

Алексей Михнин

“

**СНАТГРТ ДЛЯ
САМОРАЗВИТИЯ:
ОСВОЕНИЕ
ПРОГРАММИРОВАНИЯ,
КОНСАЛТИНГА И
ИЗУЧЕНИЕ
ИНОСТРАННЫХ
ЯЗЫКОВ**

”

Алексей Михнин

ChatGPT для саморазвития: Освоения программирования, консалтинга и изучения иностраннх языков

http://www.litres.ru/pages/biblio_book/?art=69628900

SelfPub; 2023

Аннотация

Эта книга посвящена возможностям использования чат-бота ChatGPT для самообразования и повышения производительности в различных областях. В ней подробно рассматриваются примеры применения чат-бота для изучения программирования на Python, АВАР. Для консультирования в сфере SAP ERP, для создания макросов в Excel, а также для изучения английского языка. Книга демонстрирует пошаговые примеры обучения чата выполнению конкретных задач, анализирует качество ответов и дает рекомендации по улучшению эффективности общения с чат-ботом. Книга содержит множество практических советов по формулировке запросов, интерпретации результатов и применению полученных знаний на практике. Особое внимание уделено сравнению возможностей бесплатной и платной версий чат-бота. Приводятся рекомендации по настройке чат-бота с

помощью промтов для получения более точных и подробных ответов. Книга будет полезна всем, кто хочет повысить свою квалификацию и производительность с помощью искусственного интеллекта.

Содержание

Введение	5
Шаг1 Загрузка и изучение данных	11
Шаг2 Предварительная обработка данных	14
Шаг3 Визуализация и статистический данных	22
Шаг 4 Построение модели машинного обучения	38
Выводы	46
Конец ознакомительного фрагмента.	47

Алексей Михнин ChatGPT для саморазвития: Освоения программирования, консалтинга и изучения иностраных языков

Введение

Добро пожаловать в книгу о том, как использовать ChatGPT для саморазвития в различных областях! В этой книге вы узнаете, как использовать чата для обучения программированию на языке Python, АВАР, для изучения макросов в MS Excel и для повышения уровня английского языка и не только.

В разделе, посвященном изучению Python, мы преподнесем вам уникальную возможность научиться задавать правильные вопросы, а также продемонстрируем вам полный процесс машинного обучения, начиная с загрузки данных, предобработки, статистического анализа и заканчивая обу-

чением модели машинного обучения.

Одним из примеров, которые мы рассмотрим, будет модель, содержащая данные пассажиров Титаника. Используя свои собственные данные, вы сможете предсказать вероятность выживания на корабле благодаря тому, что мы покажем вам, как использовать ChatGPT для обучения модели машинного обучения.

Затем мы перейдем к изучению языка программирования АВАР, где наш ChatGPT будет играть роль программиста, а вы будете проверять его код в системе SAP и создавать отчеты согласно функциональной спецификации.

В разделе, посвященном макросам в MS Excel, наш ChatGPT будет выступать в роли специалиста по макросам, который будет писать готовый код на основе ваших команд и пожеланий, чтобы облегчить вам работу по анализу данных и подготовке красивых отчетов.

В разделе посвящённому изучению иностранных языков, мы покажем вам, как использовать ChatGPT для изучения английского языка, где чат выступит в роли личного репетитора.

Кроме того, в конце книги вы узнаете о том, как продуктивно использовать ChatGPT для продолжения вашего саморазвития. Мы покажем вам, где и как можно применять чат для повышения ваших компетенций и навыков, которые напрямую влияют на вашу производительность и успех в работе.

Вы также узнаете о преимуществах использования ChatGPT в качестве консультанта, который может помочь вам в решении различных задач и проблем, связанных с программированием, анализом данных, работой в SAP и других областях.

Таким образом, читая эту книгу и осваивая различные навыки, вы сможете стать более продуктивным и эффективным в своей работе, достигая новых высот в своей карьере.

Будьте готовы к увлекательному путешествию по саморазвитию с помощью ChatGPT!

Изучаем

Python

– с

ChatGPT

Для демонстрации возможностей ChatGPT в обучении программированию на языке Python, мы предлагаем вам пройти пошаговый сквозной пример машинного обучения, используя набор данных (dataset), содержащий данные о пассажирах Титаника. Файл `titanic.csv` содержит информацию о 887 пассажирах, включая их выживаемость, возраст, класс пассажира, пол и стоимость проезда.

Вы, в качестве пользователя, будете задавать вопросы на русском языке чат-боту, который будет предоставлять развернутые ответы с решением, которые Вы заносите в среду разработки, в данном случае – Jupyter lab, и проверяете результаты и адекватность ответов.

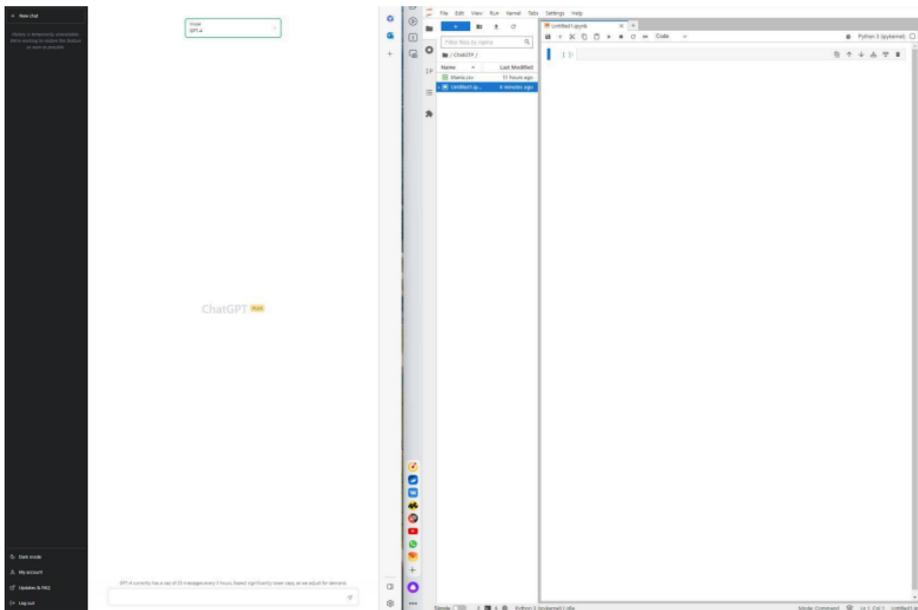
По результатам прохождения сквозного примера, Вы будете наблюдать, как правильно писать вопросы и как взаимодействовать с чат-ботом. В конце данной главы вы сможете уже самостоятельно не только анализировать большие объёмы бизнес-данных на языке python но и познакомитесь с основами машинного обучения затратив минимальное время и сэкономив на курсах по обучению python которые стоят десятки тысяч рублей.

