Анализ данных (Data Science)

В данном проекте перед аналитиками данных стояло 3 глобальные задачи: классификация,сегментация изображений и внедрение модели в проект. В первую очередь был произведен анализ предоставленных данных, ознакомление с предметной областью задачи, вопросы заказчику, поиск информации которая могла бы помочь при дальнейшей работе.

Процесс работы был выстроен следующим образом: поиск дескрипторов для выделения ключевых признаков на изображениях, поиск метрик для оценки выбранных алгоритмов, предобработка данных, применение классических алгоритмов классификации, выбор главных признаков для улучшения показателей модели, преобразование изображений, выбор классических алгоритмов сегментации, выбор метрик для оценки, подбор параметров для лучшей сегментации, построение целостной модели, интеграция.

# 1. Исследовательский анализ данных (Exploratory data analysis)

Для выстраивания хода решения был произведен анализ предоставленных данных.

Количество изображений каждого из типов изображений можно увидеть на Таблице №1.

|  |  |  |
| --- | --- | --- |
|  | Количество изображений | Классификация |
| Фото в ультрафиолетовом свете (УФ) | 760 | Отсутствует, Насыщенное, Карбонатное |
| Фото в дневном свете (ДС) | 1300 | Аргиллит, Переслаивание пород, Алевролит глинистый, Песчаник, Глинисто-кремнистая порода, Песчаник глинистый, Уголь, Аргиллит углистый, Алевролит, Карбонатная порода, Известняк, Глина аргиллитоподобная |

Таблица №1. Распределение количества изображений и классов свечений/пород

В предоставленных json файлах более детально описываются изображения:

1. Свечение -> Насыщенное -> Интенсивность -> ['Слабое' 'Среднее' 'Сильное']
2. Свечение -> Насыщенное -> Характер -> ['Пятнистое' 'Полосчатое' 'Сплошное']
3. Свечение -> Карбонатное -> Тип -> ['Известковый' 'Известковистый']
4. Порода -> Цвет -> ['Зеленовато-серый' 'Серый' 'Темно-серый' 'Светло-серый' 'Бурый' 'Черный' 'Светло-серый с буроватым оттенком' 'Серо-коричневый' 'Черный с жирным блеском' 'Черный со стеклянным блеском']
5. Порода -> Целостность -> ['Разрушен' 'Не разрушен' 'Частично разрушен на плитки и обломки' 'Частично разрушен на крупные и средние куски.']
6. Порода -> Крепость -> ['Сцементированный' 'Слабосцементированный' 'Рыхлый']
7. Порода -> Трещиноватость -> ['Трещиноватый' 'Нет трещин']
8. Порода -> Форма трещин -> ['Ветвистая' 'Извилистая' 'Прямая' 'Ломаная' 'Сетчатая']
9. Порода -> Степень заполнения -> ['Открытая' 'Выполненная' 'Частично выполненная']
10. Порода -> Происхождение -> ['Техногенная' 'Естественная']
11. Порода -> Переслаивание пород -> Состав -> ['Песчаник & алевролит' 'Песчаник & алевролит & аргиллит' 'Алевролит & аргиллит' 'Кремнисто-глинистая & Глинисто-кремнистая & Кремнистая порода & алевролит' 'Песчаник & аргиллит' 'Уголь & аргиллит']
12. Порода -> Состав заполнения -> ['Минеральное']
13. Порода -> Ихнофоссилии (?)
14. Порода -> Текстура (?)

Для анализа данных были реализованы следующие функции:

* вывод изображения,
* получение изображение из данных,
* получение маска из данных,
* получение изображения по id,
* получение json файла с данными,описывающими изображение,
* вывод изображения и всех данных, связанные с ним,
* вывод 10 случайных изображений с одинаковыми сегментами,

Для исследования корреляций необходимо перевести признаки в категориальные следующим методом: categorical\_features\_transform

Диаграмма корреляции признаков представлена на рис.1. Из данной диаграммы можно увидеть на каких видах изображений какие классы могут находиться и как часто они встречаются между собой.

