Министерство науки и высшего образования Российской Федерации

федеральное государственное бюджетное образовательное учреждение высшего образования

«Российский экономический университет имени Г. В. Плеханова»

Кафедра информатики

Выпускная квалификационная работа

по программе профессиональной переподготовке «Анализ данных и машинное обучение в среде Python»

на тему «Определение риска развития диабета на ранних стадиях с использованием методов машинного обучения»

Выполнил:

ФИО

Сенюта Александр Александрович

Преподаватель:

ст. преп. Савинова Виктория Михайловна

Москва

2025

1. Понимание бизнес-целей

1.1. Понимание бизнеса

Повысить эффективность специалистов медиков в области диагностики развития диабета на ранних стадиях. В связи с этим необходимо:

1. Повысить точность верных диагнозов специалистов и снизить количество ошибок.

2. Повысить уровень доверия пациентов к специалистам-медикам, что повысит лояльность клиентов к клинике и повысит уровень репутации.

1.2. Доступные ресурсы

Для успешной реализации проекта необходимы следующие категории специалистов: аналитик данных, бизнес-аналитик, специалист по базам данных, руководитель проекта.
Заказчик располагает всем необходимым оборудованием для поведения анализа данных.

1.3. Риски

1. Несоблюдение сроков проекта
2. Риск неплатежеспособности заказчика
3. Риск нехватки и неполноты данных
4. Риск несоответствия полученных результатов требованиям заказчика.

1.4. Ограничения

Ограничение сроков: 6 месяцев. Ставки по сотрудникам:
Аналитик данных – 1 ставка.
Бизнес-аналитик – 1 ставка.

1.5. Цели исследования данных

1. Проведение разведочного анализа данных, включая визуализацию влияния различных факторов на возникновение диабета, построение таблиц описания данных.
2. Построение модели классификации для применения в диагностике исследуемого заболевания. В качестве моделей предполагается использование модели логистической регрессии, случайного соседа, деревьев решений, случайного леса.

1.6. Критерии успешности изучения данных.

Метрики оценки точности и качества построенных моделей:
Метрики оценки качества классификации: accuracy, recall, precision, f1.
Диапазон целевых значений метрик моделей: accuracy, recall, precision, f1 >= 0.8.

2. Начальное изучение данных.

2.1. Сбор данных.

Внутренние данные – 'age', 'gender', 'polyuria', 'polydipsia', 'sudden_weight_loss', 'weakness', 'polyphagia', 'genital thrush', 'visual blurring', 'itching', 'irritability', 'delayed healing', 'partial paresis', 'muscle stiffness', 'alopecia', 'obesity', 'class' (https://www.kaggle.com/datasets/andrewmvd/early-diabetes-classification/data)
Внешние данные – Не требуется
Дополнительные данные – Не требуются