Файл с данными о пассажирах (Титаника), который мы будем использовать в нашем сквозном примере для обучения – вы можете скачать бесплатно на сатей kaggle.

Дано:

В левой части перед нами открытый чат ChatGPT, регистрация на данном ресурсе бесплатная. Как зарегистрироваться для граждан России (см. раздел Секреты – регистрация на сайте ChatGPT)

В правой части экрана представлена среда разработки (Colab) от google для языка Python – доступ к данной среде предоставляется бесплатно. Как только регистрируетесь в среде разработки Colab, не забываем загрузить туда наш файл с данными (датасет) по Титанику.



После того как мы зарегистрировались в ChatGPT и получили доступ в Colab и разобрались как загрузить туда наш датасет, можно приступать к нашему увлекательному путешествию.

В следующих главах этой книги вы будете замечать слова "Пользователь" – это означает, что мы будем писать запросы и команды для передачи в ChatGPT от вашего лица. А когда вы получите ответ, вы увидите слово "Чат", которое будет означать ответ ChatGPT на наши запросы и команды.

Кроме того, мы будем обращать ваше внимание на важные моменты, которые могут помочь вам работать с ChatGPT более продуктивно и эффективно. Эти моменты будут помечены

ны словом "Внимание", чтобы вы смогли легко заметить их и улучшить свой опыт работы с ChatGPT.

Когда мы говорим о "Результате", мы имеем в виду тестирование решения, предложенного ChatGPT, в соответствующей среде разработки.

Для языка программирования Python мы будем использовать среду разработки Colab/Jupyter для тестирования наших решений. В случае работы с SAP мы будем использовать среду разработки SAP ERP, а для работы с MS Excel – MS Excel.

Мы убедимся, что предложенное решение, созданное с помощью ChatGPT, проходит тестирование в соответствующей среде разработки, что гарантирует его готовность для практического применения в вашей работе.

Шаг1 Загрузка и изучение данных

На этом шаге мы попробуем загрузить информацию из нашего датасета в среду разработки и получить общую информацию о данных.

Это позволит нам более детально изучить и понять наш набор данных, что в свою очередь поможет нам выбрать правильные инструменты и методы для анализа данных и решения поставленных задач.

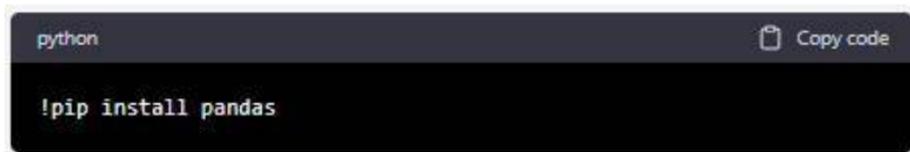
Пользователь:

Напиши мне код на python – Открой файл `titanic.csv` с данными и изучите общую информацию по данному файлу и представь все в табличном виде.

Чат:

Для выполнения этого задания мы будем использовать библиотеку `pandas`.

Если у вас ее нет, установите с помощью команды:

A screenshot of a terminal window with a dark background. The top bar shows 'python' on the left and a 'Copy code' button on the right. The main area contains the command '!pip install pandas' in a light-colored font.

```
python Copy code  
  
!pip install pandas
```

Теперь, когда у вас есть библиотека `pandas`, следуйте приведенному ниже коду для открытия файла `'titanic.csv'` и вы-

вода общей информации о данных в табличном виде:

```
python Copy code  
  
import pandas as pd  
  
# Загрузка данных из файла CSV  
file_path = 'titanic.csv'  
data = pd.read_csv(file_path)  
  
# Вывод общей информации о данных  
print(data.info())  
  
# Вывод первых 10 строк данных для предварительного просмотра  
print(data.head(10))
```

После выполнения этого кода вы увидите общую информацию о данных, такую как количество строк, столбцов, типы данных и количество непустых значений. Также вы увидите первые 10 строк данных для предварительного просмотра.

Учтите, что вам нужно будет заменить 'titanic.csv' на действительный путь к файлу CSV на вашем компьютере или URL-адрес файла, если он доступен онлайн.

Результат:

