МИНОБРНАУКИ РОССИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ   
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ   
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

(ФГБОУ ВО «ВГУ»)

Факультет прикладной математики, информатики и механики

Кафедра математических методов исследования операций

**Отчет**

о прохожденииучебной практики, проектной

Направление 01.03.02 – Прикладная математика и информатика

Сроки прохождения практики: 06.07.2024 – 19.07.2024

|  |  |
| --- | --- |
| Обучающийся 2к.7 гр. | Сидоров П.П. |
| Руководитель | к.ф.-м.н., Замятин И.В. |

Воронеж 2024

Оглавление

[Введение 3](#_Toc172091128)

[1. Теоретическая часть 4](#_Toc172091129)

[1.1. Немного о языке python 4](#_Toc172091130)

[1.2. Машинное обучение и подготовка данных 4](#_Toc172091131)

[1.3. Библиотеки и функции, используемые для подготовки данных 5](#_Toc172091132)

[2. Практическая часть 6](#_Toc172091133)

[2.1. Работа с кодом 6](#_Toc172091134)

[Заключение 10](#_Toc172091135)

# Введение

Машинное обучение является одним из наиболее динамично развивающихся направлений в области искусственного интеллекта и анализа данных. Оно позволяет автоматизировать процессы принятия решений на основе анализа больших объемов информации. Однако, прежде чем начать применять алгоритмы машинного обучения, необходимо тщательно обработать и подготовить данные. Этот процесс включает в себя несколько ключевых этапов: сбор данных, их очистку, преобразование, выборку и кодирование признаков, а также оценку качества данных.

Цель данной практической работы — научиться эффективно работать с данными для задач машинного обучения. Мы рассмотрим основные методы предварительной обработки данных, которые помогут вам улучшить качество модели машинного обучения и повысить точность ее прогнозов. В процессе выполнения заданий мы получим практический опыт работы с реальными наборами данных, что является важным шагом на пути к профессии аналитика данных или инженера по машинному обучению.

После прохождения этой практической работы мы сможем самостоятельно обрабатывать и готовить данные к использованию в задачах машинного обучения, что является неотъемлемой частью процесса разработки решения на основе ИИ.

# 1. Теоретическая часть

## 1.1. Немного о языке python

Python — высокоуровневый, интерпретируемый, объектно-ориентированный язык программирования с динамической семантикой. Он известен своим простым и читаемым синтаксисом.

Python поддерживает модули и пакеты, что способствует модульности программ и повторному использованию кода. Благодаря отсутствию этапа компиляции, цикл редактирования-тестирования-отладки невероятно быстрый. Отладка программ на Python упрощается благодаря механизму исключений и возможности добавления операторов вывода.

Python широко используется в веб-разработке, обработке данных, системном скриптинге, математических расчетах, машинном обучении и многих других областях.

## 1.2. Машинное обучение и подготовка данных

Машинное обучение — это область искусственного интеллекта, которая использует статистические методы для создания моделей, способных делать прогнозы или принимать решения без явного программирования. Эти модели учатся на основе данных и могут адаптироваться к новым данным.

Подготовка данных — это критический этап в проектах машинного обучения. Она включает сбор, очистку, трансформацию и организацию сырых данных в формат, понятный алгоритмам машинного обучения.

Процесс подготовки данных начинается с сбора данных из различных источников, таких как базы данных, электронные таблицы или API. Затем данные очищаются от пропущенных значений, выбросов или несоответствий. Далее происходит трансформация данных через нормализацию и кодирование для совместимости с алгоритмами машинного обучения.

Подготовка данных является непрерывным процессом, а не однократной задачей. По мере развития вашей модели или получения новых данных, потребуется пересмотреть и уточнить шаги подготовки данных.

## 1.3. Библиотеки и функции, используемые для подготовки данных

NumPy предоставляет поддержку для больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами.

* array(): Создание массива;
* mean(), sum(), std(): Вычисление среднего значения, суммы и стандартного отклонения элементов массива соответственно.

Pandas — это библиотека для обработки и анализа данных, которая предоставляет структуры данных и функции для манипуляций с ними. Она особенно полезна для работы с табличными данными.

