УДК 004.8

МАШИННОЕ ОБУЧЕНИЕ: АНАЛИЗ, ВИЗУАЛИЗАЦИЯ И ОБУЧЕНИЕ МОДЕЛИ (НА ПРИМЕРЕ ДАТАСЕТА «ТИТАНИК»)

Ларин С.Э.

студент,

ФГБОУ ВО «Калужский государственный университет

им. К.Э. Циолковского»

Калуга, Россия

Белаш В.Ю.

к.пед.н., доцент,

ФГБОУ ВО «Калужский государственный университет

им. К.Э. Циолковского»

Калуга, Россия

Аннотация: Статья охватывает актуальные аспекты анализа данных с применением языка программирования Python. Описаны основные этапы анализа данных: обработка, детальный анализ, визуализация данных. На основе всех этих этапов была построена модель машинного обучения, которая предсказывает выживаемость пассажиров с точностью 82%.

Ключевые слова: анализ данных, Python, машинное обучение, обработка данных, информационные технологии.

MACHINE LEARNING: ANALYSIS, VISUALIZATION AND MODEL TRAINING (BASED ON THE EXAMPLE OF THE TITANIC DATASET)

Larin S.E.

student,

Kaluga State University named after K. E. Tsiolkovsky

Kaluga, Russia

Belash V.Yu.
Ph.D., Associate Professor,
Kaluga State University named after K. E. Tsiolkovsky
Kaluga, Russia

Abstract: The article covers current aspects of data analysis using the Python programming language. The main stages of data analysis are described: processing, detailed analysis, data visualization. Based on all these steps, a machine learning model was built that predicts passenger survival with 82% accuracy.

Keywords: data analysis, Python, machine learning, data processing, information technology.

В данной статье рассматривается известный учебный датасет "Титаник", ориентированный на начинающих в области машинного обучения. Цель исследования заключается в разработке модели, способной с высокой точностью предсказывать выживание пассажиров на основе доступных атрибутов. Основные признаки, такие как возраст, пол и класс билета, рассматриваются как ключевые атрибуты для выполнения задачи предсказания выживаемости.

Для достижения данной цели необходимо выполнить следующие задачи:

- Предобработка данных: данные считываются из CSV-файла и загружаются в объект DataFrame при использовании библиотеки pandas. Этот этап включает в себя обнаружение и обработку пропущенных значений, выбор стратегии заполнения пропусков для столбцов, а также удаление несущественных признаков. Предобработка данных проведена авторами в статье [3].
- Исследовательский анализ: после предобработки данных проводится анализ распределения, центральных тенденций и вариаций

различных признаков. Этот этап включает в себя визуализацию данных с использованием графиков, что способствует глубокому пониманию структуры данных.

- Выбор модели и обучение: на основе результатов анализа выбираются наиболее подходящие модели машинного обучения. Эти модели обучаются на тестовом наборе данных, и их производительность оценивается с использованием различных метрик.

В результате выполнения задачи будет получена модель, способная предсказывать выживаемость пассажиров Титаника на основе доступных атрибутов. Составление данной модели может быть полезна для тренировки навыков по анализу данных и использования их на другом датасете.

Рассмотрим реализацию поставленных задач. Первая задача – предобработка данных. Необходимо определить количество мужчин и женщин на борту. Для выполнения данной операции используем код: df["Sex"].value_counts(). В ходе его выполнения был выведен результат – мужчин (577) и женщин (314) (рис. 1).

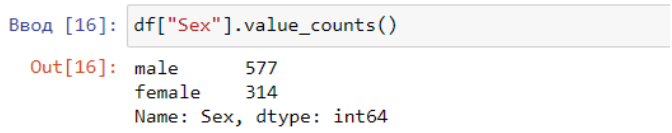

Рис. 1. Количество пассажиров[1]

В Титанике было три класса билетов. Чтобы получить информацию о том, сколько было пассажиров в каждом классе используем метод value.counts(). Если необходимо узнать сколько мужчин и женщин было в каждом классе используется маска df["Pclass"]df["Sex"] == male].value_counts(), которая выбирает только те строки, где “Sex” равен male(мужчины) или female(женщины) (рис.2).

