ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
по курсу «Data Science»

Прогнозирование конечных свойств композиционных материалов

Докладчик: Харин Сергей Семенович

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Этапы работы

1. **Разведочный анализ данных**
2. **Предобработка данных**
3. **Создание и обучение моделей**
4. **Создание и обучение нейросетей**
5. **Разработка приложения**

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

Обьединение датасетов по индексу, тип объединения INNER

```
[3] # обьединяем датасеты
    df = pd.merge(X_bp, X_nup, how = 'inner')
    df.drop(['Unnamed: 0'], axis=1, inplace=True)
    # 13 столбцов и 1023 строки
    print('Размер датасета: {}'.format(df.shape))
    df.head()
```

Размер датасета: (1023, 13)

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
0	1.857143	2030.0	738.736842	30.00	22.267857	100.000000	210.0	70.0	3000.0	220.0	0	4.0	57.0
1	1.857143	2030.0	738.736842	50.00	23.750000	284.615385	210.0	70.0	3000.0	220.0	0	4.0	60.0
2	1.857143	2030.0	738.736842	49.90	33.000000	284.615385	210.0	70.0	3000.0	220.0	0	4.0	70.0
3	1.857143	2030.0	738.736842	129.00	21.250000	300.000000	210.0	70.0	3000.0	220.0	0	5.0	47.0
4	2.771331	2030.0	753.000000	111.86	22.267857	284.615385	210.0	70.0	3000.0	220.0	0	5.0	57.0

Просмотр числовых статистик датасета

```
[9] df.describe()
```

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
count	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000	1023.000000
mean	2.930366	1975.734888	739.923233	110.570769	22.244390	285.882151	482.731833	73.328571	2466.922843	218.423144	44.252199	6.899222	57.153929
std	0.913222	73.729231	330.231581	28.295911	2.406301	40.943260	281.314690	3.118983	485.628006	59.735931	45.015793	2.563467	12.350969
min	0.389403	1731.764635	2.436909	17.740275	14.254985	100.000000	0.603740	64.054061	1036.856605	33.803026	0.000000	0.000000	0.000000
25%	2.317887	1924.155467	500.047452	92.443497	20.608034	259.066528	266.816645	71.245018	2135.850448	179.627520	0.000000	5.080033	49.799212
50%	2.906878	1977.621657	739.664328	110.564840	22.230744	285.896812	451.864365	73.268805	2459.524526	219.198882	0.000000	6.916144	57.341920
75%	3.552660	2021.374375	961.812526	129.730366	23.961934	313.002106	693.225017	75.356612	2767.193119	257.481724	90.000000	8.586293	64.944961
max	5.591742	2207.773481	1911.536477	198.953207	33.000000	413.273418	1399.542362	82.682051	3848.436732	414.590628	90.000000	14.440522	103.988901