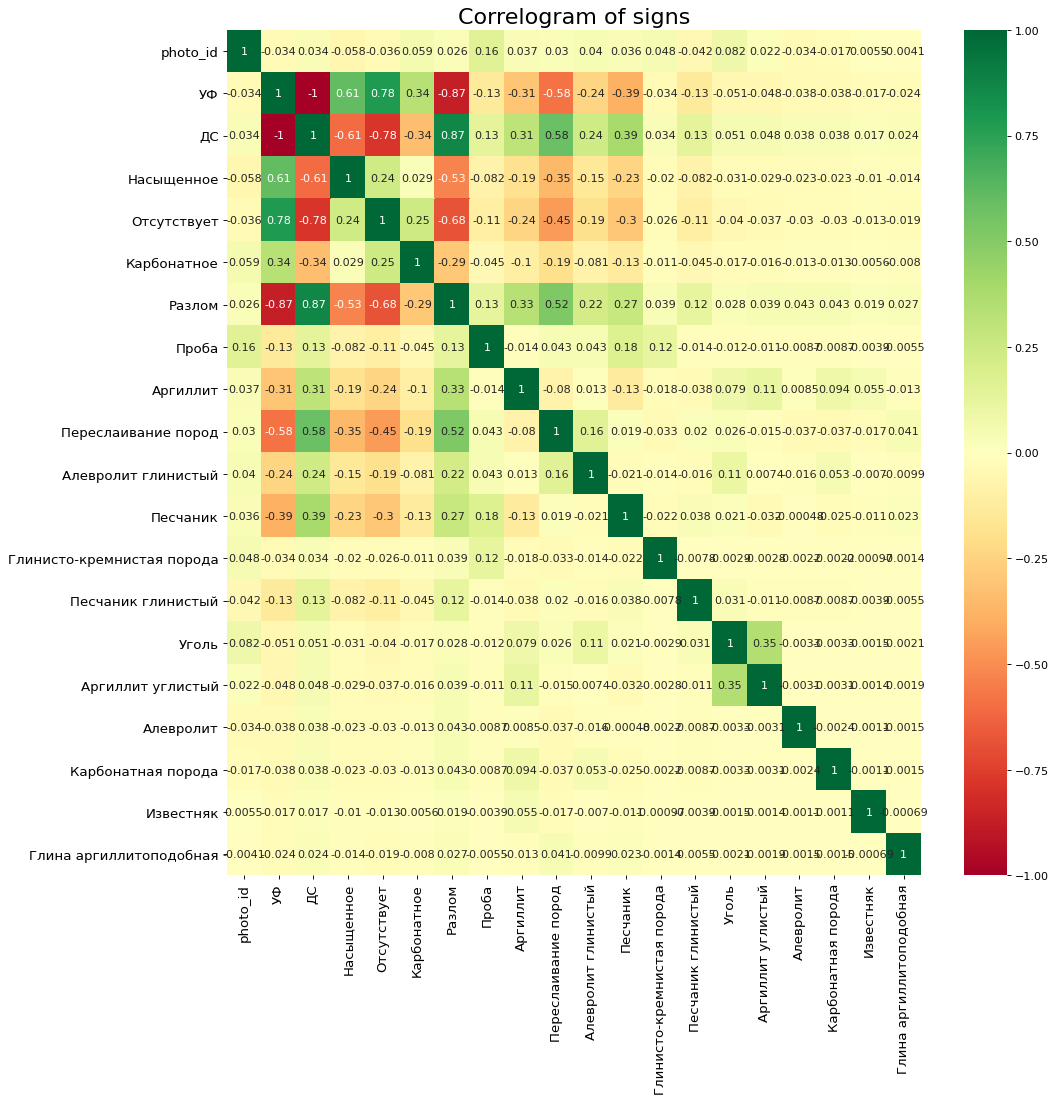


Рисунок №1. Матрица корреляций признаков на изображениях в ультрафиолетовом свете.

# 2. Изображения в ультрафиолетовом свете

На Диаграмме №2мы видим распределение подклассов свечения. Заметим, что количество случаев с карбонатным свечением значительно меньше. Было бы лучше если бы выборка была сбалансированной.