[1] составлено авторами

```
Ввод [17]: df["Pclass"].value_counts() #распределение переменной Pclass по всем классам

Out[17]: 3    491
         1    216
         2    184
         Name: Pclass, dtype: int64

Ввод [18]: df["Pclass"][df["Sex"] == "male"].value_counts() #распределение переменной Pclass по всем классам только для мужчин

Out[18]: 3    347
         1    122
         2    108
         Name: Pclass, dtype: int64

Ввод [19]: df["Pclass"][df["Sex"] == "female"].value_counts()#распределение переменной Pclass по всем классам только для женщин

Out[19]: 3    144
         1     94
         2     76
         Name: Pclass, dtype: int64
```

Рис. 2. Количество пассажиров в классах[2]

Далее рассмотрим вопрос относительно соотношения выживших мужчин и женщин после крушения. Подобные данные удобно представлять с использованием диаграмм. Для этого перейдем к демонстрации второй задачи работы – исследовательскому анализу.

Для визуализации необходимо использовать библиотеку Seaborn, чтобы создать столбчатую диаграмму (countplot) на основе атрибута «Sex». Параметр «hue» используется для того, чтобы разделить данные по категориям. В данном случае будет разделение по категориям мужчины и женщины в соответствии со значением в столбце «Survived» (рис.3).

[2] составлено авторами

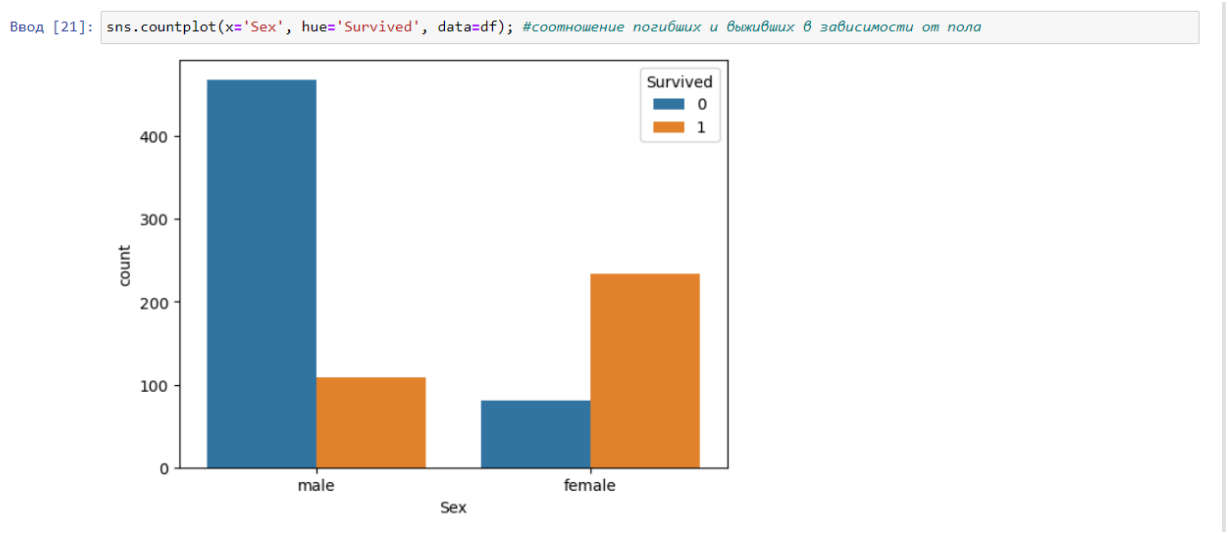

Рис. 3. Графическое представление данных о выживших[3]

Графическое представление данных указывает на тот факт, что наибольший процент выживших приходится на женщин.

Предположим, что люди из 1 класса выжили с более высокой вероятностью по сравнению с другими, поскольку 1 класс является самым дорогим и наилучшим в обслуживании.

Рассмотрим соотношение погибших и выживших в зависимости от класса каюты. Для этого используется код: sns.countplot(x='Pclass', hue='Survived', data=df); В результате получена столбчатая диаграмма, где по оси X представлены классы кают, которые разделены по цветам (рис.4).

