 978–5–97060–768–8. – Текст : электронный // Лань : электронно–библиотечная система. – URL: <https://e.lanbook.com/book/140585>. – Режим доступа: для авториз. пользователей.

Требования к оформлению работы

Курсовая работа оформляется в виде пояснительной записки, которая должна включать:

- титульный лист (приложение А)
- содержание;
- введение;
- основная часть;
- заключение;
- список литературы;
- приложения.

Титульный лист – первая страница работы – служит источником для информации об авторе, руководителе, теме и т.д. Пример оформления титульного листа представлен в приложении А.

Титульный лист включают в общую нумерацию страниц отчета. Номер страницы на титульном листе не проставляют.

Структурный элемент пояснительной записки «СОДЕРЖАНИЕ» размещается после титульного листа, начиная со следующей страницы.

«СОДЕРЖАНИЕ» включает:

- введение;
- наименование разделов (глав), подразделов (параграфов), пунктов (если они имеют наименование) основной части,
- заключение,
- список использованных источников,
- наименование приложений с указанием номеров страниц.

Слово «СОДЕРЖАНИЕ» записывают в виде заголовка в середине строки симметрично относительно текста прописными буквами. Наименования, включенные в содержание, записывают строчными буквами, начиная с прописной.

Пример оформления содержания приведен в приложении Б.

Структурный элемент отчета «ВВЕДЕНИЕ» отражает актуальность темы, объект и предмет исследования, цель и задачи исследования, методы исследования, методологические основы исследования.

К написанию введения целесообразно приступить после изложения основной части работы.

Рекомендуемый удельный вес введения – до 5% в общем объеме работы.

Основная часть работы должна содержать данные, отражающие существо, методику и основные результаты выполненной работы.

Основную часть работы следует делить на главы (разделы). Главы могут делиться на параграфы. Параграфы, при необходимости, могут делиться на пункты. Каждый параграф должен содержать законченную информацию.

Основная часть содержит следующие разделы:

- словесное описание задачи;
- построение математической модели;

- выбор, обоснование и описание метода решения рассматриваемой задачи;
- описание программы;
- решение поставленной задачи;
- анализ модели на чувствительность;

Структурный элемент «ЗАКЛЮЧЕНИЕ» является выводом (следствием) из содержания работы. В нем освещаются основные теоретические положения, обобщаются результаты разработки, даются наиболее важные выводы. Заключение должно быть связано с основной частью и вытекать из нее.

В структурном элементе «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» должна быть указана вся использованная литература, в том числе издания на иностранном языке (при необходимости) и электронные ресурсы, а по тексту пояснительной записки – ссылки на нее. Все перечисленные в списке работы нумеруются. Допускается два варианта расположения источников: в порядке появления ссылок в тексте и по алфавиту. Пример оформления использованных источников находится в приложении В.

Структурный элемент «ПРИЛОЖЕНИЯ» включают в структуру отчета при необходимости.

В приложения следует включать вспомогательный материал, необходимый для полноты изложения результатов работы, например:

- 1) промежуточные математические расчеты;
- 2) таблицы вспомогательных данных;
- 3) иллюстрации вспомогательного характера;
- 4) технологические инструкции;
- 5) результаты тестирования и т.д.
- 6) листинги программ;
- 7) формы входных и выходных документов;
- 8) иллюстрации, таблицы, распечатки с компьютера.

Минимальный объем курсовой работы должен составлять 20–25 страниц машинописного текста.

Записка должна быть выполнена с использованием компьютера и принтера на одной стороне листа белой бумаги формата А4. Цвет шрифта должен быть черным, высота букв, цифр и других знаков – не менее 1,8 мм (14 кегль). Междустрочный интервал – полуторный (для таблиц допускается одинарный), тип шрифта – Times New Roman, абзацный отступ – 1,25 см, выравнивание по ширине текста.

Текст записки следует печатать, соблюдая следующие размеры полей: правое – 10 мм, верхнее – 20 мм, левое – 25 мм, нижнее – 20 мм.