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

Анализ и предобработка датасета

```
df.info()
```

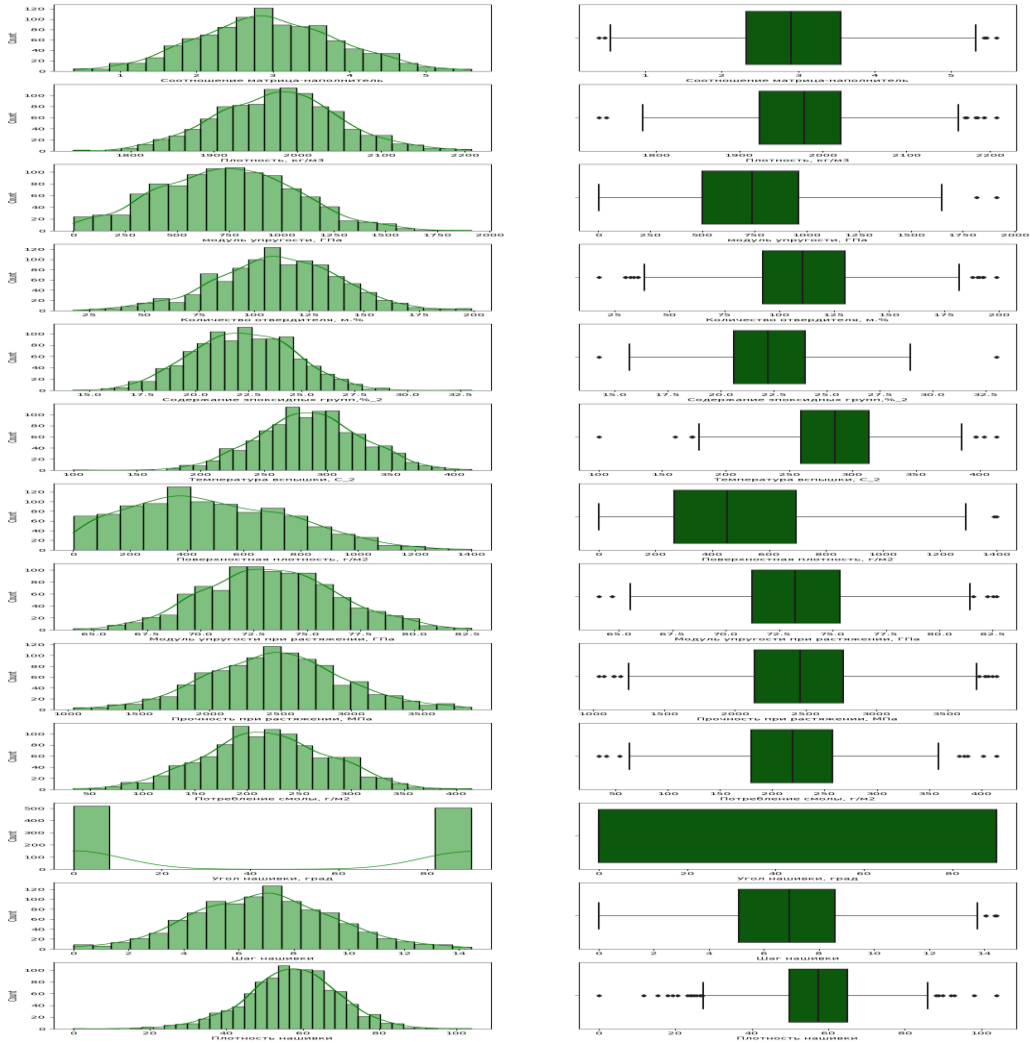
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1023 entries, 0 to 1022
Data columns (total 13 columns):
 #   Column                             Non-Null Count  Dtype  
---  ------                             --------------  -----  
 0   Соотношение матрица-наполнитель    1023 non-null   float64
