ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по курсу
«Data Science»

Докладчик
Царева Наталья Викторовна

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Постановка задачи

Тема выпускной квалификационной работы – прогнозирование конечных свойств новых материалов (композиционных материалов).

На входе имеются данные о начальных свойствах компонентов композиционных материалов (количество связующего, наполнителя, температурный режим отверждения и т.д.), представленные в виде 2х таблиц, которые нужно соединить.

ПЛАН РАБОТЫ:

- Необходимо провести разведочный анализ полученных на входе данных с помощью изученных в курсе методов.
- Данные необходимо предварительно обработать, продемонстрировать этапы обработки и выбранные методы.
- На выходе необходимо обучить модели машинного обучения и спрогнозировать значения **Модуль упругости при растяжении и Прочность при растяжении,** а также написать нейронную сеть, которая будет рекомендовать **Соотношение матрица-наполнитель**. Полученные модели необходимо оценить.
- Также нужно разработать приложение с графическим интерфейсом или интерфейсом командной строки, которое будет выдавать прогноз на основе обученных моделей.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Описание используемых методов

Выполнено на языке Python 3.8.
Anaconda Jupiter Notebook (основной код)
Visual Studio Code (разработка приложения)

Библиотеки

- Numpy - для работы с многомерными массивами
- Pandas - для обработки датафреймов
- Matplotlib - для визуализации
- Seaborn - для визуализации
- Xgboost - для построения особой регрессионой модели – деревья решений с градиентным усилением
- Sklearn - для разработки моделей машинного обучения
- Pickle - для сохранения полученных моделей
- Tensorflow.keras - для разработки нейронной сети

Так как поставленная задача сводится к предсказанию некого значения, т.е. к задаче регрессии, будут проведен анализ следующих регрессионных моделей МО:

```
regressors = [
    KNeighborsRegressor(),
    GradientBoostingRegressor(),
    ExtraTreesRegressor(),
    RandomForestRegressor(),
    DecisionTreeRegressor(),
    LinearRegression(),
    Lasso(),
    Ridge(),
    XGBRegressor(),
    MLPRegressor()
]
```

Оценка моделей: коэффициент детерминации (R^2) и средняя абсолютная ошибка (MAE)

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

Цель разведочного анализа – получение первоначальных представлений о характерах распределений переменных исходного набора данных, формирование оценки качества исходных данных (наличие пропусков, выбросов), выявление характера взаимосвязи между переменными с целью последующего выдвижения гипотез о наиболее подходящих для решения задачи машинного обучения.

В данном разделе были использованы методы, реализованные в библиотеке pandas, которые позволяют провести разведочный анализ данных, а именно:

.info() общая информация по датасету;

.shape структура датасета;

.describe() описательная статистика датасета;

.isnull() выявление пропусков;

.dublicated() выявление дублей в строках;

.corr() проверка корреляции.

Библиотеки seaborn и matplotlib - для визуализации данных, построения различных графиков.

Так же с помощью метода SelectKBest из библиотеки Sklearn была проведена попытка ранжирования признаков по важности для каждой целевой переменной.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных Результаты

На входе имеется два файла с наборами данных X_bp.xlsx, X_nup.xlsx, число строк в них различное, 1023 и 1040 соответственно. По условию задачи объединим данные файлы в один датасет по индексу с типом объединения INNER, при этом остаются строки, присутствующие только в обоих датасетах.

Результат: датасет 1023 элемента, 13 параметров.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1023 entries, 0 to 1022
Data columns (total 13 columns):
 #   Column                             Non-Null Count  Dtype
---  ------                             --------------  -----
 0   Соотношение матрица-наполнитель    1023 non-null   float64
 1   Плотность, кг/м3                   1023 non-null   float64
 2   модуль упругости, ГПа              1023 non-null   float64
 3   Количество отвердителя, м.%        1023 non-null   float64
 4   Содержание эпоксидных групп,%_2    1023 non-null   float64
 5   Температура вспышки, С_2           1023 non-null   float64
 6   Поверхностная плотность, г/м2      1023 non-null   float64
 7   Модуль упругости при растяжении, ГПа 1023 non-null   float64
 8   Прочность при растяжении, МПа      1023 non-null   float64
 9   Потребление смолы, г/м2            1023 non-null   float64
 10  Угол нашивки, град                 1023 non-null   int64
 11  Шаг нашивки                        1023 non-null   float64
 12  Плотность нашивки                  1023 non-null   float64
dtypes: float64(12), int64(1)
```

Дубликаты, пропуски отсутствуют