Опечатки, описки и графические неточности, обнаруженные в процессе подготовки отчета, допускается исправлять подчисткой или закрашиванием белой краской и нанесением на том же месте исправленного текста (графики) машинописным способом или черными чернилами, пастой или тушью – рукописным способом.

Повреждения листов текстовых документов, помарки и следы не полностью удаленного прежнего текста (графики) не допускаются.

Наименования структурных элементов записки «СОДЕРЖАНИЕ», «ВВЕДЕНИЕ», «ЗАКЛЮЧЕНИЕ», «СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ» служат заголовками структурных элементов записки.

Заголовки структурных элементов пояснительной записки КП пишутся в середине строки симметрично относительно текста прописными буквами без точки, не подчеркиваются.

Каждый структурный элемент следует начинать с нового листа (страницы), в том числе разделы (главы) основной части и приложения.

Основную часть записки следует делить на разделы, подразделы и пункты.

Разделы, подразделы, пункты следует нумеровать арабскими цифрами и записывать с абзацного отступа.

Разделы должны иметь порядковую нумерацию в пределах всего текста, за исключением приложений.

Пример – 1, 2, 3 и т.д.

Номер подраздела или пункта включает номер раздела и порядковый номер подраздела или пункта, разделенные точкой.

Пример – 1.1, 1.2, 1.3 и т.д.

Номер пункта включает номер раздела и подраздела, разделенные точкой.

Пример – 1.1.1, 1.1.2, 1.1.3 и т.д.

После номера раздела, подраздела и пункта в тексте точку не ставят. Если текст отчета подразделяют только на пункты, их следует нумеровать, за исключением приложений, порядковыми номерами в пределах всего отчета. Если раздел или подраздел имеет только один пункт, то нумеровать его не следует. Если документ не имеет подразделов, то нумерация пунктов в нем должна быть в пределах каждого раздела, и номер пункта должен состоять из номеров раздела и пункта, разделенных точкой. В конце номера пункта точка не ставится.

Разделы, подразделы должны иметь заголовки. Заголовки должны четко и кратко отражать содержание разделов, подразделов. Заголовки разделов, подразделов и пунктов следует печатать с абзацного отступа с прописной буквы без точки в конце, не подчеркивая. Если заголовок состоит из двух предложений, их разделяют точкой.

Если текст записки подразделяется только на пункты, то они нумеруются порядковыми номерами в пределах всей записки. Внутри пунктов могут быть приведены перечисления. Перед каждым перечислением следует ставить дефис или, при необходимости, ссылки в тексте документа на одно из перечислений, строчную букву (за исключением ё, з, о, г, ь, й, ы, ъ), после которой ставится скобка. Для дальнейшей детализации перечислений необходимо использовать арабские цифры, после которых ставится скобка, а запись производится с абзацного отступа, как показано в примере.

Пример

- а) _____
б) _____
 1) _____
 2) _____
в) _____

Работа должна быть написана грамотным литературным языком. При этом следует соблюдать следующие требования:

- излагать мысли четко, кратко и конкретно;
- каждая новая мысль должна начинаться с абзаца;
- каждый абзац начинается с красной строки;
- каждая глава – с новой страницы, а параграф – с абзаца;
- каждая глава (параграф, пункт) должна иметь заголовок, который нумеруется арабскими цифрами.

Страницы пояснительной записки и приложений, входящих в состав записки, следует нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всей работе. Номер страницы проставляют в центре нижней части листа без точки. Нумерация страниц проставляется со структурного элемента «СОДЕРЖАНИЕ».

Иллюстрации (чертежи, графики, схемы, компьютерные распечатки) следует располагать в записке непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. Иллюстрации могут быть в компьютерном исполнении, в том числе и цветные. На все иллюстрации должны быть даны ссылки в записке.

Если рисунок один, то он обозначается «Рисунок 1». Слово «Рисунок» и его наименование располагают посередине строки.