Выполняем все рекомендации Чата и посмотрим посмотрим что из этого получилось.

```
Untitled1.ipynb x +
Python 3 (ipykernel)

[17]: !pip install pandas

Requirement already satisfied: pandas in /home/mih30/miniconda3/lib/python3.10/site-packages (1.5.2)
Requirement already satisfied: python-dateutil<=2.8.1 in /home/mih30/miniconda3/lib/python3.10/site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /home/mih30/miniconda3/lib/python3.10/site-packages (from pandas) (2022.7)
Requirement already satisfied: numpy<=1.21.0 in /home/mih30/miniconda3/lib/python3.10/site-packages (from pandas) (1.23.5)
Requirement already satisfied: six>=1.5 in /home/mih30/miniconda3/lib/python3.10/site-packages (from python-dateutil<=2.8.1->pandas) (1.16.0)

[18]: import pandas as pd

# Загрузка данных из файла CSV
file_path = "titanic.csv"
data = pd.read_csv(file_path)

# Вывод общей информации о данных
print(data.info())

# Вывод первых 10 строк данных для предварительного просмотра
print(data.head(10))

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 887 entries, 0 to 886
Data columns (total 8 columns):
 # Column          Non-Null Count  Dtype
---  ---
 0 Survived         887 non-null   int64
 1 Pclass          887 non-null   int64
 2 Name            887 non-null   object
 3 Sex             887 non-null   object
 4 Age            887 non-null   float64
 5 Siblings/Spouses Aboard 887 non-null   int64
 6 Parents/Children Aboard 887 non-null   int64
 7 Fare           887 non-null   float64
dtypes: float64(4), int64(4), object(2)
memory usage: 55.6+ KB

None
Survived  Pclass          Name \
0      0      3                Mr. Owen Harris Braund
1      1      1  Mrs. John Bradley (Florence Briggs Thayer) Cum...
2      1      3                Miss. Laina Meikkinen
3      1      1  Mrs. Jacques Heath (Lily May Peel) Futrelle
4      0      3                Mr. William Henry Allen
5      0      3                Mr. James Moran
6      0      1                Mr. Timothy J McCarthy
7      0      3                Master. Gosta Leonard Palsson
8      1      3  Mrs. Oscar W (Elisabeth Vilhelmina Berg) Johnson
9      1      2                Mrs. Nicholas (Adele Achem) Nasser

Sex  Age  Siblings/Spouses Aboard  Parents/Children Aboard  Fare
0  male  22.0                1                0  7.2500
1  female  38.0                1                0  71.2833
2  female  26.0                0                0  7.9250
3  female  35.0                1                0  53.1000
4  male  35.0                0                0  8.0500
5  male  27.0                0                0  8.4583
6  male  54.0                0                0  51.8625
7  male  2.0                3                1  21.0750
8  female  27.0                0                2  11.1333
9  female  14.0                1                0  30.0780
```

Шаг2 Предварительная обработка данных

Давайте попробуем предварительно обработать данные на этом шаге. Мы проверим на наличие дубликатов, пропущенных значений и другие аномалии. Мы не знаем, какое качество данных находится в нашем файле на входе, поэтому этот шаг очень важен для последующего анализа данных. Предварительная обработка позволит нам избавиться от мусорных значений и повысит качество нашего анализа.

Пользователь

Выведи все названия столбцов в одну строку и выполни шаги по предварительной обработке данных т.е. проверь на дубликаты, на наличие пропущенных значений и т.п. и выведи отчет по результатам предварительной обработки.

Чат:

Для выполнения этой задачи мы будем использовать библиотеку `pandas`.

Приведенный ниже код выполняет предварительную обработку данных и выводит отчет по результатам:

```
import pandas as pd

# Загрузка данных из файла CSV
file_path = 'titanic.csv'
data = pd.read_csv(file_path)

# Вывод названий столбцов
print("Названия столбцов:")
print(', '.join(data.columns))

# Проверка на дубликаты
duplicate_rows = data.duplicated().sum()
print(f"\nКоличество дубликатов: {duplicate_rows}")
if duplicate_rows > 0:
    data.drop_duplicates(inplace=True)
    print("Дубликаты удалены.")

# Проверка на пропущенные значения
missing_values = data.isnull().sum()
print("\nКоличество пропущенных значений в каждом столбце:")
print(missing_values)

# Заполнение пропущенных значений (по одному примеру для каждого столбца)
# Здесь мы заполняем пропущенные значения медианным значением для чисел
```

python

Copy code

```
# Заполнение пропущенных значений (по одному примеру для каждого столбца)
# Здесь мы заполняем пропущенные значения медианным значением для чисел

for column in data.columns:
    if data[column].dtype == 'object':
        data[column].fillna(data[column].mode().iloc[0], inplace=True)
    else:
        data[column].fillna(data[column].median(), inplace=True)

# Проверка на пропущенные значения после заполнения
missing_values_after_filling = data.isnull().sum()
print("\nКоличество пропущенных значений после заполнения:")
print(missing_values_after_filling)

# Вывод отчета о предварительной обработке данных
print("\nОтчет о предварительной обработке данных:")
print(f"Удалено дубликатов: {duplicate_rows}")
print(f"Заполнено пропущенных значений: {missing_values.sum()}")

# Вывод первых 10 строк обработанных данных для предварительного пр
```

python

Copy code

```
print("\nПервые 10 строк обработанных данных:")
print(data.head(10))
```

После выполнения этого кода вы получите отчет о предварительной обработке данных, включая количество удаленных дубликатов и заполненных пропущенных значений. Код также выводит первые 10 строк обработанных данных для предварительного просмотра.

Учтите, что вам нужно будет заменить 'titanic.csv' на действительный путь к файлу CSV на вашем компьютере или URL-адрес файла, если он доступен онлайн.

Результат:

Выполняем все рекомендации Чата и посмотрим, что из этого получилось.

```
[19]: import pandas as pd

# Загрузка данных из файла CSV
file_path = 'titanic.csv'
data = pd.read_csv(file_path)

# Вывод названий столбцов
print("Названия столбцов:")
print(', '.join(data.columns))

# Проверка на дубликаты
duplicate_rows = data.duplicated().sum()
print(f"Количество дубликатов: {duplicate_rows}")
if duplicate_rows > 0:
    data.drop_duplicates(inplace=True)
    print("Дубликаты удалены.")

# Проверка на пропущенные значения
missing_values = data.isnull().sum()
print(f"Количество пропущенных значений в каждом столбце:")
print(missing_values)

Названия столбцов:
Survived, Pclass, Name, Sex, Age, Siblings/Spouses Aboard, Parents/Children Aboard, Fare

Количество дубликатов: 0

Количество пропущенных значений в каждом столбце:
Survived      0
Pclass        0
Name          0
Sex           0
Age           0
Siblings/Spouses Aboard  0
Parents/Children Aboard  0
Fare          0
dtype: int64
```