 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)
memory usage: 111.9 KB
```

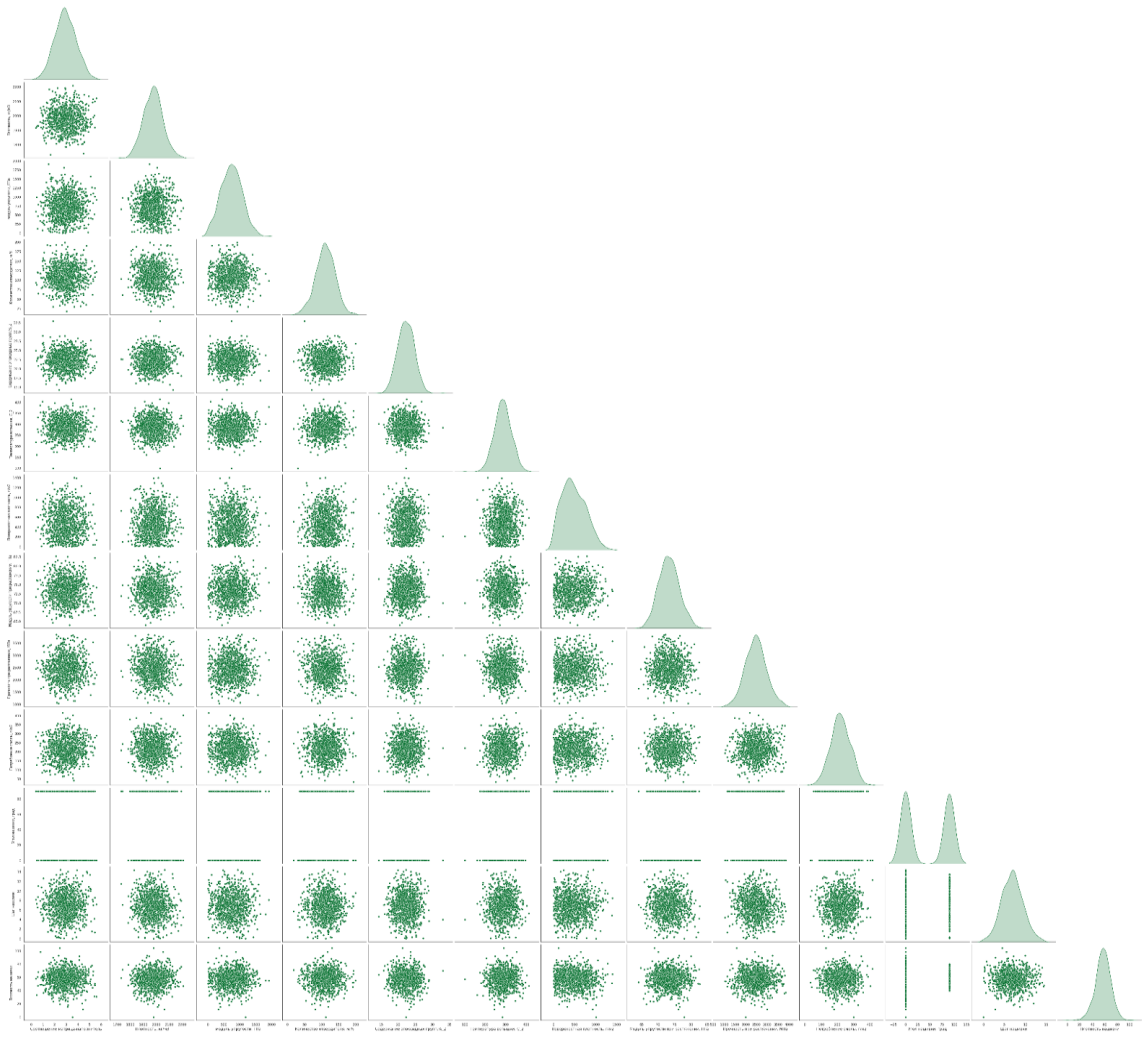
```
[ ]  df.describe().T
```

	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

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

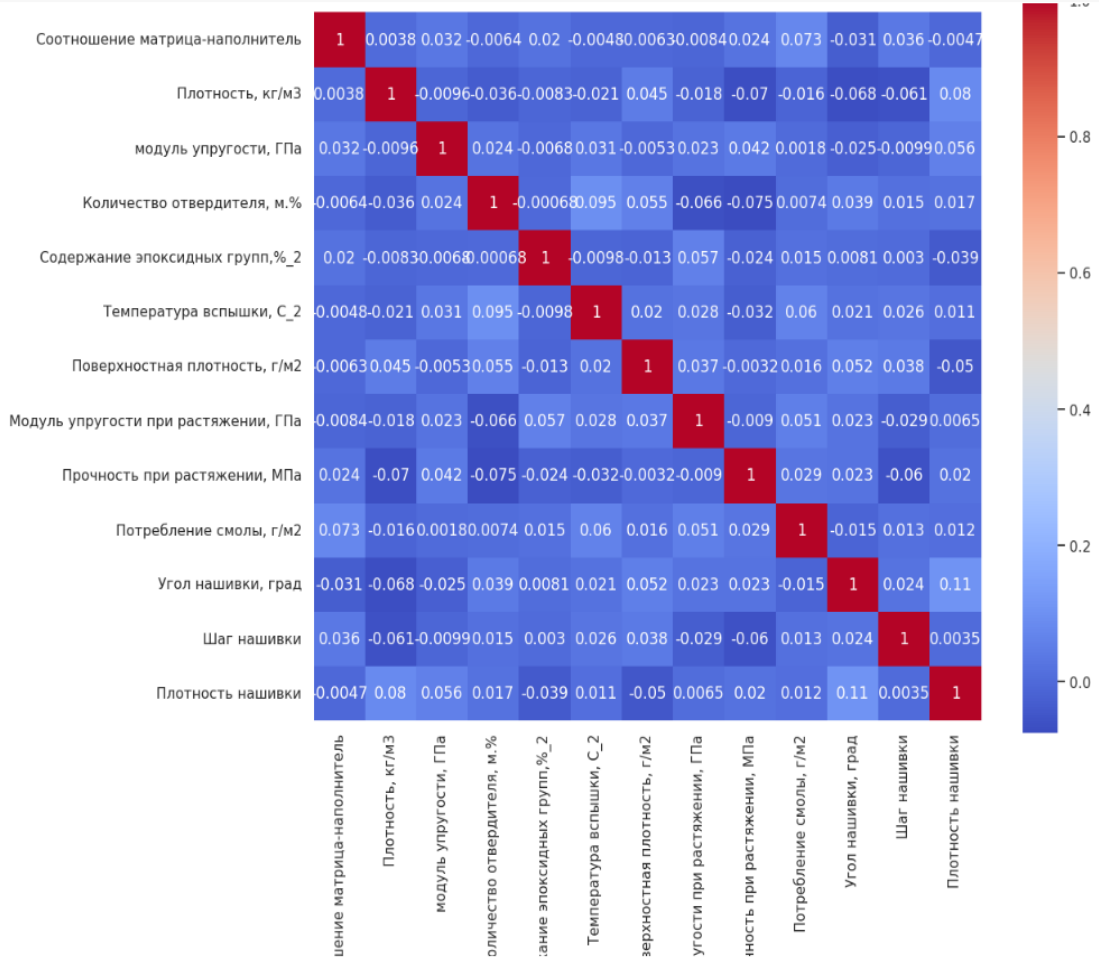




ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разведочный анализ данных