Иллюстрации нумеруют в пределах каждого раздела. В этом случае номер иллюстрации состоит из номера раздела и порядкового номера иллюстрации, разделенных точкой. Например, «Рисунок 1.1».

Иллюстрации, при необходимости, могут иметь наименование и пояснительные данные (подрисуночный текст). Слово «Рисунок» и наименование помещают после пояснительных данных и располагают следующим образом: Рисунок 1 – Детали прибора. Точка в конце наименования рисунка не ставится. Далее следует подрисуночный текст. Допускается применять размер шрифта подрисуночной надписи меньший, чем в тексте.

При ссылках на иллюстрации следует писать «... в соответствии с рисунком 1.2».

Все таблицы, если их несколько, нумеруют арабскими цифрами в пределах всего текста. Над правым верхним углом таблицы помещают надпись "Таблица..." с указанием порядкового номера таблицы (например, "Таблица 4") без значка № перед цифрой и точки после нее. Если в тексте работа только одна таблица, то номер ей не присваивается и слово "таблица" не пишут.

Все таблицы, если их несколько, нумеруют арабскими цифрами в пределах всего текста. Над правым верхним углом таблицы помещают надпись "Таб-

лица..." с указанием порядкового номера таблицы (например, "Таблица 4") без значка № перед цифрой и точки после нее. Если в тексте работа только одна таблица, то номер ей не присваивается и слово "таблица" не пишут. Таблицы снабжают тематическими заголовками, которые располагают посередине страницы и пишут с прописной буквы без точки на конце.

При переносе части таблицы название помещают только над первой частью таблицы, нижнюю горизонтальную черту, ограничивающую таблицу, не проводят. Таблицу следует располагать в записке непосредственно после текста, в котором она упоминается впервые, или на следующей странице. На все таблицы должны быть ссылки в записке. При ссылке следует писать слово «Таблица» с указанием ее номера.

Таблицу с большим количеством строк допускается переносить на другой лист. При переносе части таблицы на другой лист слово «Таблица» и ее номер указывается один раз над первой частью таблицы, над другими частями пишут слово «Продолжение» и указывают номер таблицы, например: «Продолжение таблицы 7.1». При переносе таблицы на другой лист заголовок помещают только над её первой частью.

Таблицу с большим количеством граф допускается делить на части и помещать одну часть под другой в пределах одной страницы. Если строки и графы выходят за формат страницы, то в первом случае в каждой части таблицы повторяется головка, во втором случае – боковик.

Таблицы нумеруют в пределах каждого раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы, разделённых точкой.

Уравнения и формулы следует выделять из текста в отдельную строку, если они являются длинными и громоздкими. Выше и ниже каждой формулы или уравнения должно быть оставлено не менее одной свободной строки. Если формула не уместится в одну строку, то она должна быть перенесена после знака равенства (=) или после знаков плюс (+), минус (–), умножения (х), деления (:), или других математических знаков, причем знак в начале следующей строки повторяют. При переносе формулы на знаке, символизирующем операцию умножения, применяют знак “х”.

Формулы нумеруют в пределах каждого раздела арабскими цифрами. Номер формулы состоит из номера раздела и порядкового номера формулы, разделённых точкой. Номер формулы указывают в круглых скобках в крайнем правом положении на строке.

Пример

$$A=b / c, \quad (5.1)$$

где b –, кг;

c –, м.

Приложение оформляют как продолжение текста пояснительной записки КП на последующих его листах или выпускают в виде самостоятельного документа. В тексте документа на все приложения должны быть даны ссылки. При-

ложения располагают в порядке ссылок на них в тексте документа.

Каждое приложение следует начинать с новой страницы с указанием наверху посередине страницы слова “Приложение”, его обозначения и степени. Если текст одного приложения расположен на нескольких страницах, то в правом верхнем углу страницы пишут «Продолжение приложения» и указывают его обозначение и степень.

Приложение, как правило, должно иметь заголовок, который записывают симметрично относительно текста с прописной буквы отдельной строкой.