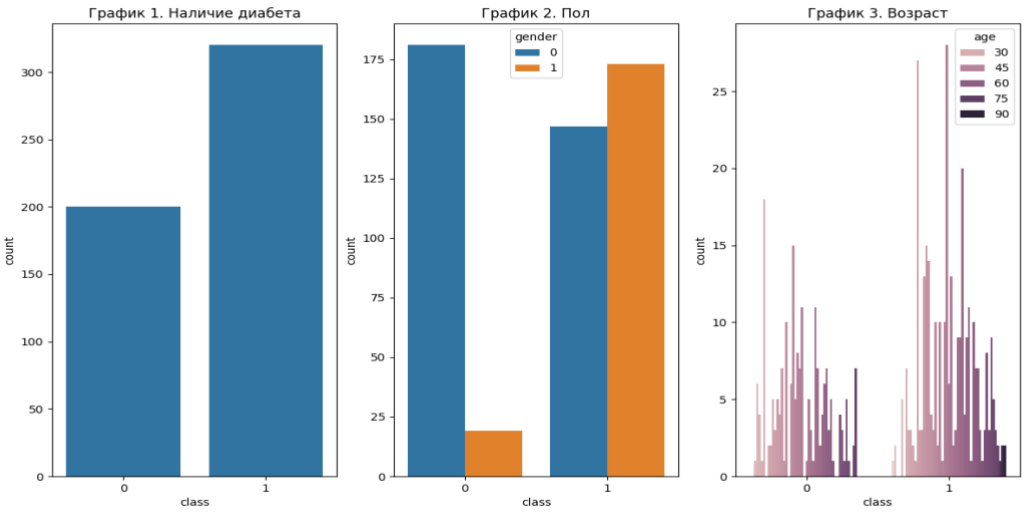
2.2. Описание данных

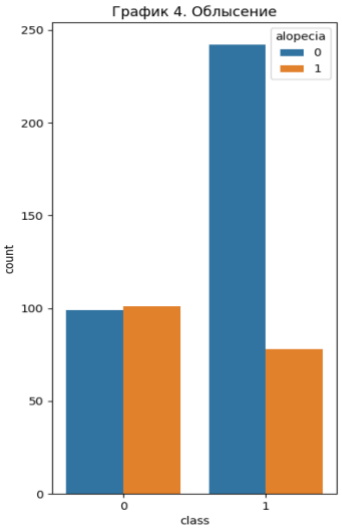
Объем данных – 69.2 KB
Типы, виды данных и схемы кодирования

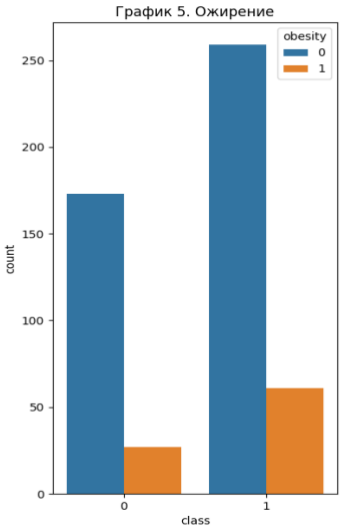
Наименование	Тип данных	Вид данных	Схема кодирования
age	int	Дискретный	-
gender	int	Дискретный	‘Male’ = 0, ‘Female’ = 1
polyuria	int	Дискретный	-
polydipsia	int	Дискретный	-
sudden_weight_loss	int	Дискретный	-
weakness	int	Дискретный	-
polyphagia	int	Дискретный	-
genital_thrush	int	Дискретный	-
visual_blurring	int	Дискретный	-
itching	int	Дискретный	-
irritability	int	Дискретный	-
delayed_healing	int	Дискретный	-
partial_paresis	int	Дискретный	-
muscle_stiffness	int	Дискретный	-
alopecia	int	Дискретный	-
obesity	int	Дискретный	-
class	int	Дискретный	-

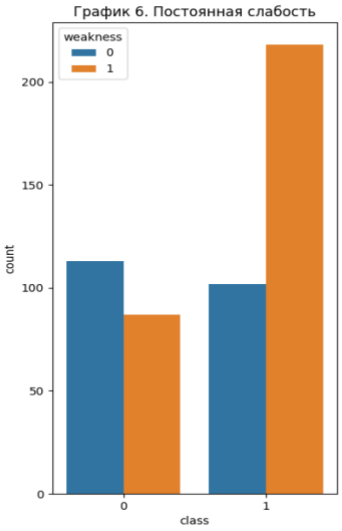
Формат данных – файл csv, разделитель – “;”.