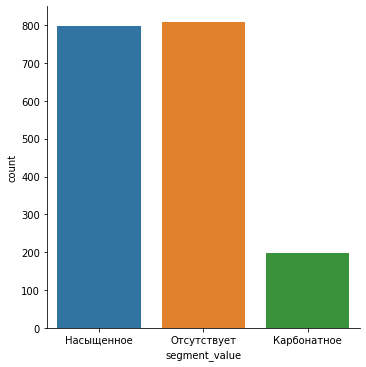


Рисунок №2. Распределение классов на разметке фотографий   
в ультрафиолетовом свете

При поиске дескрипторов были рассмотрены все дескрипторы,предоставленные библиотекой skimage. Были сделаны следующие выводы:

Blog дескрипторы не подходят так как они занимаются поиском каплей на изображениях.

Fast,Corner и подобные им дескрипторы не подходят так как они специализируются на поиске и описании углов.

Brief и ORB дескрипторы не подходят, т.к. специализируются на поиске особых точек, особых точек на изображениях в данном датасете нет, так как это целостные текстуры одного цвета.

Дескриптор GLCM подходит, так как используется в определении различий между сегментами и дальнейшем их разделении.

Дескриптор HOG и Canny также были выбраны, так как они специализируются на выделении границ.

Local binary pattern и multiblock\_lbp также могли быть полезными, так как специализируются на классификации целостных текстур.

### **2.1 Классификация**

После отбора дескрипторов была поставлена задача выделения признаков сегментов, преобразования сегментов в численные вектора и обучения алгоритма классификации.

В первую очередь была выдвинута гипотеза проверки классификации изображений в градациях серого.

В качестве преобразования изображения в численный вектор была выбрана гистограмма распределения.

Были выбраны следующие классические алгоритмы машинного обучения: SVM, RandomForest, ExtraTrees.

В качестве метрик были выбраны accuracy, f1-мера. После анализа объемов классов свечения было принято решение не опираться на accuracy, так как в датасете данные не были сбалансированными.

Результаты SVM, RandomForest при выборе гистограммы распределения для изображений в градациях серого можно увидеть в Таблице №1.

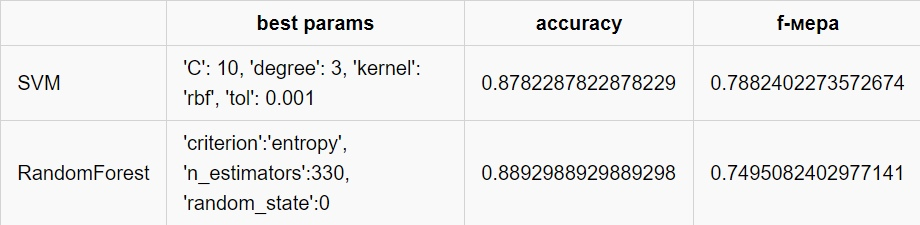


Таблица №1

Было принято отказаться от анализа изображений в градациях серого, так как было наблюдение, что при градациях серого классы перемешиваются и нет четкого разделения.

Далее было принято решение рассматривать трехканальные изображения.

Для анализа RGB цветовой модели изображений была реализована функция уравнивания гистограммы: histogram\_equalize(img:ndarray).

Пример применения функции можно увидеть на Рис. №2

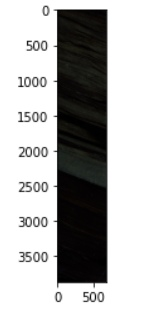
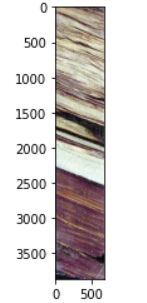
Исходное изображение: После уравнивания гистограммы  

Рис. №1 Рис. №2

При вычислении гистограммы распределения цветов использована группировка цветов пикселей до 64,128,32.При уравненной гистограмме получились результаты которые можно увидеть на Таблице №2