* DataFrame: Основная структура данных в Pandas, представляет собой двумерную таблицу с метками строк и столбцов;
* read\_csv(): Чтение данных из CSV-файла в DataFrame;
* dropna(): Удаление строк или столбцов с пропущенными значениями;
* fillna(): Заполнение пропущенных значений определенным значением или методом;
* set\_option(): Установка количества столбцов/строк на печать;
* DataFrame.head(): Печать первых 5 строк;
* DataFrame.tail():Печать указанного количества строк с конца;
* DataFrame.sample():Печать указанного количества строк случайно;
* DataFrame.info(): Общая информация о столбцах;
* DataFrame.describe(): Подробное описание столбцов;
* DataFrame.loc(): Копирование части столбцов датасета;
* Series: Структура данных столбца, представляет собой массив;
* Series.value\_count(): Сколько раз встречается каждое значение в таблице;
* Series.replace(): Замена одного значения на другое в столбце;
* Series.factorize(): сопоставление каждому значению соответствующего ему числа;

# 2. Практическая часть

В качестве практической задачи было необходимо выполнить предобработку данных для машинного обучения. Выполнять задачу было решено на наборе с данными о торнадо

Тип задачи - задача регрессии, прогноз на основе выборки объектов с различными признаками. Будем прогнозировать количество жертв урагана на основе данных о разрушениях, которые он принес и его размерах

Столбцы данных, с которыми будет вестись работа:

* fat – число погибших, целевой столбец;
* len – длина урагана (в милях);
* wid – ширина в ярдах;
* loss – информация о предполагаемых потерях имущества в долларах;
* st – двухбуквенное почтовое сокращение штата;
* ns – количество штатов, пострадавших от этого торнадо;
* inj – количество травм.

## 2.1. Работа с кодом

import numpy as np

import pandas as pd

Первое, что необходимо сделать при начале работы – это подключить нужные библиотеки. В данной работе используются библиотеки pandas и numpy.

data = pd.read\_csv('tornados.csv')

pd.set\_option('display.max\_columns', 100)

pd.set\_option('display.max\_rows', 1000)

Следующим действием загружаем датасет в программу при помощи функции read\_csv, и устанавливаем настройки вывода на экран – максимальное количество столбцов на экране 100, строк 1000.

data.head()

С помощью команды head печатаем на экран первые 5 строк из датасета (рис. 1.)

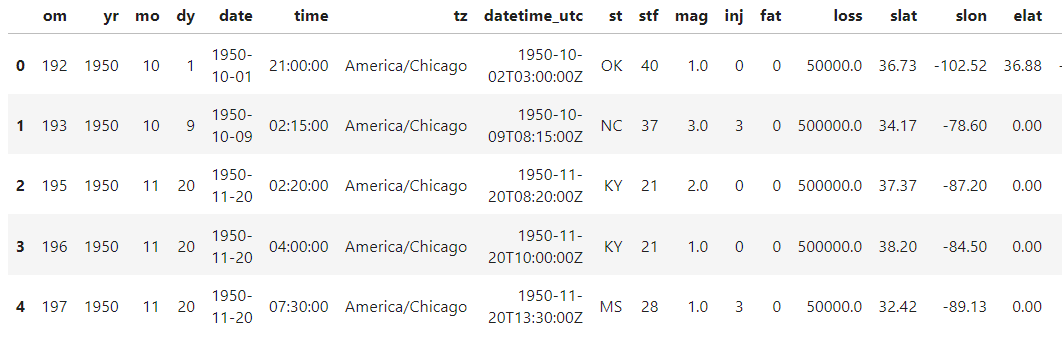


Рис. 1. Часть выведенных первых 5 строк датасета

data.tail(5)

data.sample(5)

С помощью команд tail, sample мы можем вывести на экран 5 последних и 5 случайных строк из нашего датасета. Выглядит аналогично печати head на рис. 1.

data.info()

Команда info позволяет получить такие данные о столбцах датасета, как название столбца, количество непустых значений и тип данных. Пример на рис. 2.

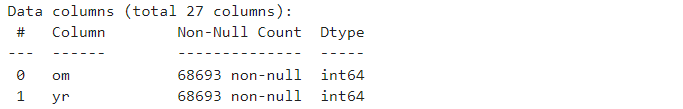


Рис. 2. Частичный результат команды info

data.describe()

data.describe(include="object")

Команда describe позволяет получить более подробную статистику по столбцу. Для числовых данных это количество записей, среднее значение, стандартное отклонение, минимум и максимум. Для объектов это количество записей, количество уникальных, самое часто встречающееся значение, и сколько раз оно встречается.