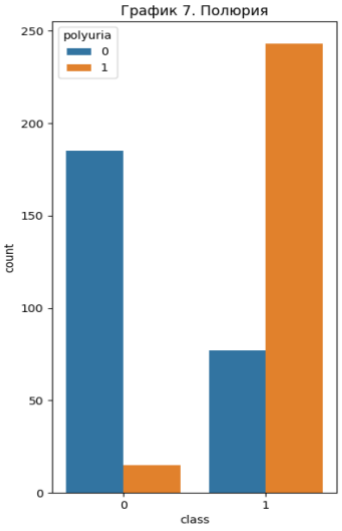
2.3. Исследование данных

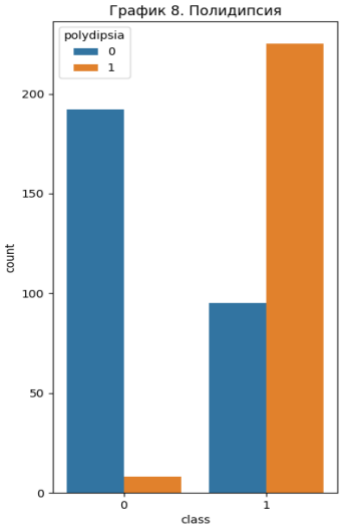


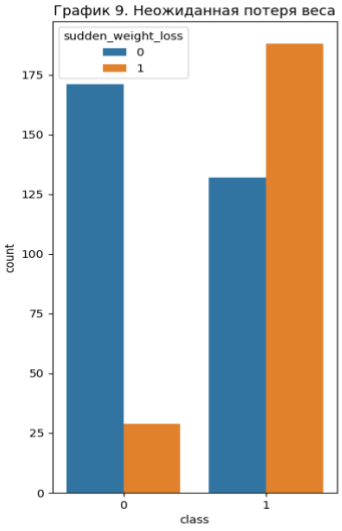


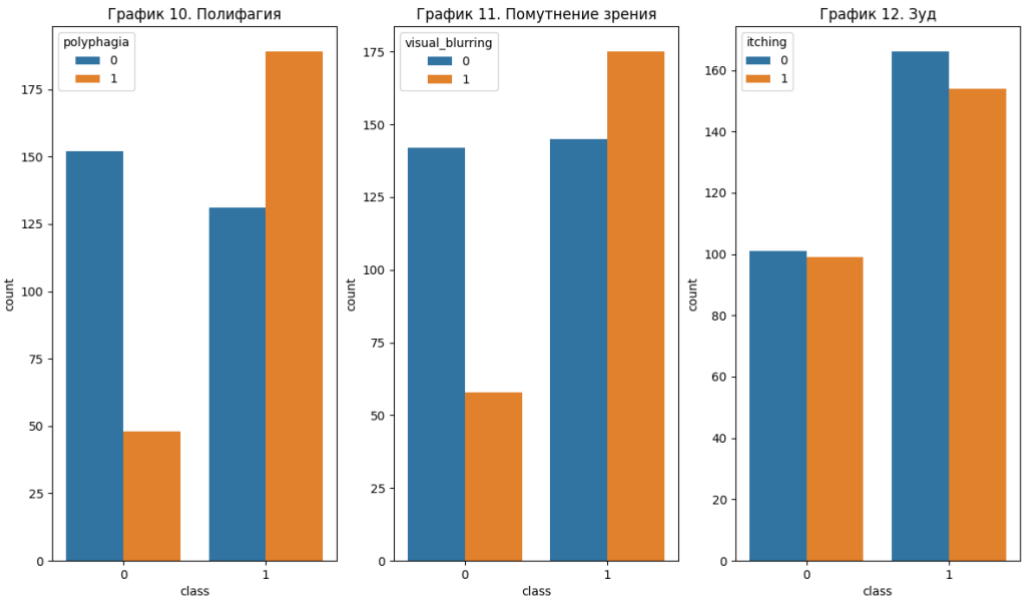


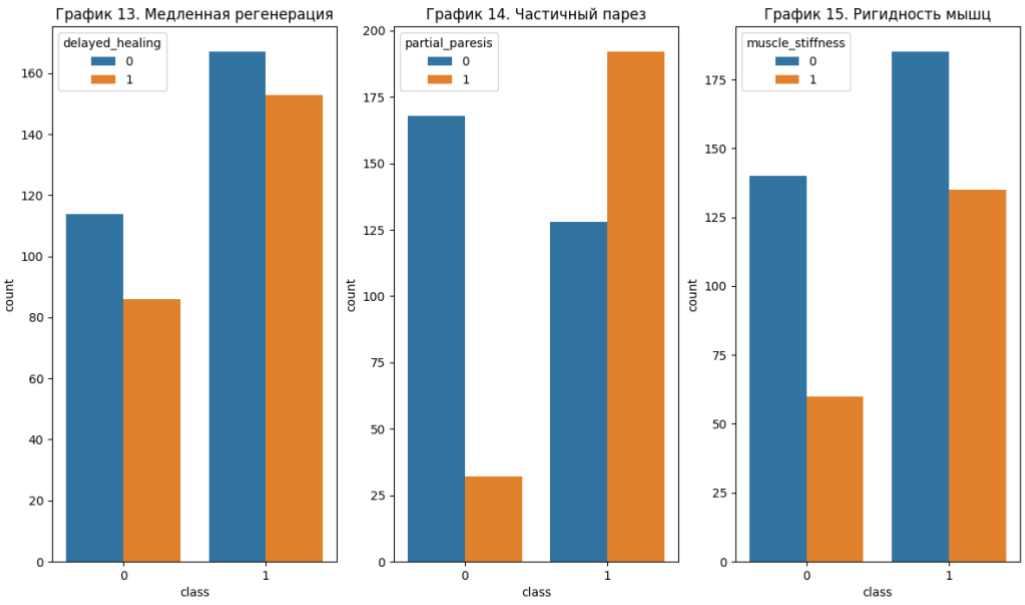












На графике 1 представлено разбиение по целевой переменной, а именно наличие диабета на ранних стадиях у пациента. Из графика видно, что количество пациентов с обнаруженным диабетом на ранних стадиях преобладает в ходе исследования.

На графике 2 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от пола пациента. Из графика видно, что количество женщин с обнаруженным диабетом на ранних сильно преобладает над количеством женщин без диабета.

На графике 3 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от возраста пациента. Из графика видно, что диабет на ранних стадиях чаще наблюдется у пациентов с молодым возрастом.

На графике 4 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, страдал ли пациент от облысения или нет. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые не сталкивались с сильным облысением.

На графике 5 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, страдал ли пациент от ожирения или нет. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые не сталкивались с ожирением.

На графике 6 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, испытывал ли пациент постоянную слабость или нет. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые сталкивались с постоянной слабостью.

На графике 7 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, испытывал ли пациент обильное мочеиспускание или нет. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые болели полиурией.

На графике 8 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, испытывал ли пациент чрезмерную жажду/избыточное питье или нет. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые болели полидипсией.