```
df['Угол нашивки, град'].unique() #выявлена категориальная переменная

array([ 0, 90], dtype=int64)
```

```
df.describe().T#общая описательная статистика
```

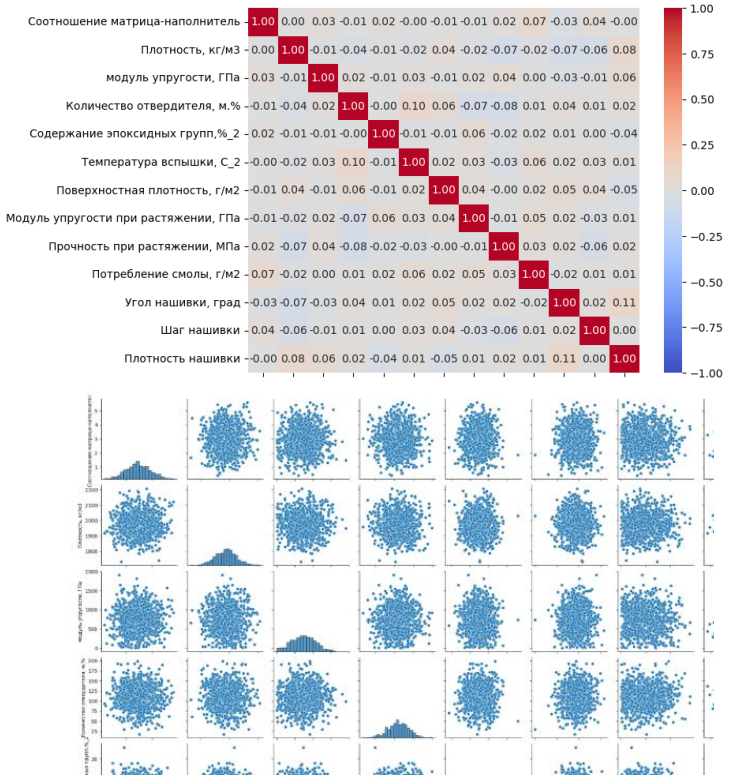
	count	mean	std	min	25%	50%	75%	max
Соотношение матрица-наполнитель	1023.0	2.930366	0.913222	0.389403	2.317887	2.906878	3.552660	5.591742
Плотность, кг/м3	1023.0	1975.734888	73.729231	1731.764635	1924.155467	1977.621657	2021.374375	2207.773481
модуль упругости, ГПа	1023.0	739.923233	330.231581	2.436909	500.047452	739.664328	961.812526	1911.536477
Количество отвердителя, м.%	1023.0	110.570769	28.295911	17.740275	92.443497	110.564840	129.730366	198.953207
Содержание эпоксидных групп,%_2	1023.0	22.244390	2.406301	14.254985	20.608034	22.230744	23.961934	33.000000
Температура вспышки, С_2	1023.0	285.882151	40.943260	100.000000	259.066528	285.896812	313.002106	413.273418
Поверхностная плотность, г/м2	1023.0	482.731833	281.314690	0.603740	266.816645	451.864365	693.225017	1399.542362
Модуль упругости при растяжении, ГПа	1023.0	73.328571	3.118983	64.054061	71.245018	73.268805	75.356612	82.682051
Прочность при растяжении, МПа	1023.0	2466.922843	485.628006	1036.856605	2135.850448	2459.524526	2767.193119	3848.436732
Потребление смолы, г/м2	1023.0	218.423144	59.735931	33.803026	179.627520	219.198882	257.481724	414.590628
Угол нашивки, град	1023.0	44.252199	45.015793	0.000000	0.000000	0.000000	90.000000	90.000000
Шаг нашивки	1023.0	6.899222	2.563467	0.000000	5.080033	6.916144	8.586293	14.440522
Плотность нашивки	1023.0	57.153929	12.350969	0.000000	49.799212	57.341920	64.944961	103.988901

Описательная статистика

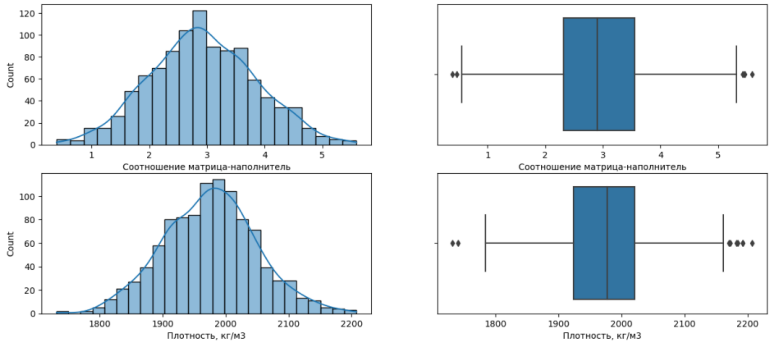
ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных Результаты

Оценка корреляции между параметрами на тепловой карте и попарными графиками



Визуализация распределения параметров – предварительно признаки распределены нормально, есть выбросы



Скоринг важности признаков методом SelectKBest

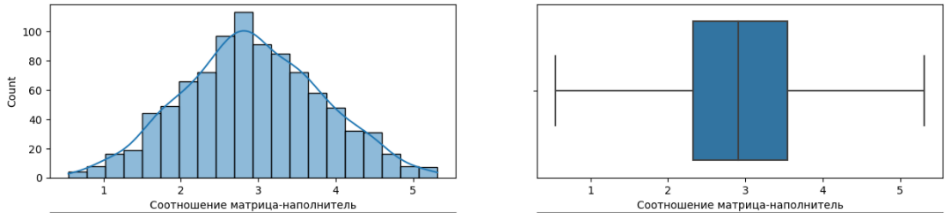
Важность признаков для Модуль упругости при растяжении, ГПа:

	Specs	Score
3	Количество отвердителя, м.%	4.457238
4	Содержание эпоксидных групп,%_2	3.307961
8	Потребление смолы, г/м2	2.656047
6	Поверхностная плотность, г/м2	1.377153
10	Шаг нашивки	0.887374
5	Температура вспышки, С_2	0.824993
2	модуль упругости, ГПа	0.553027
9	Угол нашивки, град	0.540559
1	Плотность, кг/м3	0.316430
7	Прочность при растяжении, МПа	0.082870
0	Соотношение матрица-наполнитель	0.072228
11	Плотность нашивки	0.042818

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Предобработка данных

Обработка выбросов: первоначально была проведена методом 3х сигм, что даже при повторении не дало удовлетворительный результат. Финально удалены методом межквартильных интервалов.



Кодировка категориальной переменной Угол нашивки