[3] составлено авторами

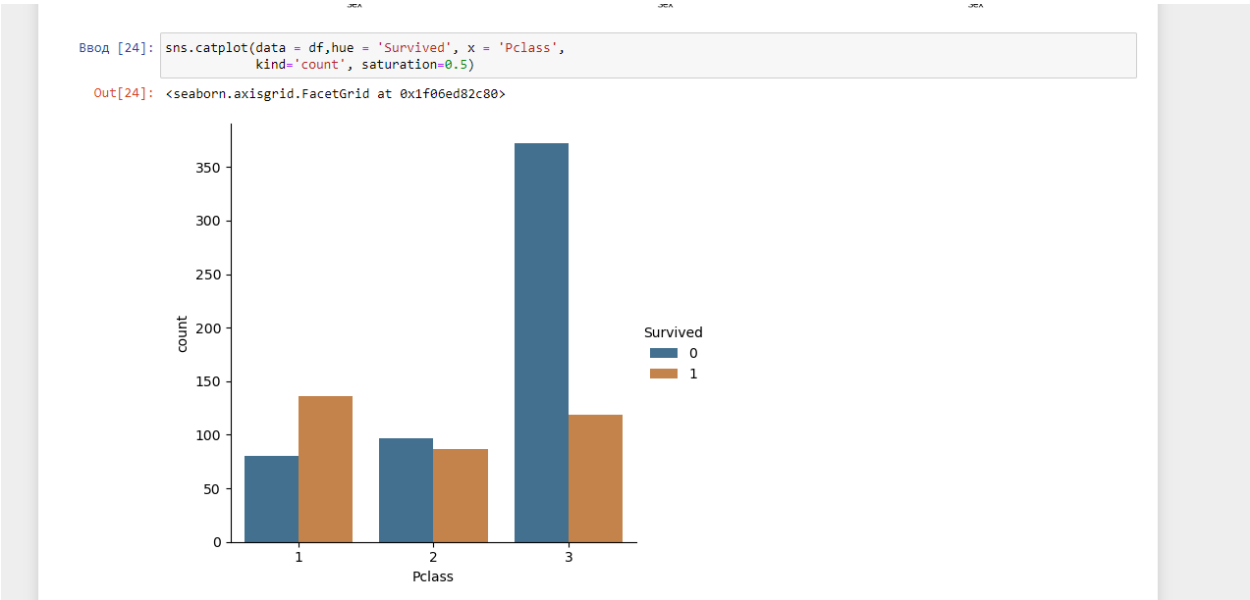

Рис. 4. Графическое представление данных о выживших по классам[4]

Как и предполагалось больше всего выживших людей именно из 1 класса.

Третья задача работы – выбор модели и обучение. Перед тем как обучать модель большинство алгоритмов машинного обучения требуют именно числовые значения в качестве входных данных, поэтому необходимо изменить столбцы с типом object, а также удалить столбцы PassengerId, Ticket и Name, так как от них не зависит выживаемость.

С помощью словаря закодируем значения в «Sex»: d2 = {"male": 0, "female": 1}, где мужской пол кодируется 0, а женский 1 (рис.5).

```
Ввод [37]: del df["Name"]

Ввод [38]: del df["Ticket"]

Ввод [39]: del df["PassengerId"]

Ввод [40]: d2 = {"male": 0, "female": 1}
           df["Sex"] = df["Sex"].map(d2)
```

Рис. 5. Доработка данных[5]

[4] составлено авторами
[5] составлено авторами

Теперь необходимо обозначить целевую переменную и разбить данные на тренировочную и тестовую выборки. Целевой переменной является «Survived» (выжившие). Код x = df_scal.copy() создает копию датафрейма и присваивается переменной x, с целью избежания воздействия на первоначальный набор данных и сохранения их неизменными. Целевую переменную обозначим y.

Для разделения на тренировочный и тестовый набор используется функция train_test_split из библиотеки scikit-learn. Параметр test_size=0.3 определяет долю из данных, которая выделяется в тестовый набор. Значение 0.3 представляет собой размер 30%, а остальные 70% будут использоваться в обучении. Параметр random_state определяет начальное значение для генератора случайных чисел внутри функции, то есть начинает свою последовательность с указанного числа. При каждом запуске этого кода с указанным значением 1 результат будет одинаковым, что полезно для воспроизводимости результата. Чтобы сохранить баланс при обучении и тестировании модели необходимо использовать параметр stratify. Таким образом, в тренировочном наборе представлено 549 не выживших и 342 выживших (рис.6).

```
Ввод [44]: x = df_scal.copy() # копирует массив в x

Ввод [45]: y = df["Survived"] # y - целевая переменная

Ввод [46]: from sklearn.model_selection import train_test_split

Ввод [47]: x_train, x_test, y_train, y_test = train_test_split(x, y, test_size = 0.3, random_state = 1, stratify = y)

Ввод [48]: y.value_counts()

Out[48]: 0    549
         1    342
         Name: Survived, dtype: int64
```

Рис. 6. Разбиение на тестовую и тренировочную выборки[6]

После проделанных этапов можно переходить к обучению модели. В качестве примера рассмотрим метод к-ближайших соседей, поскольку данная

[6] составлено авторами

модель является простой, интуитивной и чувствительной к масштабу признаков.

В коде, где происходит инициализация и обучение модели:

knn_clf = KNeighborsClassifier(n_neighbors=31)

knn_clf.fit(x_train, y_train)

Создается объект KNeighborsClassifier с параметром n_neighbors=31, указывающим, что для классификации будет учитываться 31 ближайший сосед. Затем модель обучается на обучающем наборе данных (x_train и y_train), где x_train – это матрица признаков, а y_train – вектор меток классов (рис.7).