```
f, ax = plt.subplots(figsize=(12,10))
sns.heatmap(df.corr(), annot=True, ax=ax, square = True, cmap='coolwarm')
plt.show()
```



- Данные объединенного датасета не имеют выраженной зависимости
- В датасете есть выбросы

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

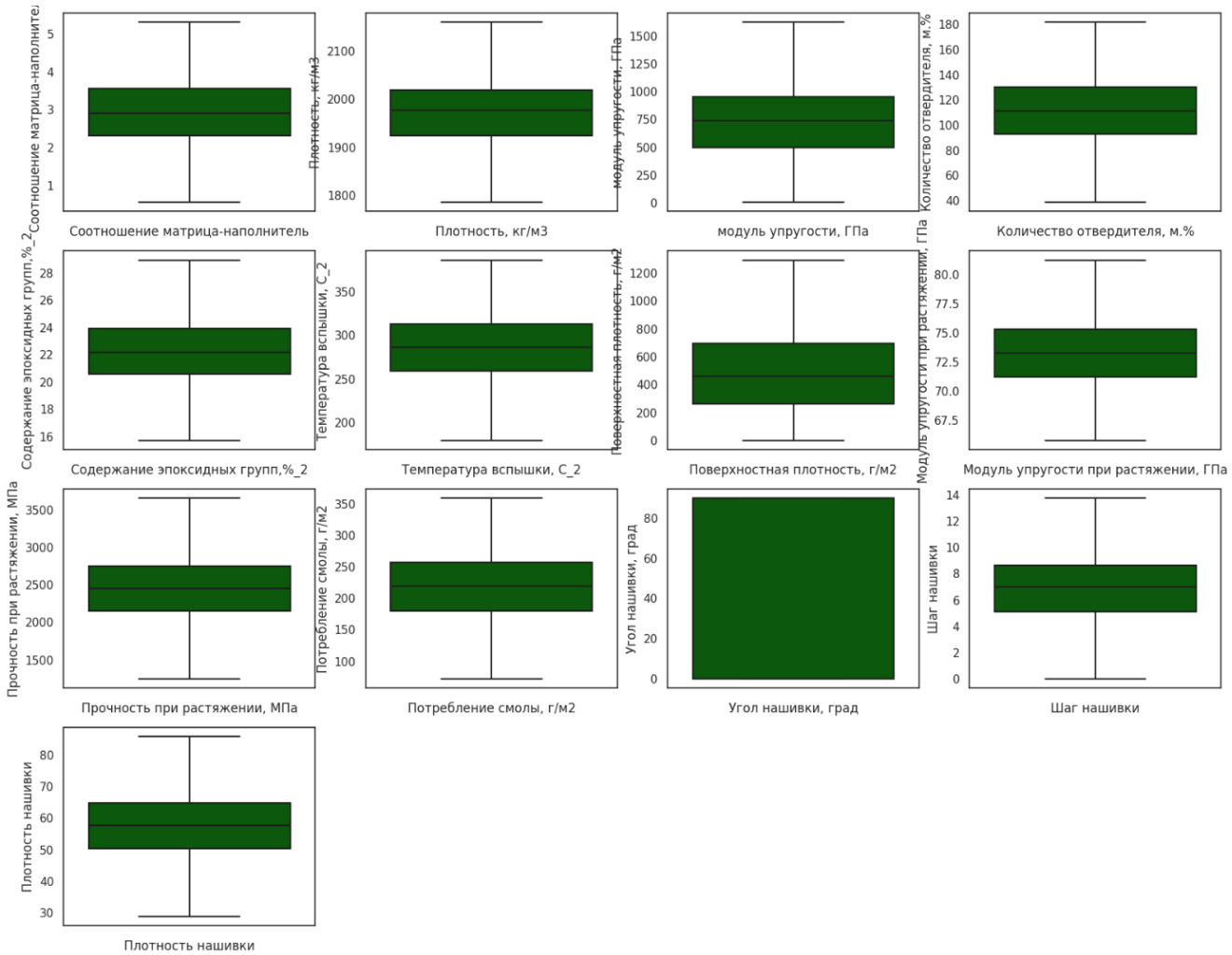
Предобработка данных

Удаление выбросов

