Разработка документации по системе

Предоставлен экспертам соревнований World Skills Russia

Оглавление

[1. Модуль первый 3](#_Toc88729456)

[Что можно делать с API? 10](#_Toc88729457)

[Загружать данные в Prombota 10](#_Toc88729458)

[Как работает API 10](#_Toc88729459)

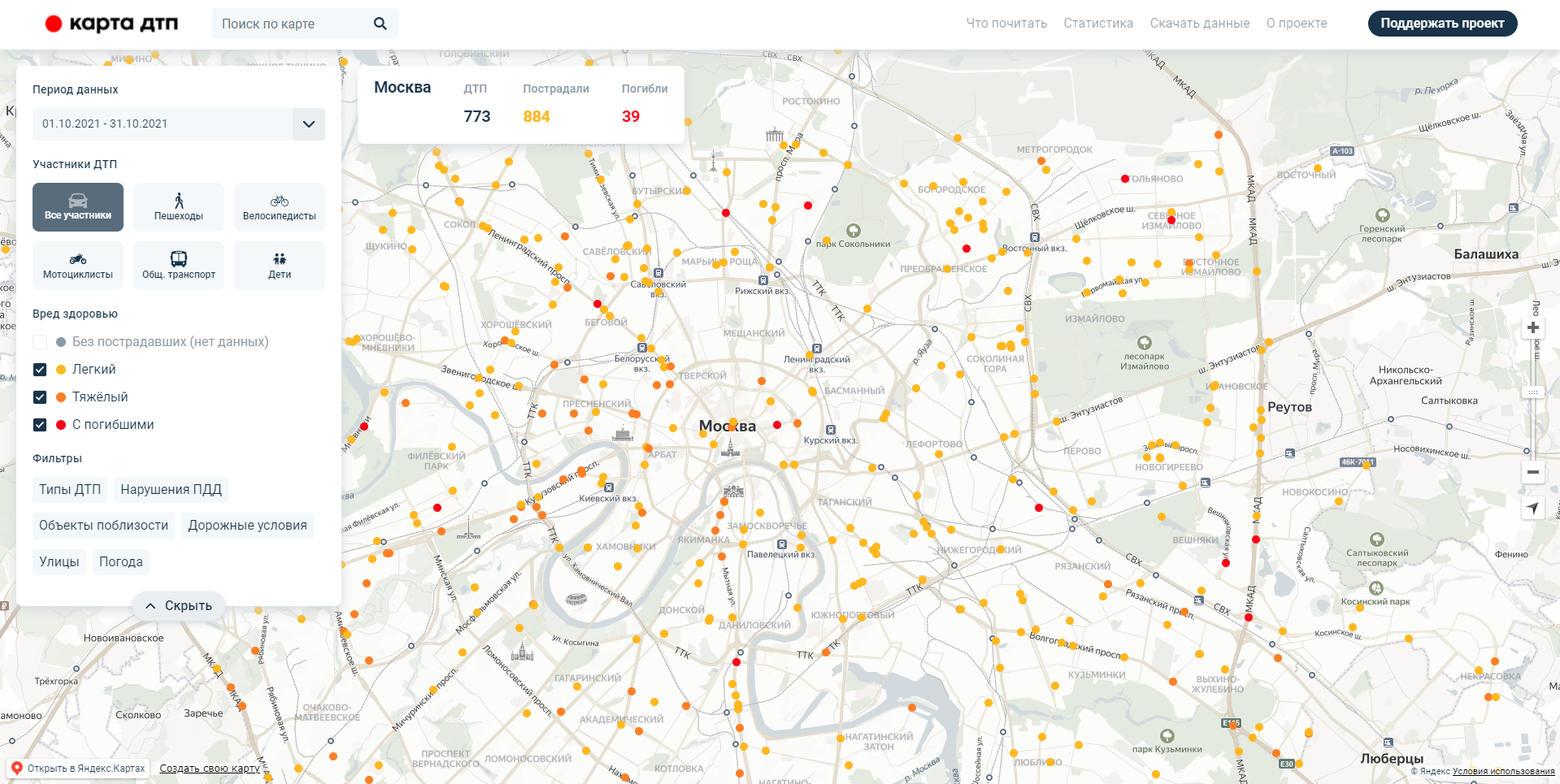
[Малый вес и системные требования 10](#_Toc88729460)

[Простота использования 11](#_Toc88729461)

# Модуль первый

Анализ и постановка задачи

В первой сессии нам предложили ознакомиться с заданием и исходными данными. Данные находились в сети интернет по адресу [dtp-stat.ru](https://dtp-stat.ru/)



(рис. 1 представленный сайт с данынми)

Во вкладке «[Скачать данные](https://dtp-stat.ru/opendata/)» находилось описание атрибутов, которые были в каждом датасете по области.