На графике 9 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, был ли у пациента эпизод внезапной потери веса или нет. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые имели эпизод внезапной потери веса.

На графике 10 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, был ли у пациента эпизод чрезмерного/экстремального голода или нет. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые болели полифагией.

На графике 11 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, был ли у пациента эпизод помутнения зрения. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые имели эпизод помутнения зрения.

На графике 12 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, был ли у пациента эпизод зуда. Из графика нельзя сделать каких-либо выводов, распределение групп примерно одинаковое.

На графике 13 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, было ли замечено у пациента замедленное заживление ран. Из графика нельзя сделать каких-либо выводов, распределение групп примерно одинаковое.

На графике 14 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, был ли у пациента эпизод ослабления мышцы/группы мышц или нет. Из графика видно, что диабетом на ранних стадиях страдали пациенты, которые сталкивались с частичным парезом.

На графике 15 представлено разбиение выборки по целевой переменной, а именно наличие диабета на ранних стадиях, в зависимости от того, был ли у пациента эпизод мышечной ригидности или нет. Из графика нельзя сделать каких-либо выводов, распределение групп примерно одинаковое.

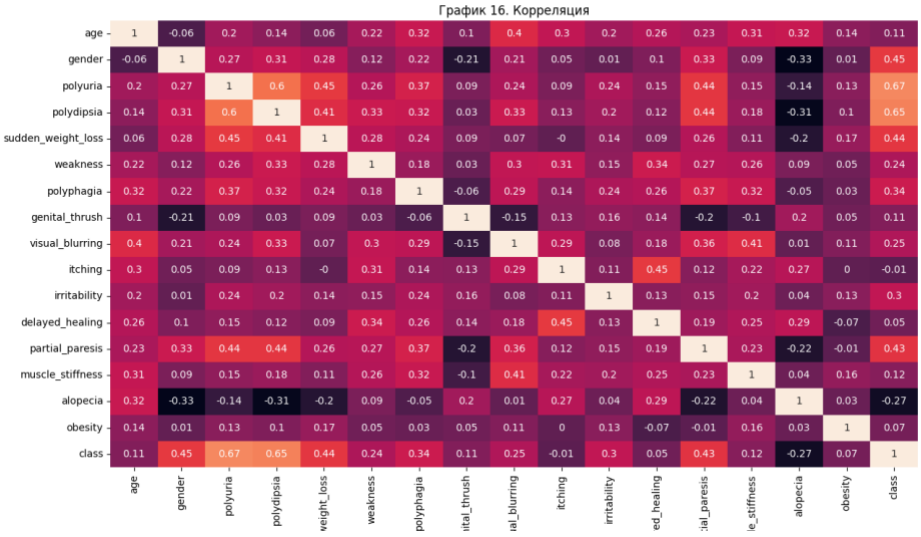


График 16. Корреляция

Из графика 16, видно, что наличие диабета на ранних стадиях сильно взаимосвязано с полиурией и полидипсией. Это же подтверждается графиками 7 и 8.