```
[13] df_clean = df.copy()
     for col in df_clean.columns:
         q75,q25 = np.percentile(df_clean.loc[:,col],[75,25])
         intr_qr = q75-q25

         max = q75+(1.5*intr_qr)
         min = q25-(1.5*intr_qr)

         df_clean.loc[df_clean[col] < min,col] = np.nan
         df_clean.loc[df_clean[col] > max,col] = np.nan
```

```
[14] df_clean.isnull().sum()
```

```
Соотношение матрица-наполнитель          6
Плотность, кг/м3                         9
модуль упругости, ГПа                    2
Количество отвердителя, м.%             14
Содержание эпоксидных групп,%_2          2
Температура вспышки, С_2                 8
Поверхностная плотность, г/м2            2
Модуль упругости при растяжении, ГПа     6
Прочность при растяжении, МПа           11
Потребление смолы, г/м2                  8
Угол нашивки, град                       0
Шаг нашивки                              4
Плотность нашивки                       21
dtype: int64
```



- Датасет очищен от выбросов в 3 подхода

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Предобработка данных

Проверка на нормальность (тест Шапиро-Уилка)

```
from scipy.stats import shapiro
for col in df_clean_3.columns:
    print(df_clean_3[col].name, shapiro(df_clean_3[col]))
```