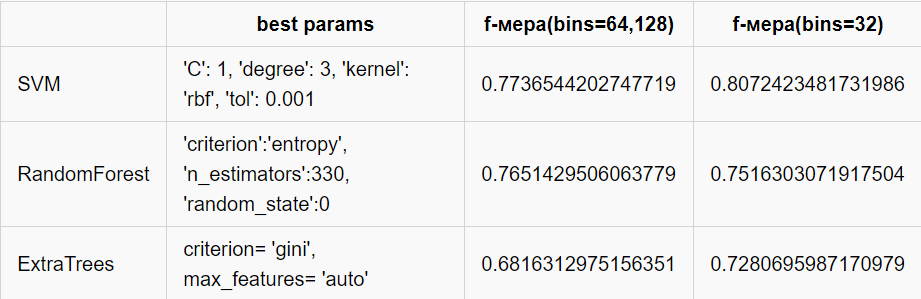


Таблица №2. Результаты работы алгоритмов классификации

Видно, что SVM показал результаты лучше, в то время как деревья показали результаты немного хуже.

Следующим решением было использование HSV цветовой модели изображения.Результаты можно увидеть в Таблице №3.

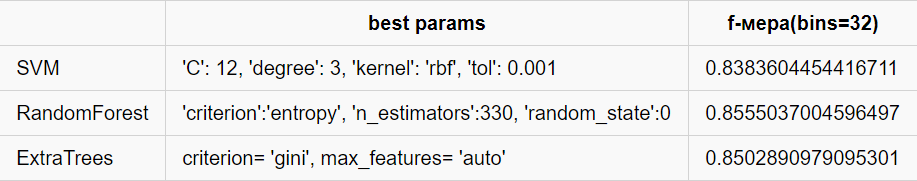


Таблица №3.Результаты работы алгоритмов классификации

Комбинирование HSV и RGB цветовых моделей дали наилучшие результаты на кроссвалидации при произведенном выборе признаков(были выбраны 50 наиболее важных признаков).Их можно посмотреть в Таблице №4.

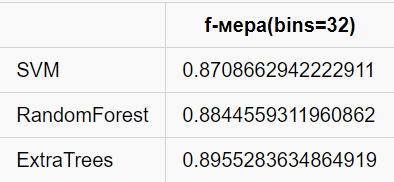


Таблица №4.Результаты работы алгоритмов классификации

Пример наложения фильтра EdgeDetection на HSV цветовую модель изображения можно увидеть на Рис. №5.

RGB модель: HSV модель: EdgeDetection:

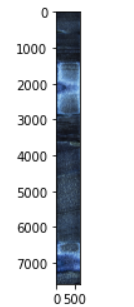
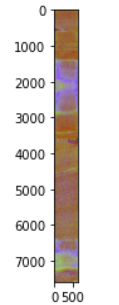
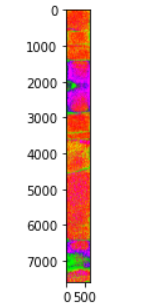
  

Рис. №3 Рис. №4 Рис. №5

В RGB и HSV цветовых моделях было произведено наложение следующих фильтров: uniform, gaussian, hessian, sobel, laplace и комбинировать их. Лучший результат показало сложение отдельно примененных к изображению фильтров uniform, gaussian, sobel, maximum\_filter.

Также были применены следующие свертки : Edge detection, Line detection, Gaussian blur, Simple box blur, The Sobel Edge, The laplacian Operator, The Laplacian of Gaussian (Свертки взяты с сайта: <https://aishack.in/tutorials/image-convolution-examples/>). Каких-то отличительных признаков не выявлено. Появилась идея не использовать какие-то фотографии как выбросы (например,темное изображение с 1 единственным свечением в небольшой точке), но результаты также не улучшились.Нелинейное преобразование такое как нахождение дисперсии (rgb\_photo - rgb\_photo\*\*2) не дало ничего интересного

Одновременное извлечение и конкатенация векторов гистограмм в HSV,RGB,grayscale изображениях не дали результаты выше, чем результат при тех же операциях в HSV,RGB (попытка в связи с тем, что в начале анализа изображений гистограммы изображений в градациях серого дали неплохие результаты) При суммировании изображений в HSV и RGB и извлечении векторов гистограмм также не было выявлено улучшений.