```
encoder=LabelEncoder()
df_pred1['Угол нашивки, град'] =encoder.fit_transform(df_pred1['Угол нашивки, град'])

df_pred1['Угол нашивки, град']
```

Нормализация MimMaxScaler: датасет df_norm
Стандартизация StandartScaler: датасет df_std

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Обучение и тестирование модели для Модуль упругости при растяжении, гПа

Разбиение на тренировочную и тестовую выборки для подачи в модели машинного обучения методом train_test_split библиотеки sklearn.
В целях оптимизации работы был собрана функция и список регрессионных моделей regressors.

Модель	R2 score, df_norm	R2 score, df_std
KNeighborsRegressor	-0.247	-0.305
GradientBoostingRegressor	-0.075	-0.076
ExtraTreesRegressor	-0.058	-0.038
RandomForestRegressor	-0.031	-0.021
DecisionTreeRegressor	-1.084	-1.111
LinearRegression	0.006	0.006
Lasso	-0.001	-0.001
Ridge	0.006	0.006
XGBRegressor	-0.331	-0.228
MLPRegressor	-0.016	-0.171

Оценка моделей с подобранными гиперпараметрами с помощью метода GridSearch

Model	Explained variance	MAE	R2 score
LinearRegression	0.007377	0.146740	0.006
RidgeRegression	0.005644	0.146948	0.004
LassoRegression	0.000000	0.147274	-0.001
RandomForestRegressor	-0.008845	0.147400	-0.010

```
#сохраняем модель Линейной Регрессии для Модуля упругости при растяжении
#Модуль упругости при растяжении,гПа    LinearRegression    0.007377    0.146740    0.006

with open("lr1_model.pkl", "wb") as f:
    pickle.dump(lr1, f)
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Обучение и тестирование модели для Прочность при растяжении, мПа

Разбиение на тренировочную и тестовую выборки для подачи в модели машинного обучения методом train_test_split библиотеки sklearn.

В целях оптимизации работы был собрана функция и список регрессионных моделей regressors.