Пример оформления титульного листа пояснительной записки
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное
учреждение высшего образования
«ТЮМЕНСКИЙ ИНДУСТРИАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра кибернетических систем

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к курсовой работе
по дисциплине «Технологии интеллектуального анализа BigData»
на тему «.....»
Вариант № ____

РУКОВОДИТЕЛЬ:

должность, ученая степень

_____ И. О. Фамилия

РАЗРАБОТЧИК:

студент группы _____

_____ И.О. Фамилия

202__

**Пример оформления содержания курсовой работы
по дисциплине «Технологии интеллектуального анализа BigData»**

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1. Обзор предметной области	4
2. Проектирование и теоретическое обоснование системы.....	10
2.1. Постановка задачи	10
2.2. Алгоритм анализа данных в приложении.....	12
2.3. Показатели качества нейронной сети.....	15
2.4. Сравнение и оценка результатов.....	20
3. Реализация системы	24
ЗАКЛЮЧЕНИЕ	28
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	30
ПРИЛОЖЕНИЯ	31

Пример оформления списка использованных источников
Список использованных источников

Однотомное издание (книга) одного автора

1 Иванов, А. А. Автоматизация технологических процессов и производств [Текст]: учебное пособие / А. А. Иванов. – Москва: ФОРУМ, 2012. – 223 с.

Однотомное издание (книга) двух авторов

2 Андреев, Е. Б. Технические средства систем управления технологическими процессами в нефтяной и газовой промышленности [Текст]: учебное пособие / Е. Б. Андреев, В. Е. Попадько. – М.: Нефть и газ РГУ нефти и газа им. И.М. Губкина, 2004. – 269 с.

Однотомное издание трех авторов

3 Тюрин, О. Г. Управление потенциально опасными технологиями [Текст] / О. Г. Тюрин, В. С. Кальницкий, Е. Ф. Жеглов; ПластикЭнтерпрайз, Федеральный центр двойных технологий Союз. – М.: Инфра–Инженерия, 2011. – 280 с.

Однотомное издание четырех и более авторов

4 Автоматизация технологических процессов добычи и подготовки нефти и газа [Текст]: учебное пособие / Е. Б. Андреев [и др.]. – М.: Недра, 2008. – 399 с.

Однотомное издание под редакцией

5 Теплотехника [Текст]: учебник для вузов / под общ. ред. А. П. Баскакова. – М.: Энергоиздат, 1991. – 224 с.

Переводное издание

6 Дорф, Р. Современные системы управления [Текст] = Modern control systems / Р. Дорф, Р. Бишоп; пер. с англ. Б. И. Копылова. – М.: Лаборатория Базовых Знаний, 2004. – 832 с.

Том многотомного издания

7 Методы классической и современной теории автоматического управления [Текст]: учебник для студентов вузов: в 5 т. Т. 2:

Статья из журнала, газеты

8 Субарев, Д.Н. Факторная модель динамики освоения ресурса ЭЦН / И.Г. Соловьев, А.Г. Кожин, Д.Н. Субарев // НТЖ “Автоматизация, телемеханизация и связь в нефтяной промышленности”. – 2013. – №8. – С.25–29.

Законодательные материалы: законы, указы, постановления

9 Конституция Российской Федерации [Текст]: офиц. текст. – М.: Маркетинг, 2001. – 39 с.

10 Об охране окружающей среды [Текст]: федер. закон от 10 янв. 2002 г. № 7–ФЗ // Собрание законодательства РФ. – 2002. – № 2. – Ст. 133.

11 О Федеральных природных ресурсах [Текст]: указ Президента РФ

от 16 дек. 1993 г. № 2144 // Собр. актов Президента и Правительства РФ. – 1993. – № 51. – Ст. 4932.

Электронные ресурсы

12 ПЛК компании VIPA с поддержкой сети EtherCAT открывают новые возможности [Электронный ресурс]. URL: <http://avtprom.ru/news/2013/11/22/plk-kompanii-vipa-s-podd> (дата обращения: 27.11.2013).