(рис. 2 описание данных)



(рис. 3 датасеты)

Была поставлена задача разработать программный продукт в виде интерактивной карты и чат бота с программным API

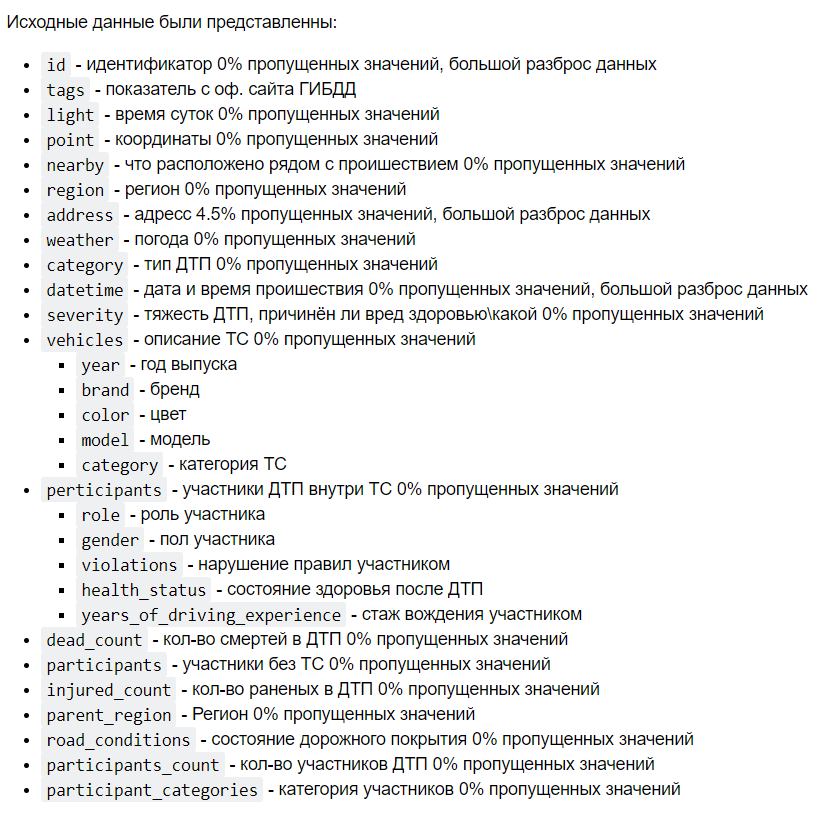
Загрузка и обработка данных

Из-за большого объёма данных (85 штук) было принято решение разработать цикл, который скачивал бы с сервера все имеющиеся датасеты с определёнными атрибутами. В этом мне помогла библиотека *BeautifulSoup и Request*, с помощью них я сделал запрос данных с сайта [dtp-stat.ru/opendata/](https://dtp-stat.ru/opendata/) на наличие кнопок «Скачать .geojson». Те, которые имели в своём атрибуте ещё и ссылку на сторонний репозиторий скачивали и формировали в два датасета – один с координатами ДТП, другой с общей информацией о них.



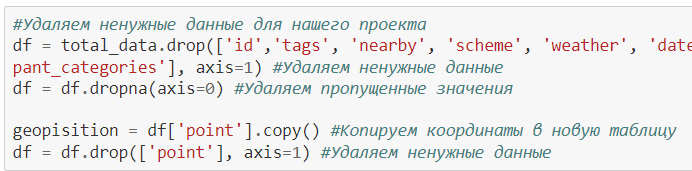
(рис. 4 цикл загрузки данных с сайта)

После проанализировав полученный датасет составил список пропущенных значений в данных.



(рис. 6 анализ данных)

Заодно было принято решение избавиться сначала от ненужных столбцов, может столбец с пропущенными значениями окажется нам не нужен и данные будут сохранены в первозданном варианте.



