

Результат всех работ

Модуль А

1. Сначала я выгрузил данные с сайта(habr) и с файлов которые нам предоставили ,
2. после мы разделили на нужные нам атрибуты(day, time, rate, views)
- 3.мы обработали весь текст и превели к одному виду

'минтранс предлагает новую версию законопроекта об организации перевозок пассажиров и багажа легковым такси в рф и о внесении и изменений в отдельные законодательные акты рф ведомство предлагает обязать службы заказа легкового такси предоставить автоматизированный удаленный доступ к своим информационным системам и данным федеральной службе безопасности под это требование должны по пасть данные используемые для получения хранения обработки и передачи заказов легкового такси кроме того от агрегаторов потребуют передавать сведения о перевозках в том числе о местоположении такси и водителя в региональную навигационно информационную систему рнис платформа по управлению и мониторингу транспортной инфраструктуры в российских регионах законопроект уже согласовали министры транспорта труда финансов экономического развития промышленности и торговли фнс и минздрав в минюсте подтвердили что документ находился на рассмотрении и разработке в министерстве по развитию транспортной инфраструктуры с 2017 года и утвержденной директивой была разработана

Модуль Б

Мы делали следующие

1 Поиск ключевых слов/n-грамм. Векторизация текстов

2 для второго задания я воспользовался моделью

Взял я CountVectorizer

Это будет выглядеть примерно так

[illegible]

Модель В

3.1.3 обучение моделей.

Для обучения модели я взял

RandomForestClassifier

Алгоритм также основанный на деревьях решений, но вместо последовательного обучения обучает множество деревьев на разных частях датасета и усредняет значения. Подходит для достаточно больших датасетов, однако HistGradientBoostingClassifier зачастую справляется быстрее и с большей точностью

3.1.3 оценку моделей

Для оценки я буду использовать метрику:

f1_score -Вычислите оценку F1, также известную как сбалансированная F-оценка или F-мера. он прост в использование а так же F1 пытается достигнуть своего наилучшего значения после каждой проверенной модели, можно посмотреть какая модель показала наилучший показатель.

3.2 Оптимизация модели.

3.2.1 выбор значимых признаков.

Для этой части я буду использовать две библиотеки.

SelectKBest / chi2

Одномерный отбор признаков работает путем выбора лучших признаков на основе одномерных статистических тестов. Это можно рассматривать как шаг предварительной обработки к оценщику.

я её выбрал так как она удаляет всё, кроме самых результативные функции.

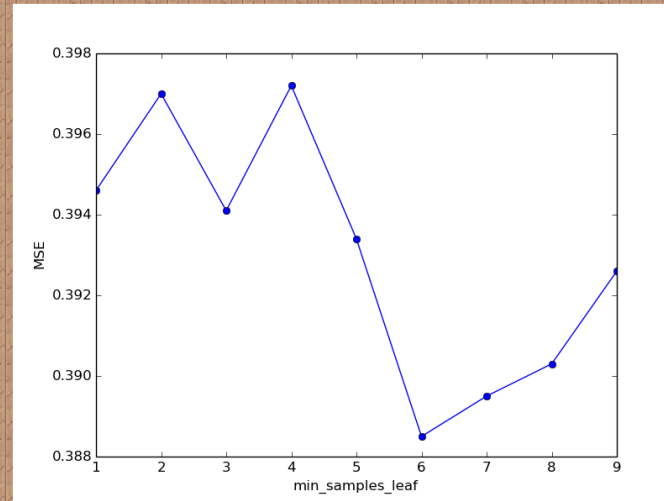
3.2.2 понижение размерности

для понижения размерности я буду использовать

PCA - PCA используется для разложения многомерного набора данных на набор последовательных ортогональных компонентов, которые объясняют максимальную величину дисперсии. это то что нам нужно для наших данных.

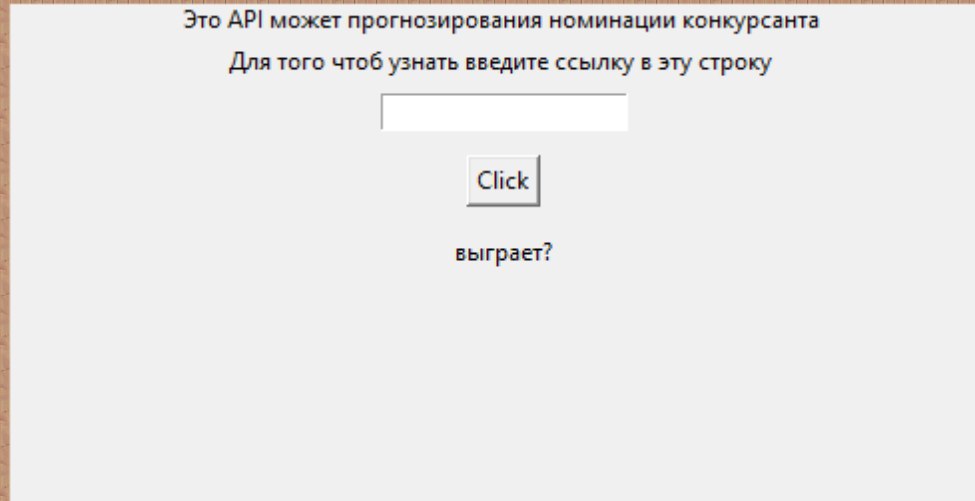
3.2.3 настраивая гиперпараметры

Вот так будет выглядит результат



4.2 Разработка прикладного решения.

Дальше мы делаем арі которое будет прогнозировать номинации конкурсанта, и так же добавил окно и немного текста



Это API может прогнозирование номинации конкурсанта

Для того чтоб узнать введите ссылку в эту строку

Click

выиграет?

Для создание окна я решил использовать tkinter, так как он прост в использование.



Библиотеки которые я использовал для всех работ

```
import pandas as pd
import requests
import re
from tkinter import *
import pymorphy2
from bs4 import BeautifulSoup as bs
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.datasets import load_iris
from sklearn.linear_model import LogisticRegression
from sklearn.datasets import load_iris
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.decomposition import PCA
```