• [20]: # Заполнение пропущенных значений (по одному примеру для каждого столбца)
 # Здесь мы заполняем пропущенные значения медианным значением для числовых столбцов и модой для категориальных столбцов.
 # Вы можете изменить это в соответствии с вашими потребностями.

```
for column in data.columns:
    if data[column].dtype == 'object':
        data[column].fillna(data[column].mode().iloc[0], inplace=True)
    else:
        data[column].fillna(data[column].median(), inplace=True)

# Проверка на пропущенные значения после заполнения
missing_values_after_filling = data.isnull().sum()
print("\nКоличество пропущенных значений после заполнения:")
print(missing_values_after_filling)

# Вывод отчета о предварительной обработке данных
print("\nОтчет о предварительной обработке данных:")
print(f"Удалено дубликатов: {duplicate_rows}")
print(f"Заполнено пропущенных значений: {missing_values.sum()}")

# Вывод первых 10 строк обработанных данных для предварительного пр
print("\nПервые 10 строк обработанных данных:")
print(data.head(10))
```

Количество пропущенных значений после заполнения:

```
Survived      0
Pclass        0
Name          0
Sex           0
Age           0
Siblings/Spouses Aboard  0
Parents/Children Aboard  0
Fare          0
dtype: int64
```

Отчет о предварительной обработке данных:

```
Удалено дубликатов: 0
Заполнено пропущенных значений: 0
```

Первые 10 строк обработанных данных:

Survived	Pclass	Name
0	3	Mr. Owen Harris Braund
1	1	Mrs. John Bradley (Florence Briggs Thayer) Cum...
2	1	Miss. Laina Heikkinen
3	1	Mrs. Jacques Heath (Lily May Peel) Futrelle
4	3	Mr. William Henry Allen
5	3	Mr. James Moran
6	1	Mr. Timothy J McCarthy
7	3	Master. Gosta Leonard Palsson
8	3	Mrs. Oscar W (Elisabeth Vilhelmina Berg) Johnson
9	2	Mrs. Nicholas (Adele Achen) Nasser

Sex	Age	Siblings/Spouses Aboard	Parents/Children Aboard	Fare
0	male	22.0	1	0 7.2500
1	female	38.0	1	0 71.2833
2	female	26.0	0	0 7.9250
3	female	35.0	1	0 53.1000
4	male	35.0	0	0 8.0500
5	male	27.0	0	0 8.4583
6	male	54.0	0	0 51.8625
7	male	2.0	3	1 21.0750
8	female	27.0	0	2 11.1333
9	female	14.0	1	0 30.0708

Внимание:

Мы заметили, что при каждом ответе ChatGPT повторяется код, который устанавливает библиотеку pandas для работы с табличными данными, и загружает данные из нашего датасета в переменную data. Чтобы избежать повторения этого кода, мы можем поступить следующим образом:

Пропустить эти строки кода и использовать только код, который расположен ниже.

Конкретизировать наши запросы, напоминая ChatGPT о том, что библиотека pandas уже установлена, и что файл titanic.csv уже загружен в переменную data.

Кроме того, мы рекомендуем напоминать ChatGPT о том, какие названия столбцов у нас есть в датасете, и какого они типа. Это поможет нам точнее работать со столбцами в нашем датасете. Для этого мы можем попросить ChatGPT вывести названия столбцов и их типы в одну строку, которую мы сможем скопировать и использовать для конкретизации наших последующих запросов.

Пользователь:

Выведи в одну строку название столбцов и тип каждого столбца

Внимание:

Мы не указали, какой именно датасет мы должны использовать в данном запросе, потому что мы формируем запрос в текущем чате обсуждения, и ChatGPT будет анализировать

предыдущие шаги, чтобы понять, какой датасет мы используем. Однако, если в текущем чате было выполнено много шагов или если есть несколько альтернативных датасетов, то необходимо конкретизировать запрос, указав, какой датасет нужно использовать.

Чат:

Для вывода названия столбцов и типа каждого столбца, используйте следующий код:

```
python Copy code  
  
import pandas as pd  
  
# Загрузка данных из файла CSV  
file_path = 'titanic.csv'  
data = pd.read_csv(file_path)  
  
# Вывод названий столбцов и типов данных  
column_info = ', '.join([f'{column}: {data[column].dtype}' for column  
print(column_info)
```

Результат:

Выполняем все рекомендации Чата и посмотрим что из этого получилось.

```
[23]: # Вывод названий столбцов и типов данных
column_info = ', '.join(['{column}: {data[column].dtype}' for column in data.columns])
print(column_info)
```

```
Survived: int64, Pclass: int64, Name: object, Sex: object, Age: float64, Siblings/Spouses Aboard: int64, Parents/Children Aboard: int64, Fare: float64
```

Внимание:

Как вы заметили, мы пропустили строки кода с импортом библиотеки `pandas` и загрузкой данных из CSV файла. Мы воспользовались ранее предложенным советом и пропустили эти строки кода, чтобы избежать повторения одних и тех же команд.

Шаг3 Визуализация и статистический данных

Для построения различных графиков, которые могут помочь визуализировать и провести статистический анализ данных, мы просим Чат предоставить нам соответствующий код.

Такой код позволит нам быстро и удобно построить необходимые графики и провести необходимый анализ данных.

Благодаря этому мы сможем получить более глубокое понимание нашего датасета и извлечь более полезные выводы из данных.

Пользователь:

Библиотека pandas уже установлена и наш файл titanic.csv уже загружен в переменную data.

Файл titanic.csv содержит следующие столбцы:

Survived: int64, Pclass: int64, Name: object, Sex: object, Age: float64, Siblings/Spouses Aboard: int64, Parents/Children Aboard: int64, Fare: float64

Выполни визуализацию данных построив гистограмму, box plot, bar plot, scatter plot и корреляционную матрицу.

Чат:

Если данные уже загружены в переменную data, вы можете использовать следующий код для визуализации данных с

помощью гистограмм, box plot, bar plot, scatter plot и корреляционной матрицы:

```
import matplotlib.pyplot as plt
import seaborn as sns

# Гистограмма для числовых признаков
numeric_columns = ['Survived', 'Pclass', 'Age', 'Siblings/Spouses Aboard']
data[numeric_columns].hist(figsize=(10, 8), bins=20)
plt.tight_layout()
plt.show()

# Box plot для числовых переменных
for column in numeric_columns:
    plt.figure()
    sns.boxplot(x=data[column])
    plt.title(f'Boxplot for {column}')
    plt.show()

# Bar plot для категориальных переменных
categorical_columns = ['Survived', 'Pclass', 'Sex']
for column in categorical_columns:
    plt.figure()
    sns.countplot(x=data[column])
    plt.title(f'Bar plot for {column}')
    plt.show()

# Scatter plot для числовых переменных
scatter_columns = ['Age', 'Fare']
sns.pairplot(data[scatter_columns])
plt.show()

# Корреляционная матрица
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.show()
```

Этот код создаст графики визуализации, включая гистограммы, box plot, bar plot, scatter plot и корреляционную матрицу для различных столбцов данных. Гистограммы и box plot используются для числовых переменных, в то время как bar plot используется для категориальных переменных. Scatter plot и корреляционная матрица позволяют наблюдать взаимосвязь между числовыми переменными.