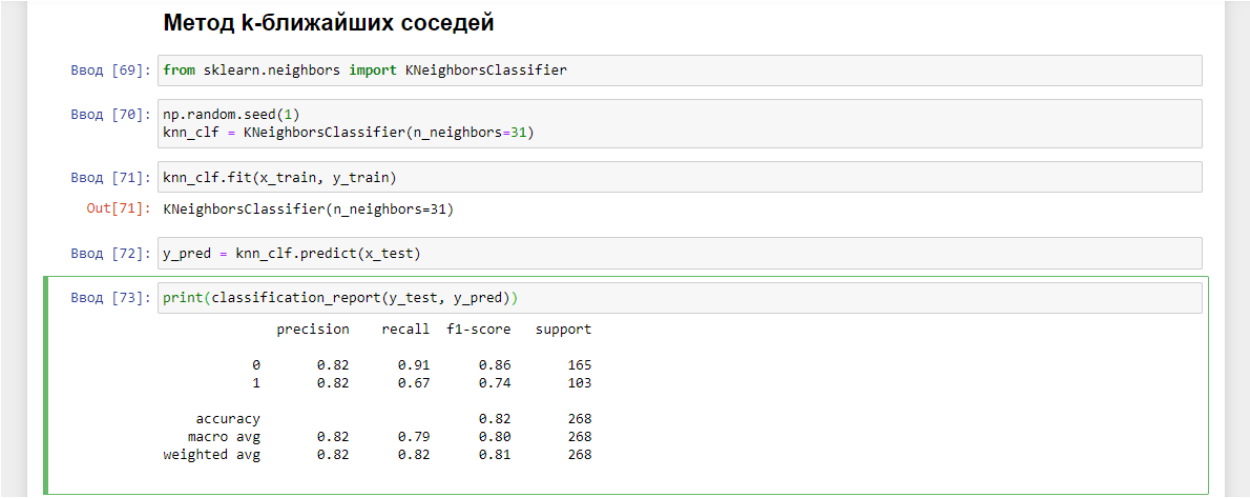

Рис. 7. Метод к-ближайших соседей[7]

В результате представлены метрики оценки производительности модели классификации на тестовом наборе данных. Основные метрики: precision (точность), recall (полнота), f1-score(мера) и accuracy(точность).

Метрики показывают, насколько хорошо работает модель. Например, высокая полнота для класса 0, указывает что модель отлично определяет отсутствие выживших.

[7] составлено авторами

Данная модель предсказывает выживаемость пассажиров с точностью 82%, что является достаточно неплохим результатом.

В данной статье был проведен анализ данных датасета Титаник для предсказания выживаемости пассажиров. В результате анализа были выявлены следующие факторы, влияющие на выживаемость:

- Пол: женщины выживали с большей вероятностью, чем мужчины.
- Класс билета: пассажиры первого класса выживали с большей вероятностью, чем пассажиры второго и третьего классов.

На основе этих факторов была построена модель машинного обучения, которая предсказывает выживаемость пассажиров с точностью 82%.

Данная модель может быть использована для понимания факторов, влияющих на выживаемость в чрезвычайных ситуациях, а также для разработки методов повышения выживаемости людей в подобных ситуациях.

Библиографический список

1. Data Visualization with Titanic / [Электронный ресурс] // Kaggle: [сайт]. – URL: https://www.kaggle.com/code/enessasmaz/data-visualization-with-titanic (дата обращения: 28.11.2023).

2. Анализ данных: основные этапы процесса правильной обработки информации / [Электронный ресурс] // apipython: [сайт]. – URL: https://apipython.ru/analiz-dannyh-osnovnye-etapy-proczessa-pravilnoj-obrabotki-informaczii/ (дата обращения: 28.11.2023).

3. Ларин С.Э., Белаш В.Ю. Библиотеки Python для анализа данных: предобработка и подготовка данных // Дневник науки. 2023. №12 [Электронный ресурс]. – URL: http://www.dnevniknauki.ru/images/publications/2023/12/technics/Larin_Belash.pdf (Дата обращения 09.01.2024).

4. Введение в анализ данных / [Электронный ресурс] // miptstats.github.io: [сайт]. – URL: https://mipt-stats.gitlab.io/courses/ad_fivt/titanik.html (дата обращения: 28.11.2023).

5. Разбор задачи Титаник / [Электронный ресурс] // habr: [сайт]. – URL: https://habr.com/ru/articles/655955/ (дата обращения: 24.11.2023).

Оригинальность 90%