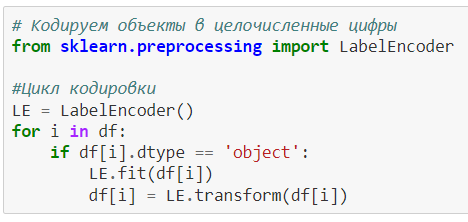
(рис. 7 предобработка данных)

1. Модуль второй

Окончательна обработка датасета

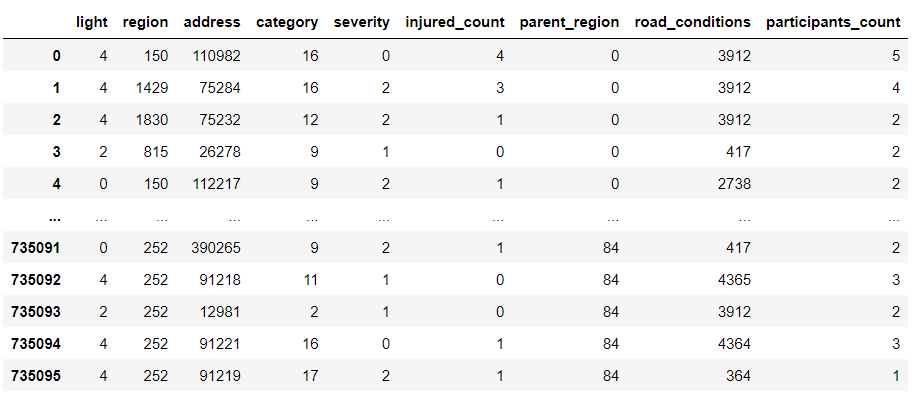
Для выполнения последующих заданий требуется обработать категориальные признаки в датасете, а именно столбцы: регион, адрес, категория ДТП, покрытие дороги, тип освещения»

Метод get dummies не подойдёт, потому что уникальных значений слишком много, поэтому было принято мной решение использовать метод кодирования категориальных признаков в числовые с помощью библиотеки *LabelEncoder*



(рис. 8 кодирование категориальных признаков)

После выполнения этого алгоритма получаем обработанный датасет без категориальных признаков



(рис. 9 обработанный датасет)

Формирование дополнительных атрибутов

Требовалось сформировать дополнительный атрибут, который показывал бы индекс срочности принятия решения для изменений дорожного участка. Индекс происшествий рассчитывается по формуле - X = (qfp)/r. Где X - Искомый индекс; q - кол-во происшествий; f - частота происхождения за год; p - средняя тяжесть (1 - легкое, 2 - среднее, 3 - тяжёлое); r - доля происшествий в регионе\городе;

Пример счёта индекса из города Братск, по адресу проезд Стройиндустрии, 0, где за последние 6 лет произошло 55 случаев ДТП

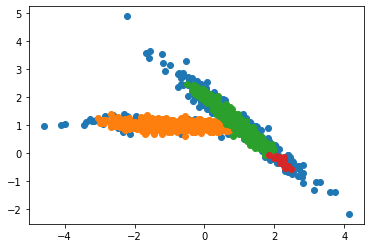
x = (55x7,8x2)/1242

x = 0.620 - индекс опасности участка и рекомендации к изменениям.

Если индекс ближе к 0, чем к 0.5, то участок не опасный и не требует особых изменений, если индекс в районе 0.5, то участок требует малых изменений, если индекс приближается к 1, то участок имеет серьёзные проблемы и требует срочных решений.

Кластеризация

Задача кластеризации – разбить данные по похожим признакам по заданному критерию. Я выбрал 3 метода кластеризации: Иерархическая кластеризация, Kmeans и DBSCAN. Из Трёх методов сработали лишь два. Иерархическая кластеризация выдала ошибку из-за нехватки ОЗУ. Kmeans выдал точность 80% и разделил на 2 кластера, что не подходит под нашу задачу, метод DBSCAN смог выделить 3 кластера и шумы в данных, поэтому будем использовать его



(рис. 8 кластеризация методом DBSCAN)

1. Модуль 3

Классификация и обучение модели

Задача классификации – некое множество объектов, разделённых на классы, для которых заранее известно в каких классах они находятся