Выполнение кода создаст графики, которые помогут вам лучше понять распределение данных, взаимосвязь между переменными и выявить возможные выбросы или аномалии в данных.

Результат:

Выполняем все рекомендации Чата и посмотрим посмотрим что из этого получилось.

```
[24]: import matplotlib.pyplot as plt
import seaborn as sns

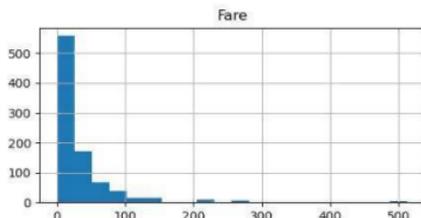
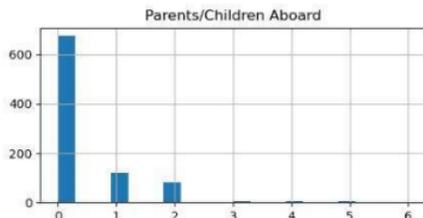
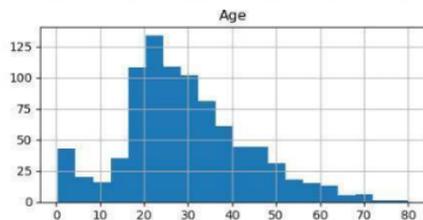
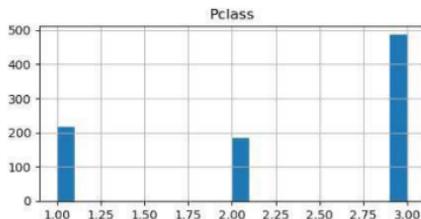
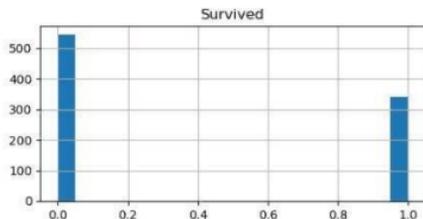
# Гистограмма для числовых признаков
numeric_columns = ['Survived', 'Pclass', 'Age', 'Siblings/Spouses Aboard', 'Parents/Children Aboard', 'Fare']
data[numeric_columns].hist(figsize=(10, 8), bins=20)
plt.tight_layout()
plt.show()

# Box plot для числовых переменных
for column in numeric_columns:
    plt.figure()
    sns.boxplot(x=data[column])
    plt.title(f'Boxplot for {column}')
    plt.show()

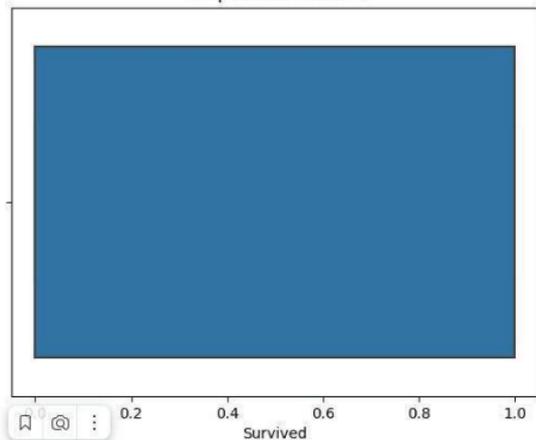
# Bar plot для категориальных переменных
categorical_columns = ['Survived', 'Pclass', 'Sex']
for column in categorical_columns:
    plt.figure()
    sns.countplot(x=data[column])
    plt.title(f'Bar plot for {column}')
    plt.show()

# Scatter plot для числовых переменных
scatter_columns = ['Age', 'Fare']
sns.pairplot(data[scatter_columns])
plt.show()

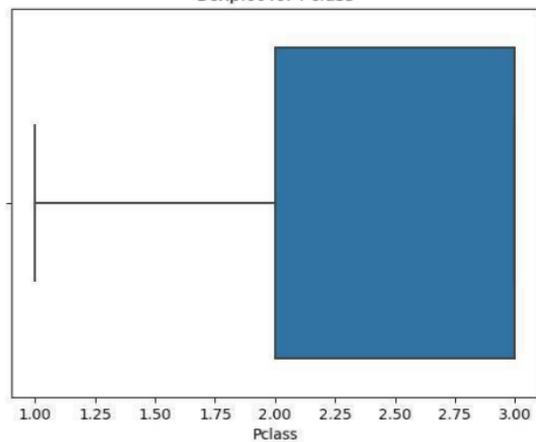
# Корреляционная матрица
sns.heatmap(data.corr(), annot=True, cmap='coolwarm')
plt.show()
```