Построение описательной статистики

	age	gender	polyuria	polydipsia	sudden_weight_loss
count	520	520	520	520	520
mean	48,02884615	0,369230769	0,496153846	0,448076923	0,417307692
std	12,151466	0,483061233	0,500466656	0,497775547	0,493589407
min	16	0	0	0	0
25%	39	0	0	0	0
50%	47,5	0	0	0	0
75%	57	1	1	1	1
max	90	1	1	1	1

	weakness	polyphagia	genital_thrush	visual_blurring	itching
count	520	520	520	520	520
mean	0,586538462	0,455769231	0,223076923	0,448076923	0,486538462
std	0,492928353	0,498519373	0,416710388	0,497775547	0,500300043
min	0	0	0	0	0
25%	0	0	0	0	0
50%	1	0	0	0	0
75%	1	1	0	1	1
max	1	1	1	1	1

	irritability	delayed_healing	partial_paresis	muscle_stiffness
count	520	520	520	520
mean	0,242307692	0,459615385	0,430769231	0,375
std	0,428892086	0,498846305	0,495660732	0,484589094
min	0	0	0	0
25%	0	0	0	0
50%	0	0	0	0
75%	0	1	1	1
max	1	1	1	1

	muscle_stiffness	alopecia	obesity	class
count	520	520	520	520
mean	0,375	0,344230769	0,169230769	0,615384615
std	0,484589094	0,475574275	0,375316674	0,486972724
min	0	0	0	0
25%	0	0	0	0
50%	0	0	0	1
75%	1	1	0	1
max	1	1	1	1

3. Подготовка данных

Преобразование столбца "gender" в числовой формат, с использованием следующей схемы кодирования: ‘Male’ = 0, ‘Female’ = 1.

4. Моделирование

Были построены следующие модели:

	Логистическая регрессия	Метод ближайшего соседа	Дерево решений	Случайный лес
Accuracy	0.9519230769230769	0.8269230769230769	0.9230769230769231	0.9711538461538461
Recall	0.96875	0.765625	0.890625	0.984375
Precision	0.9538461538461539	0.9423076923076923	0.9827586206896551	0.9692307692307692
F1	0.9612403100775194	0.8448275862068966	0.9344262295081968	0.9767441860465116

5. Оценка результатов

Наилучший результат показала модель случайного леса. Метрики качества удовлетворяют заданным.

6. Внедрение.

Рекомендуем внедрить наилучшую модель, а именно модель случайного леса с параметрами: min_samples_leaf = 6, min_samples_split = 6. Принятие модели в эксплуатацию зависит от заказчика.

Приложение 1. Код программы Python для проведенного анализа

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, recall_score, precision_score, f1_score
from sklearn.preprocessing import LabelEncoder


def log_reg (x_train, x_test, y_train, y_test):
    classif = LogisticRegression()
    classif.fit(x_train, y_train)
    y_pred = classif.predict(x_test)
    print('=========log regres==========')
```

```
    print(accuracy_score(y_test, y_pred))
    print(recall_score(y_test, y_pred))
    print(precision_score(y_test, y_pred))
    print(f1_score(y_test, y_pred))
    return (y_pred)


def KNN (x_train, x_test, y_train, y_test):
    classif = KNeighborsClassifier(n_neighbors=6)
    classif.fit(x_train, y_train)
    y_pred = classif.predict(x_test)
    print('==========KNN==========')
    print(accuracy_score(y_test, y_pred))
    print(recall_score(y_test, y_pred))
    print(precision_score(y_test, y_pred))
    print(f1_score(y_test, y_pred))
    return (y_pred)


def Dec_tree (x_train, x_test, y_train, y_test):
    classif = DecisionTreeClassifier(min_samples_leaf=6, min_samples_split=6)
    classif.fit(x_train, y_train)
    y_pred = classif.predict(x_test)
    print('==========DT==========')
    print(accuracy_score(y_test, y_pred))
    print(recall_score(y_test, y_pred))
    print(precision_score(y_test, y_pred))
    print(f1_score(y_test, y_pred))
    return (y_pred)