Модель	R2 score, df_norm
KNeighborsRegressor()	-0.190
GradientBoostingRegressor()	-0.087
ExtraTreesRegressor()	-0.048
RandomForestRegressor()	-0.007
DecisionTreeRegressor()	-0.937
LinearRegression()	0.001
Lasso()	-0.006
Ridge()	0.002
XGBRegressor()	-0.208
MLPRegressor()	-0.193

Оценка моделей с подобранными гиперпараметрами с помощью метода GridSearch

Model	Explained variance	MAE	R2 score
RidgeRegression	0.007792	0.152404	0.002
LinearRegression	0.007064	0.152871	0.001
MLPRegressor	0.000178	0.152418	-0.004
LassoRegression	0.000000	0.152556	-0.006
RandomForestRegressor	-0.030630	0.154431	-0.035

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разработка нейронной сети для параметра Соотношение матрица-наполнитель

Разбиение на тренировочную и тестовую выборки для подачи в модели машинного обучения методом train_test_split библиотеки sklearn. Для оценки качества можеди та кже был выделен валидационный набор в размере 0.2 в процессе сборки моделей.

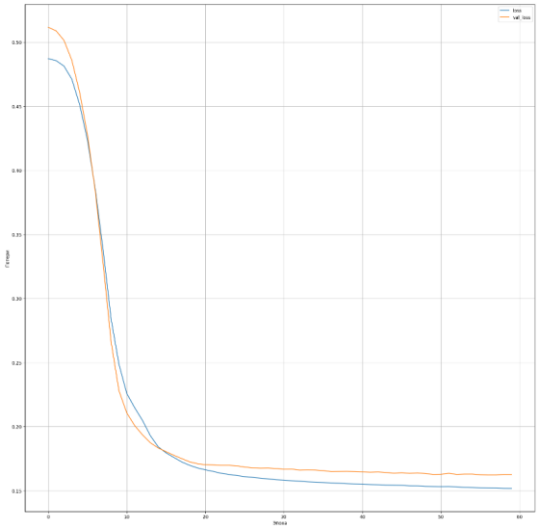
1ая модель Оценка 0.155123591423034 67

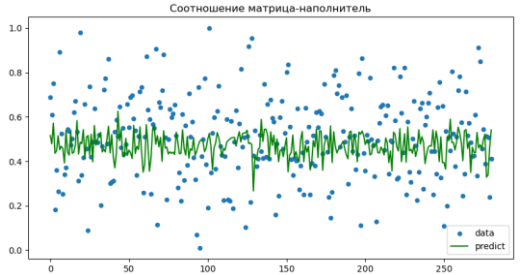
```
model_X3.summary()
```

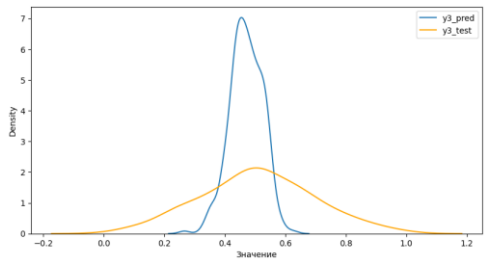
```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 8)                 104

 dense_1 (Dense)             (None, 8)                 72

 dense_2 (Dense)             (None, 1)                 9

=================================================================
Total params: 185
Trainable params: 185
Non-trainable params: 0
_________________________________________________________________
```

```
#собираем модель
model_X3.compile(loss="mae", optimizer="adam")
```

```
#обучаем модель
history = model_X3.fit(X3_train, y3_train, batch_size=100, epochs=60, validation_split=0.3)
```







ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разработка нейронной сети для параметра Соотношение матрица-наполнитель

2ая модель Оценка 0,1580493003129959

```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_3 (Dense)             (None, 8)                 104

 dense_4 (Dense)             (None, 8)                 72

 dense_5 (Dense)             (None, 1)                 9

=================================================================
Total params: 185
Trainable params: 185
Non-trainable params: 0
_________________________________________________________________
```