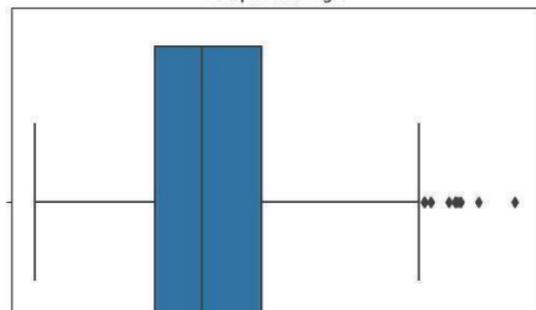
Boxplot for Survived



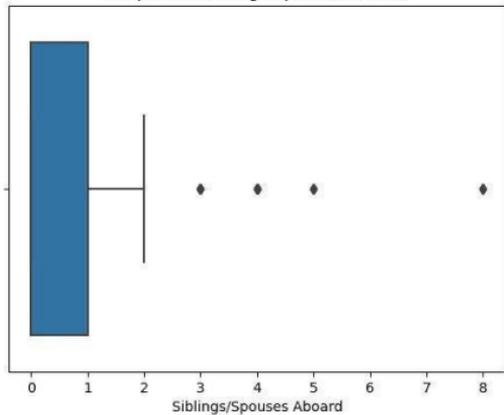
Boxplot for Pclass



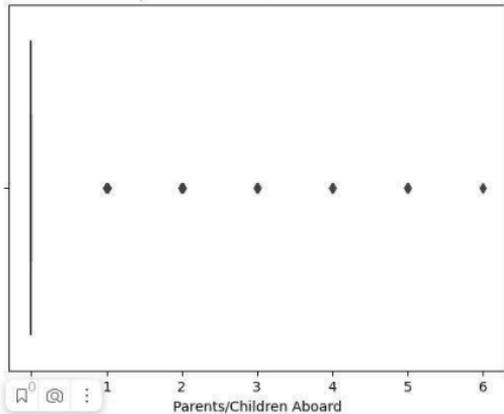
Boxplot for Age



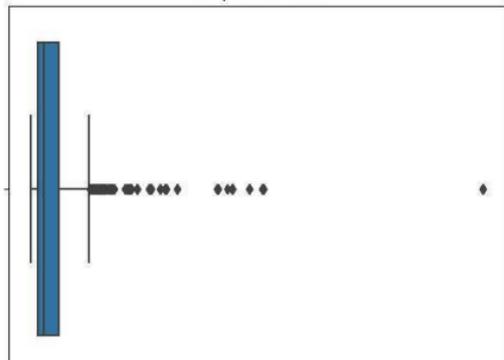
Boxplot for Siblings/Spouses Aboard



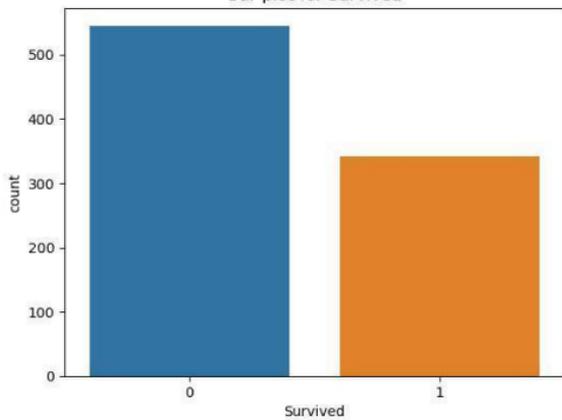
Boxplot for Parents/Children Aboard



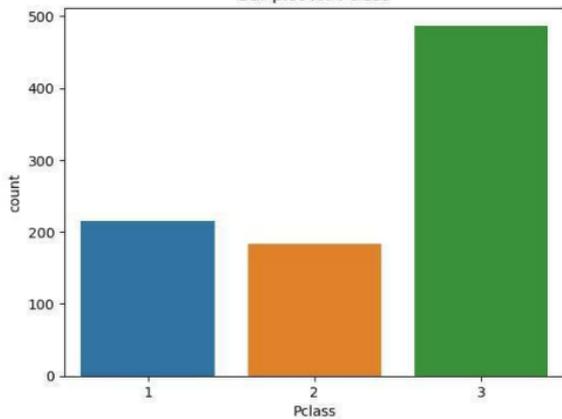
Boxplot for Fare



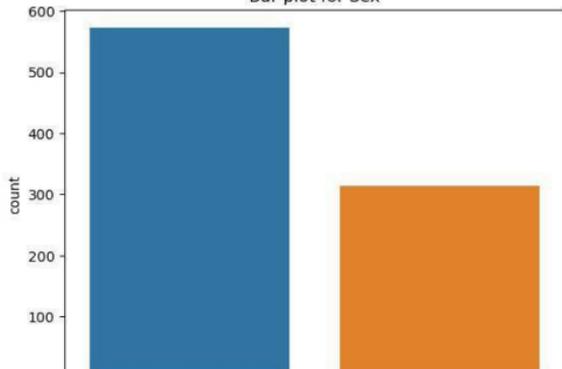
Bar plot for Survived

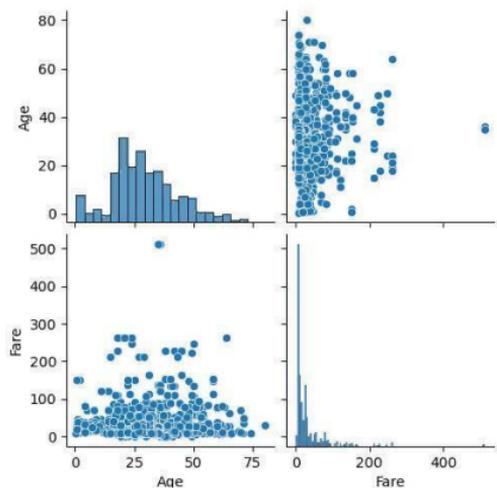


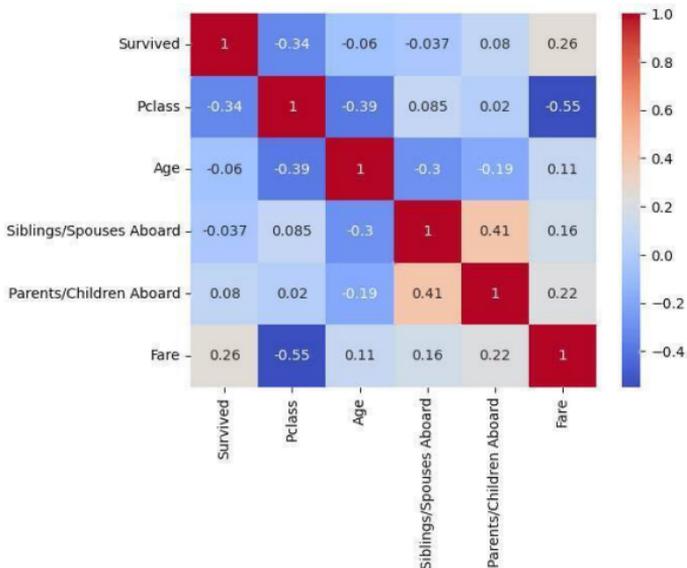
Bar plot for Pclass



Bar plot for Sex







Пользователь:

Ты можешь анализировать графики?
как их загрузить тебе для анализа?