def RF (x_train, x_test, y_train, y_test):
    classif = RandomForestClassifier(min_samples_leaf=6, min_samples_split=6)
    classif.fit(x_train, y_train)
    y_pred = classif.predict(x_test)
    print('==========RF==========')
    print(accuracy_score(y_test, y_pred))
    print(recall_score(y_test, y_pred))
    print(precision_score(y_test, y_pred))
    print(f1_score(y_test, y_pred))
    return (y_pred)


df = pd.read_csv('diabetes_data.csv', sep=';')
df['gender'] = df['gender'].replace(['Male', 'Female'], [0, 1])
```

```
df.info()

describe = df.describe()
describe.to_csv('describe.csv')

fig1, axes = plt.subplots (nrows=1, ncols=3, figsize=(16, 8))
sns.countplot(ax = axes[0], x = 'class', data = df)
axes[0].title.set_text('График 1. Наличие диабета')
sns.countplot(ax = axes[1], x = 'class', data = df, hue = 'gender')
axes[1].title.set_text ('График 2. Пол')
sns.countplot(ax = axes[2], x = 'class', data = df, hue = 'age')
axes[2].title.set_text ('График 3. Возраст')

fig2, axes = plt.subplots (nrows=1, ncols=3, figsize=(16, 8))
sns.countplot(ax = axes[0], x = 'class', data = df, hue = 'alopecia')
axes[0].title.set_text ('График 4. Облысение')
sns.countplot(ax = axes[1], x = 'class', data = df, hue = 'obesity')
axes[1].title.set_text ('График 5. Ожирение')
sns.countplot(ax = axes[2], x = 'class', data = df, hue = 'weakness')
axes[2].title.set_text ('График 6. Постоянная слабость')

fig3, axes = plt.subplots (nrows=1, ncols=3, figsize=(16, 8))
sns.countplot(ax = axes[0], x = 'class', data = df, hue = 'polyuria')
axes[0].title.set_text ('График 7. Полюрия')
sns.countplot(ax = axes[1], x = 'class', data = df, hue = 'polydipsia')
axes[1].title.set_text ('График 8. Полидипсия')
sns.countplot(ax = axes[2], x = 'class', data = df, hue = 'sudden_weight_loss')
axes[2].title.set_text ('График 9. Неожиданная потеря веса')

fig4, axes = plt.subplots (nrows=1, ncols=3, figsize=(16, 8))
sns.countplot(ax = axes[0], x = 'class', data = df, hue = 'polyphagia')
axes[0].title.set_text ('График 10. Полифагия')
sns.countplot(ax = axes[1], x = 'class', data = df, hue = 'visual_blurring')
axes[1].title.set_text ('График 11. Помутнение зрения')
sns.countplot(ax = axes[2], x = 'class', data = df, hue = 'itching')
axes[2].title.set_text ('График 12. Зуд')

fig5, axes = plt.subplots (nrows=1, ncols=3, figsize=(16, 8))
sns.countplot(ax = axes[0], x = 'class', data = df, hue = 'delayed_healing')
axes[0].title.set_text ('График 13. Медленная регенерация')
sns.countplot(ax = axes[1], x = 'class', data = df, hue = 'partial_paresis')
axes[1].title.set_text ('График 14. Частичный парез')
sns.countplot(ax = axes[2], x = 'class', data = df, hue = 'muscle_stiffness')
axes[2].title.set_text ('График 15. Ригидность мышц')
```

```
fig6, axes = plt.subplots (nrows=1, ncols=1, figsize=(16, 8))
axes.title.set_text ('График 16. Корреляция')
sns.heatmap(df.corr().round(2), annot=True, cbar=False)
plt.show()

x_train, x_test, y_train, y_test = train_test_split(df[df.columns[:-1]],
df[df.columns[-1]],
                                                    train_size = 0.8, random_state = 0)
y1 =log_reg(x_train, x_test, y_train, y_test)
y2 = KNN(x_train, x_test, y_train, y_test)
y3 = Dec_tree(x_train, x_test, y_train, y_test)
y4 = RF(x_train, x_test, y_train, y_test)
```