HOG, Canny,lbp дескрипторы не дали каких-то результатов.

**2.2 Сегментация изображений**

Были рассмотрены 4 алгоритма сегментации: threshold,Chan-Vese, Watershed, Slic.

При анализе алгоритмов была выявлена особенность датасета: типы свечения ‘Насыщенное’,’Карбонатное’ не могли встречаться на одном изображении. Следовательно есть возможность использования бинарных алгоритмов сегментации, но в таком случае на новых данных без этой особенности алгоритм будет показывать плохие результаты. Поэтому threshold и Chan-Vese не подходят.

1. Алгоритм Chan-Vese:

Предназначен для сегментации объектов без четко определенных границ поэтому на изображениях Керна он показал хороший результат. Результат можно увидеть на Рис.№6.

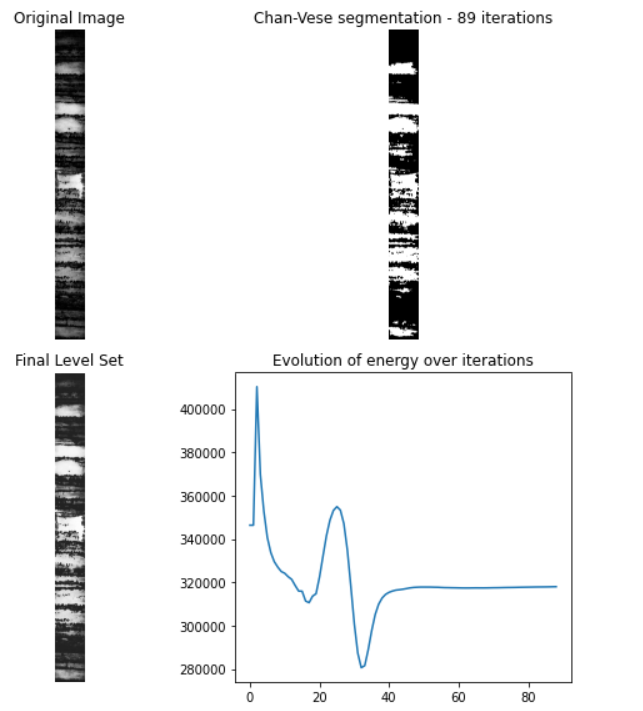


Рис. №6

Использовал следующие метрики для оценки сегментации: MSE,adapted\_rand\_error,variation\_of\_information

2. Watershed

Применение алгоритма без каких-либо преобразований изображения дало результаты,которые можно увидеть на Рис. №7.

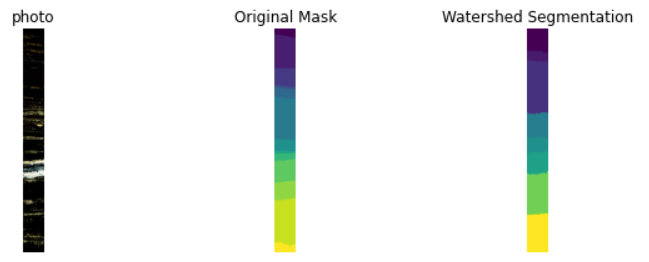


Рис. №7

При сложении изображений с наложенными на исходное изображение фильтрами гауса, лапласа и медианного границы стали лучше выделяться(markers = 35). (Рис. №8)