Чат:

К сожалению, как AI, я не могу напрямую анализировать изображения или графики, загруженные пользователем. Однако, если у вас есть вопросы о том, как интерпретировать графики или какие выводы можно сделать на основе графиков, я могу помочь вам с объяснениями и рекомендациями.

Вы можете задать вопросы, касающиеся графиков, и я постараюсь дать вам подробные ответы и объяснения.

Примеры вопросов могут быть следующими:

Как интерпретировать корреляционную матрицу?

Что такое выбросы на box plot, и что они могут означать для данных?

Какие выводы можно сделать из гистограммы или bar plot?

Пожалуйста, задайте свои вопросы, и я с радостью помогу вам разобраться в анализе графиков.

Пользователь:

Как интерпретировать корреляционную матрицу?

Что такое выбросы на box plot, и что они могут означать для данных?

Какие выводы можно сделать из гистограммы или bar plot?

Чат:

Корреляционная матрица – это таблица, которая показывает корреляционные коэффициенты между парами переменных. Корреляционные коэффициенты – это числовые значения, которые измеряют степень связи между двумя переменными.

Коэффициент корреляции может принимать значения от -1 до 1. Коэффициенты, близкие к 1, означают положительную корреляцию, т.е. если значение одной переменной увеличивается, то увеличивается и значение другой переменной. Коэффициенты, близкие к -1, означают отрицательную корреляцию, т.е. если значение одной переменной увеличивается, то значение другой переменной уменьшается. Ко-

эффиценты, близкие к 0, означают отсутствие корреляции между переменными.

Box plot – это график, который показывает распределение данных и позволяет выявлять выбросы. Выбросы – это значения, которые сильно отличаются от остальных значений в выборке. Выбросы могут означать, что данные содержат ошибки, ошибки измерения или аномальные значения. Также выбросы могут быть результатом редкого события или непредвиденных условий, которые необходимо учитывать при анализе данных.

Гистограмма – это график, который показывает распределение данных по определенным интервалам или классам. Она может помочь понять форму распределения данных, например, нормальное, скошенное или мультимодальное распределение. Bar plot – это график, который используется для сравнения количественных данных между несколькими категориями. Он может показать, какие категории имеют более высокие или более низкие значения, и помочь выявить различия в данных между категориями. Выводы из гистограммы или bar plot могут помочь понять, какие категории или значения в данных являются наиболее значимыми или важными для анализа.

Внимание:

Как вы уже заметили, ChatGPT не только предоставляет вам код с решением, но и может подробно консультировать вас, объяснять сложные термины и помогать сделать выводы

и заключения.

Это значительно упрощает процесс обучения и решения задач, позволяя вам получать не только готовые решения, но и понимать, как они работают и почему они правильны. Это очень полезно для тех, кто только начинает изучать программирование или машинное обучение, так как помогает закрепить теоретические знания на практике.

Шаг 4 Построение модели машинного обучения

Далее в книге мы предоставляем код, который поможет вам построить модель машинного обучения на основе подготовленного датасета и обучить ее предсказывать значения столбца "Survived" (выживший). С помощью этой модели вы сможете оценить вероятность своего выживания, если бы вы оказались на Титанике в тот роковой день.

Пользователь:

Библиотека `pandas`, `matplotlib`, `seaborn` уже установлены и наш файл `titanic.csv` уже загружен в переменную `data`.

Файл `titanic.csv` содержит следующие столбцы:

`Survived: int64`, `Pclass: int64`, `Name: object`, `Sex: object`, `Age: float64`, `Siblings/Spouses Aboard: int64`, `Parents/Children Aboard: int64`, `Fare: float64`

Столбец `Survived` – это целевая переменная, которую необходимо научиться предсказывать на основании переменных `Pclass`, `Name`, `Sex`, `Age`, `Siblings/Spouses Aboard`, `Parents/Children Aboard`, `Fare`

Построй модель на языке `python` и обучи ее.

Предобработка данных на проверку дублей, пустых значений уже выполнена.

Чат:

Поскольку предобработка данных уже выполнена, приступим к построению и обучению модели. В качестве модели я выберу случайный лес (Random Forest) из библиотеки `sklearn`.

Сначала разделим данные на обучающую и тестовую выборки, а затем обучим модель.

python

Copy code

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

# Кодирование категориальных признаков
data["Sex"] = data["Sex"].map({"male": 0, "female": 1})

# Выбор признаков для обучения
features = ["Pclass", "Sex", "Age", "Siblings/Spouses Aboard", "Parent
X = data[features]
y = data["Survived"]

# Разделение данных на обучающую и тестовую выборки
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.

# Создание и обучение модели случайного леса
model = RandomForestClassifier(n_estimators=100, random_state=42)
model.fit(X_train, y_train)

# Предсказание и оценка точности модели
y_pred = model.predict(X_test)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy:
```

MI

продолжай



Прошу прощения, кажется, мой предыдущий ответ был обрезан. Вот продолжение кода:



python

Copy code

```
print(f"Accuracy: {accuracy:.2f}")
```

Теперь, когда модель обучена, вы можете использовать её для предсказания выживаемости на новых данных.

Пример:

```
python Copy code

# Новые данные
new_data = pd.DataFrame({
    "Pclass": [3],
    "Sex": [0], # 0 - мужчина, 1 - женщина
    "Age": [25],
    "Siblings/Spouses Aboard": [0],
    "Parents/Children Aboard": [0],
    "Fare": [7.25]
})

# Предсказание выживаемости
survived_prediction = model.predict(new_data)
print(f"Survived prediction: {survived_prediction[0]}")
```

Таким образом, вы можете использовать обученную модель для предсказания выживаемости на основе данных о пассажирах.

Внимание:

Как вы заметили на представленном выше скриншоте, пользователь использовал слово "продолжай". Мы используем это слово в случае, когда по каким-то неизвестным нам

причинам ChatGPT не завершает свой ответ до конца и обрывается на полпути.

Написание слова "продолжай" воспринимается чатом как инструкция продолжить с того места, где он остановился. Однако, если мы напишем это слово, когда чат уже завершил свою мысль, то он воспримет это как просьбу дать нам еще примеров или информации по данному запросу.

Результат:

Выполняем все рекомендации Чата и посмотрим что из этого получилось.