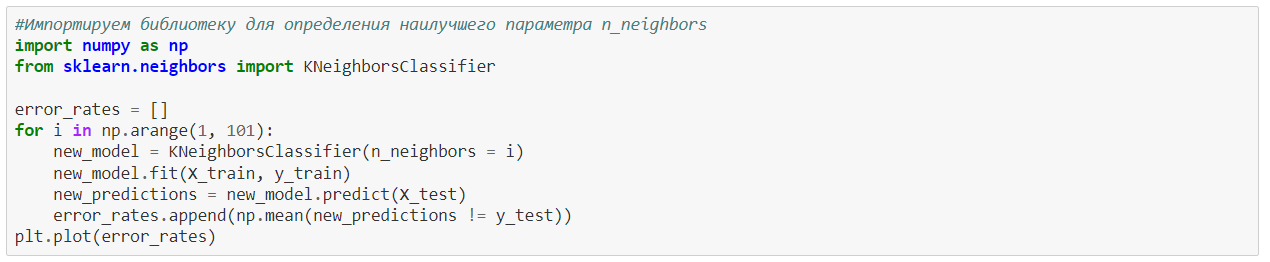
Было использовано 3 метода классификации: KNN, Случайный лес и Градиентный бустинг.

KNN показал точность 66% и 59%, случайны лес 100% и 65%, бустинг 66% и 66%, точность меня не устроила.

Feature Engineering

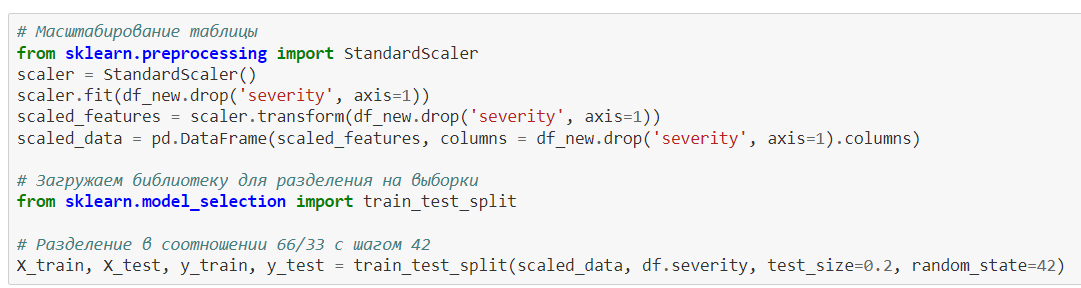
Поэтому переходим к улучшению модели путём подбора параметров

Для KNN цикл с определением наилучшего параметра от 1 до 100.



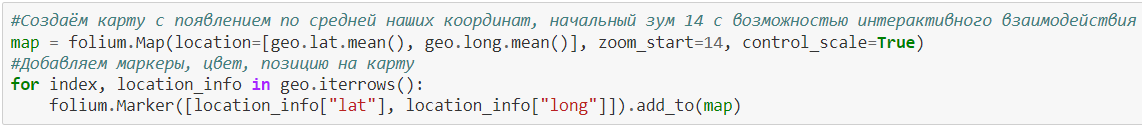
(рис. 9 реализация цикла KNN)

Цикл показал цифру 18, как самый оптимальный параметр с наименьшим числом ошибок и показал точность 86% и 91%, после дополнительной обработки данных путём скейлирования метод бустинга показал результат 88% и 86%

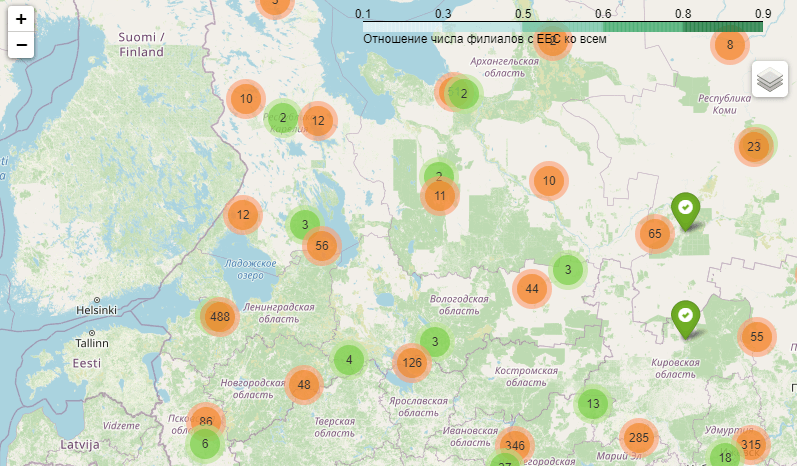


(рис. 10 скейлирование данных)

Разработка интерактивной карты

Используя заготовленные в первом модуле данные с геопозицией импортируем библиотеку *folium* и создаём карту с нашими точками

(рис. 11 код создания карты)



(рис. 12 карта)

Карта отображает участки, где были ДТП, в каком количестве и степень их тяжести.