Рис. 3. Результат использования describe на числовых данных

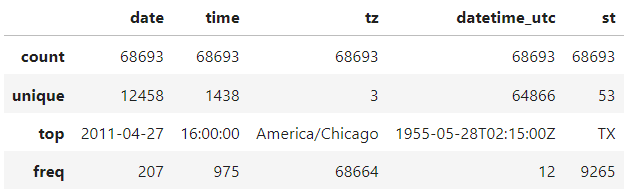


Рис. 4. Результат использования describe на объектах

selected = ["len", "wid", "loss", "st", "ns", "inj"]

data\_prepared = data.loc[:, data.columns.intersection(selected)]

В данных строчках идет выбор только тех столбцов, с которыми будет вестись работа. Сначала создается массив имен столбцов, потом с помощью команды loc создается копия исходного датасета, но с только нужными столбцами.

data['st'].value\_counts(normalize=True)

Здесь происходит просмотр, насколько часто встечается то или иное значение в столбце st (доля от общего числа).

data\_prepared['st'] = data\_prepared['st'].replace(['RI', 'AK', 'DC', 'VI'],'Other')

data\_prepared['st'].value\_counts(normalize=True)

Здесь происходит замена значений с малой долей вхождения на 1 общее ("прочее"). Это нужно для увеличения влияния этих столбцов на итоговый результат.

data\_prepared.st = data\_prepared.st.factorize()[0]

data\_prepared.head()

Далее на столбце st выполняется факторизация – замена категориальных данных на соответствующие им числа. Это необходимо, так как алгоритмы машинного обучения гораздо лучше работают с числами, чем с символами. После этого выводятся на экран первые 5 строк получившегося датасета.

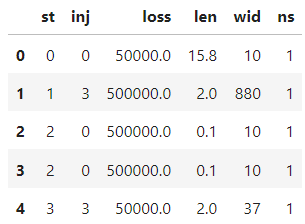


Рис. 5. Первые 5 строк получившегося набора данных

Далее было необходимо избавиться от пустых значений в датасете, потому что алгоритмы машинного обучения не приспособлены работать с None значениями. Пустые значения есть в столбце loss, и их надо убрать. Есть много способов это сделать, но здесь использовались два самых популярных.

res = data\_prepared.copy()

res.dropna(subset=['loss'], inplace=True)

Первый способ – удалить все строки, где встречаются пустые значения. Делается это с помощью функции dropna, которой в качестве параметра передается столбец, в котором и надо искать пустые значения.

res = data\_prepared.copy()

mean = res.loss.mean()

res.loss.fillna(mean, inplace=True)

Второй способ – заменить все пустые значения на другие, в данном случае на среднее значение по столбцу. Для этого с помощью функции mean находится среднее значение, и функцией fillna выполняется замена.

# Заключение

Работая с датасетом, содержащим информацию о торнадо, мы продемонстрировали важность предварительной обработки данных в контексте машинного обучения. Этот процесс включает в себя множество этапов, начиная от первоначальной загрузки и анализа данных до их очистки, преобразования и оптимизации для использования в алгоритмах машинного обучения.

Основываясь на примере датасета, мы увидели, как различные аспекты данных могут влиять на модели машинного обучения. Например, обработка пропущенных значений, нормализация числовых данных и кодирование категориальных переменных являются критически важными шагами для обеспечения качества входных данных для алгоритмов.

Также было важно учитывать специфику данных, связанные с природными катаклизмами, такие как торнадо. Эти данные часто характеризуются большими колебаниями и отсутствием некоторых значений, что требует особого подхода к их обработке. В нашем случае, мы успешно применили техники обработки данных для создания более чистого и понятного набора данных, который может быть использован для обучения модели предсказания количества жертв от торнадо.

Этот проект подчеркивает, что подготовка данных — это не просто технический этап, но и ключевой момент в развитии эффективного решения на основе машинного обучения. Он показывает необходимость глубокого понимания данных, с которыми вы работаете.

В заключение, работа с датасетом о торнадо предоставила ценный опыт в области предварительной обработки данных для машинного обучения. Это знание и навыки будут полезны в будущих проектах, где требуется анализ и интерпретация сложных наборов данных для решения различных задач.