```
[25]: from sklearn.model_selection import train_test_split
      from sklearn.ensemble import RandomForestClassifier
      from sklearn.metrics import accuracy_score

      # Кодирование категориальных признаков
      data["Sex"] = data["Sex"].map({"male": 0, "female": 1})

      # Выбор признаков для обучения
      features = ["Pclass", "Sex", "Age", "Siblings/Spouses Aboard", "Parents/Children Aboard", "Fare"]
      X = data[features]
      y = data["Survived"]

      # Разделение данных на обучающую и тестовую выборки
      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

      # Создание и обучение модели случайного леса
      model = RandomForestClassifier(n_estimators=100, random_state=42)
      model.fit(X_train, y_train)

      # Предсказание и оценка точности модели
      y_pred = model.predict(X_test)
      accuracy = accuracy_score(y_test, y_pred)
      print(f"Accuracy: {accuracy:.2f}")

Accuracy: 0.76

[26]: # Новые данные
      new_data = pd.DataFrame({
          "Pclass": [3],
          "Sex": [0], # 0 - мужчина, 1 - женщина
          "Age": [25],
          "Siblings/Spouses Aboard": [0],
          "Parents/Children Aboard": [0],
          "Fare": [7.25]
      })

      # Предсказание выживаемости
      survived_prediction = model.predict(new_data)
      print(f"Survived prediction: {survived_prediction[0]}")

Survived prediction: 0
```

Пользователь:

Что означает Accuracy: 0.76 и Survived prediction: 0

Чат:

Accuracy: 0.76 означает, что точность вашей модели составляет 76%. Точность (accuracy) – это доля правильно классифицированных объектов от общего количества объектов. В данном случае, модель правильно предсказала выживаемость пассажиров в 76% случаев на тестовой выборке. Это одна из метрик оценки качества классификации.

Survived prediction: 0 относится к примеру с новыми данными о пассажире. Значение 0 указывает на то, что модель предсказывает, что этот пассажир не выживет в случае кораблекрушения. В данной задаче классификации значение 0 соответствует классу "не выжил", а значение 1 соответствует классу "выжил".

Пользователь:

Давайте попробуем сделать изменения в коде самостоятельно, без запросов в ChatGPT. Мы заменим значения для предсказания на свои собственные, вместо мужчины мы поставим женщину, а все остальные параметры оставим без изменений.

Результат:

Выполняем все изменения и посмотрим что из этого получилось.

```
[27]: # Новые данные
new_data = pd.DataFrame({
    "Pclass": [3],
    "Sex": [1], # 0 - мужчина, 1 - женщина
    "Age": [25],
    "Siblings/Spouses Aboard": [0],
    "Parents/Children Aboard": [0],
    "Fare": [7.25]
})

# Предсказание выживаемости
survived_prediction = model.predict(new_data)
print(f"Survived prediction: {survived_prediction[0]}")

Survived prediction: 1
```

Получается, что женщина с теми же параметрами выжила. Это не удивительно, учитывая тот факт, что на борту спасательных шлюпок в первую очередь спасали женщин и детей. Мы можем сделать вывод, что пол пассажира был одним из факторов, влияющих на выживаемость в данном случае.

Пользователь:

Как добиться более высокой точности модели?

Чат:

Для достижения более высокой точности модели можно выполнить следующие шаги:

Инжиниринг признаков:

Создайте новые признаки на основе существующих данных. Например, вы можете создать признак "IsAlone" (один путешествует), если количество родственников на борту (Siblings/Spouses Aboard + Parents/Children Aboard) равно нулю.

```
data["IsAlone"] = (data["Siblings/Spouses Aboard"] + data["Parents/Children Aboard"]).apply(lambda x: 1 if x == 0
```

else 0)

Отбор признаков:

Оцените значимость признаков и исключите незначимые или шумные признаки из обучения.

Оптимизация гиперпараметров:

Подберите оптимальные значения гиперпараметров модели, например, используя поиск по сетке (`GridSearchCV`) или случайный поиск (`RandomizedSearchCV`) из `sklearn`.

Дополнительно – рекомендации автора:

Для более полного понимания метрик машинного обучения, их разнообразия и правильного выбора в различных ситуациях, мы рекомендуем ознакомиться с книгой того же автора под названием "Оценка качества моделей машинного обучения: выбор, интерпретация и применение метрик".

Также, если вам необходимо овладеть процессом машинного обучения для табличных данных и узнать, какие вопросы стоит задавать чат-боту при обработке таких данных, мы рекомендуем прочитать книгу этого же автора "Табличное Мастерство: Осваиваем Модели Машинного Обучения для Анализа Табличных Данных". Обе книги помогут вам овладеть профессиональными навыками в области машинного обучения и эффективно применять их на практике.

Выводы

Мы успешно прошли полный процесс машинного обучения на языке Python, начиная с загрузки и предобработки данных, их визуализации, разделения на обучающую и тестовую выборки, обучения модели и тестирования ее на данных, а также объяснения результатов и возможных дальнейших шагов.

Конец ознакомительного фрагмента.

Текст предоставлен ООО «ЛитРес».

Прочитайте эту книгу целиком, [купив полную легальную версию](#) на ЛитРес.

Безопасно оплатить книгу можно банковской картой Visa, MasterCard, Maestro, со счета мобильного телефона, с платежного терминала, в салоне МТС или Связной, через PayPal, WebMoney, Яндекс.Деньги, QIWI Кошелек, бонусными картами или другим удобным Вам способом.