Зая модель Оценка 0.1562628448009491

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense_6 (Dense)             (None, 12)                156

 dropout (Dropout)           (None, 12)                0

 dense_7 (Dense)             (None, 8)                 104

 dropout_1 (Dropout)         (None, 8)                 0

 dense_8 (Dense)             (None, 4)                 36

 dropout_2 (Dropout)         (None, 4)                 0

 dense_9 (Dense)             (None, 1)                 5

=================================================================
Total params: 301
Trainable params: 301
Non-trainable params: 0
_________________________________________________________________
```

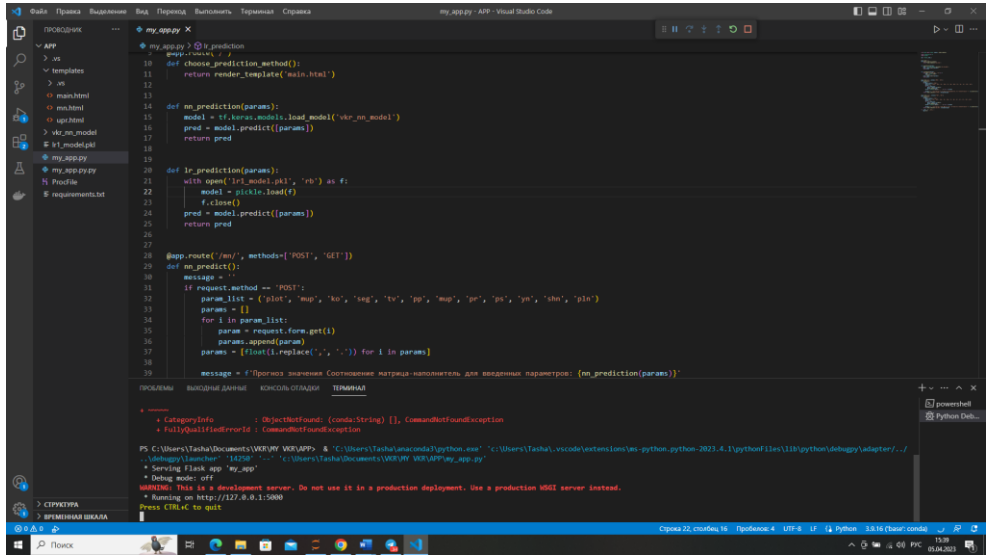
```
#сохраняем модель - выбрана нейронная сеть для Соотношения матрица-наполнительб первый вариант

model_X3.save('vkr_nn_model')
```

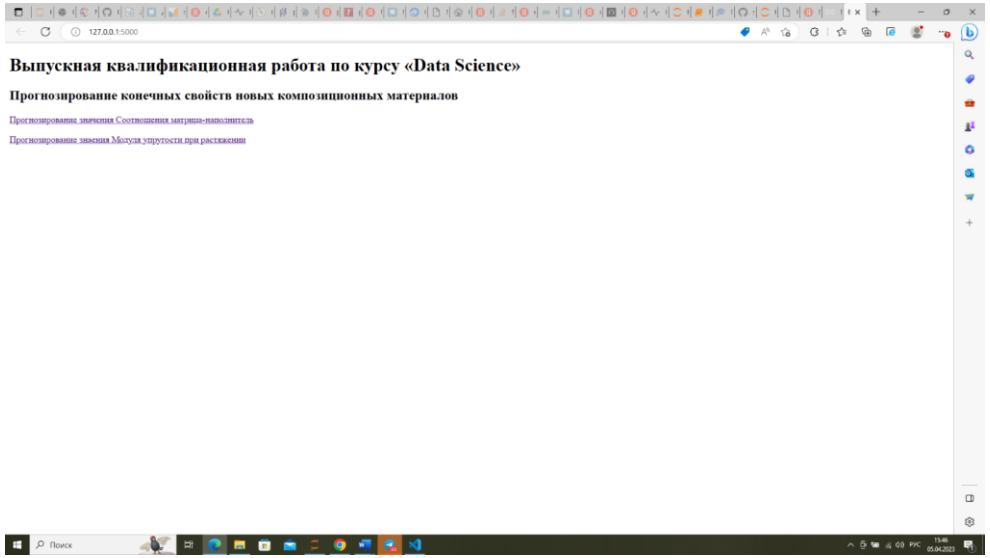
```
WARNING:absl:Found untraced functions such as _update_step_xla while saving (showing 1 of 1). These functions will not be d
tly callable after loading.

INFO:tensorflow:Assets written to: vkr_nn_model\assets
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разработка приложения

В ходе выполнения работы разработано приложение для предсказания параметра прочности при растяжении. Приложение разработано в среде Visual Studio Code с помощью библиотеки FLASK.





Все файлы с кодом, приложением, пояснительной запиской выложены в репозитарий по адресу https://github.com/TashaGit/Final_qualification_paper

ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

do.bmstu.ru