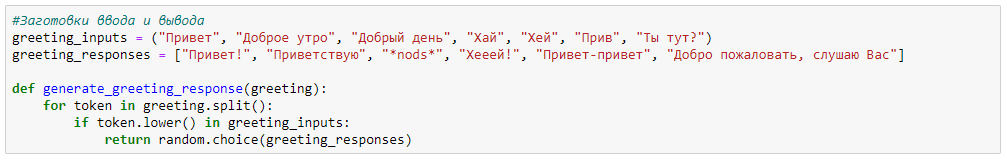
1. Модуль 4

Разработка чат бота

Задача чат бота выводить дорожные участки и перекрёстки, которые имеют повышенную опасность

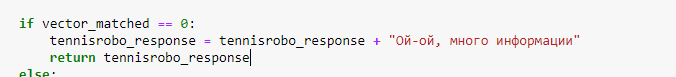
Редактирование чат бота

Для того чтобы редактировать чат бота нужно заменить команды приветствия



(рис.13 команды приветствия)

Команду вывода ошибки



(рис. 14 команда вывода ошибки в чат)

Редактирование диалога с пользователем



(рис. 15 редактирование диалога с пользователем)

Обучение

Чат бота можно дополнительно обучить, если это необходимо путём изменения ссылки исходных данных в формате HTML сайта.



(рис. 16. Обучение чатбота новым знаниям)

API

# Что можно делать с API?

# Загружать данные в Prombota

С помощью графического интерфейса пользователь может загружать данные в промбота в два клика и получать ответ в виде скачанного файла

# Как работает API

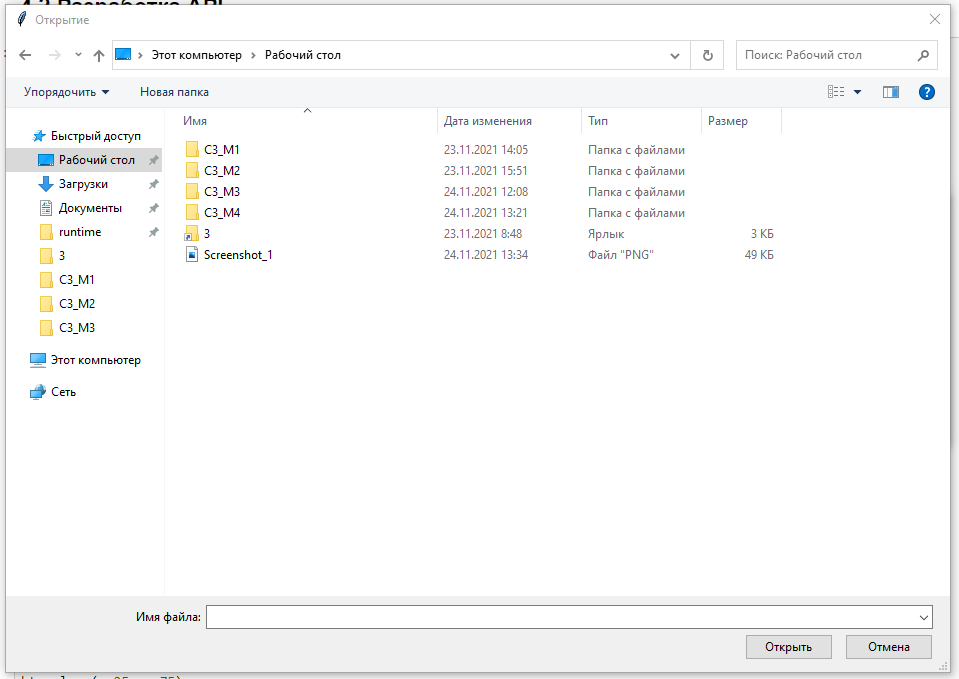
API анализирует файл и отправляет его на сервер обработки запроса, после попадает в память промбота, который повторно проверяет запрос и отправляет пользователю обработанный ответ

# Малый вес и системные требования

API можно использовать практически на любых системах Windows 10

# Простота использования

Всего одна кнопка, которая способна заменить горы работы



(рис. 17. Окно выбора файла)