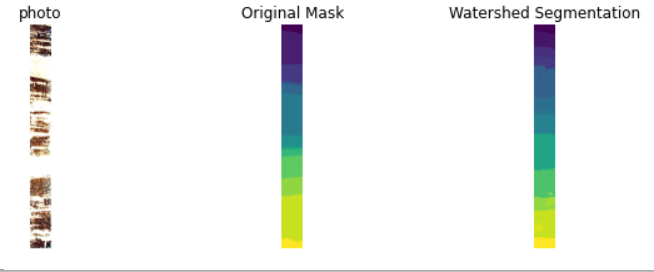


Рис. №8

3. Slic (алгоритм сегментации на основе K-means

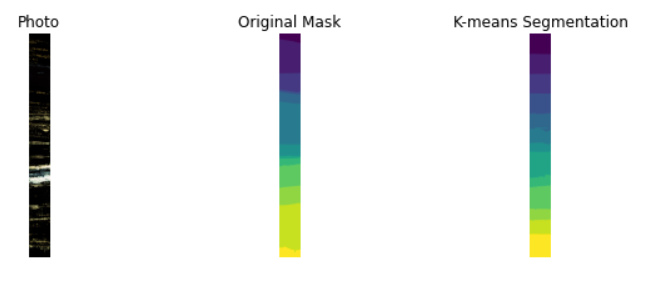


Рис. №9

Выбрана метрика оценки сегментации:

1. Сегментировать алгоритмом
2. Классифицировать выделенные сегменты
3. Произвести преобразование к семантической сегментации
4. Привести изначальную маску к семантической сегментации
5. Оценить любой метрикой(оценивал normalized\_root\_mse)

Данная метрика включает в себя оценку размера сегмента(это важно, иначе если алгоритм неправильной сегментирует и классифицирует большой объект, штраф будет такой же как если бы это был маленький объект, а это неправильно)

На 100 изображениях данная метрика показала следующие результаты:

1. watershed : 0.14481829715050887 (markers=12)
2. Slic : 0.22158974386459002 (n\_segments=23)

На 10 изображениях результаты при разных количествах классов можно увидеть в Таблице №10.

|  |  |  |
| --- | --- | --- |
|  | markers=12 | markers=20 |
| slic | 0.1769961577635625 | 0.22242965408445164 |
| watershed | 0.16757422167729205 | 0.16942944469761279 |

Тблица №4

Вывод: Watershed показывает результаты лучше и работает быстрее.

**2.3 Интеграция**

Была выбрана модульная интеграция.Класс модели представлен в Таблице №4.

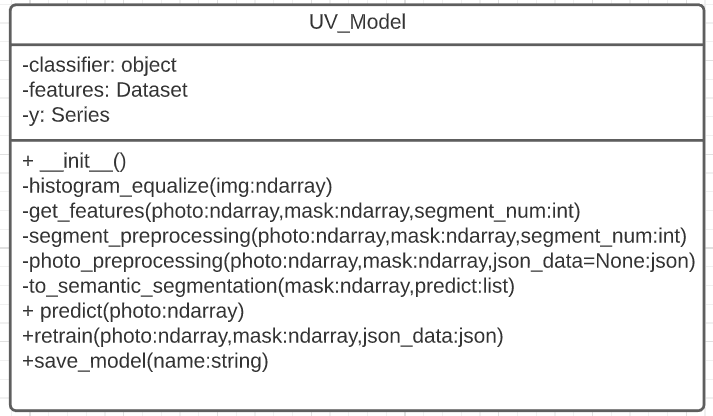


Таблица №5

Описание методов:

histogram\_equalize - выравнивание гистограммы

get\_features - преобразует сегмент изображения в вектор признаков

segment\_preprocessing - преобразует сегмент в hsv и rgv цветовых моделях и получает готовый для дальнейшего применения вектор

photo\_preprocessing - преобразует фотографию в набор векторов описывающих сегменты

to\_cemantic\_segmentation - приводит сегментацию к семантической

predict - предсказывает поданное изображение, возвращает семантическую сегментацию

retrain - дообучает модель

save\_model - сохраняет модель

В реализации retrain метода столкнулись с проблемой дообучения модели. Сначала для дообучения была использована следующая библиотека:

<https://github.com/garethjns/IncrementalTrees>. Дообучение стало проще, но точность значительно упала. На кросвалидации f1-мера 0.74. После этого была прочитана документация ExtraTreesClassifier и выявлено, что ExtraTrees имеет атрибут warm\_start, который дает порциальное обучение при значении True. Но при внедрении этого свойства при вызове методе predict после дообучения модели выдавало следующую ошибку:



Пока данная проблема не была решена, поэтому дообучение использует логику обучения модели с нуля на старых и новых данных.