```
Соотношение матрица-наполнитель ShapiroResult(statistic=0.9971880316734314, pvalue=0.10904344916343689)
Плотность, кг/м3 ShapiroResult(statistic=0.997011661529541, pvalue=0.08352091163396835)
модуль упругости, ГПа ShapiroResult(statistic=0.995254397392273, pvalue=0.005813030526041984 6)
Количество отвердителя, м.% ShapiroResult(statistic=0.9966756105422974, pvalue=0.05000615119934082)
Содержание эпоксидных групп,%_2 ShapiroResult(statistic=0.9977133870124817, pvalue=0.23541033267974854)
Температура вспышки, С_2 ShapiroResult(statistic=0.9971266984939575, pvalue=0.09941922873258591)
Поверхностная плотность, г/м2 ShapiroResult(statistic=0.9776217937469482, pvalue=1.0688074730813568e-10)
Модуль упругости при растяжении, ГПа ShapiroResult(statistic=0.9955782890319824, pvalue=0.009416559711098671)
Прочность при растяжении, МПа ShapiroResult(statistic=0.9973730444908142, pvalue=0.14375117421150208)
Потребление смолы, г/м2 ShapiroResult(statistic=0.9955164790153503, pvalue=0.008584197610616684)
Угол нашивки, град ShapiroResult(statistic=0.6364129781723022, pvalue=2.8589291269154918e-40)
Шаг нашивки ShapiroResult(statistic=0.9980173110961914, pvalue=0.3559962809085846)
Плотность нашивки ShapiroResult(statistic=0.9967383742332458, pvalue=0.05505112186074257)
```

Нормализация данных

1) Нормализация с помощью MinMaxScaler

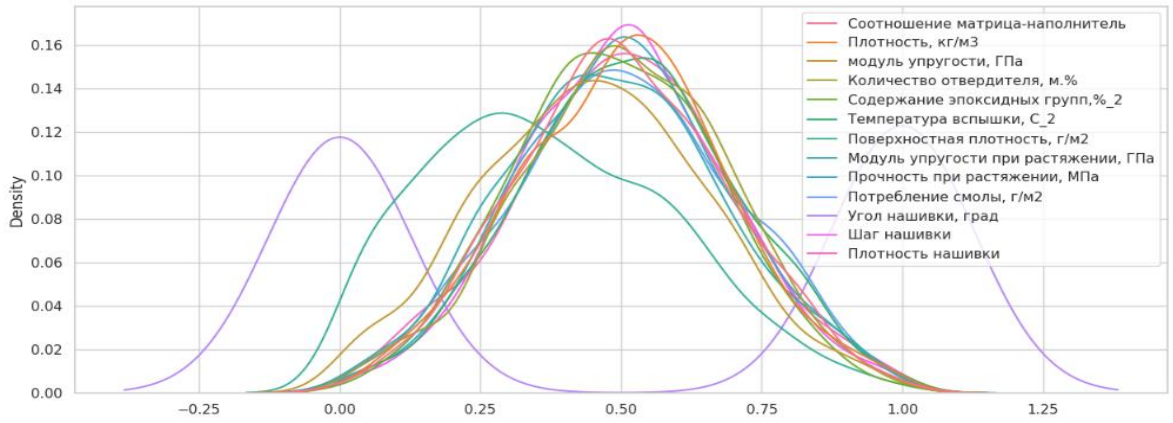
```
min_max_scaler = MinMaxScaler()
df_norm_minmax = pd.DataFrame(min_max_scaler.fit_transform(df_clean_3), columns = df_clean_3.columns, index=df_clean_3.index)
```

```
df_norm_minmax.describe()
```

	Соотношение матрица-наполнитель	Плотность, кг/м3	модуль упругости, ГПа	Количество отвердителя, м.%	Содержание эпоксидных групп,%_2	Температура вспышки, С_2	Поверхностная плотность, г/м2	Модуль упругости при растяжении, ГПа	Прочность при растяжении, МПа	Потребление смолы, г/м2	Угол нашивки, град	Шаг нашивки	Плотность нашивки
count	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000	922.000000
mean	0.499412	0.502904	0.451341	0.506200	0.490578	0.516739	0.373295	0.487343	0.503776	0.507876	0.510846	0.503426	0.503938
std	0.187858	0.188395	0.201534	0.186876	0.180548	0.190721	0.217269	0.196366	0.188668	0.199418	0.500154	0.183587	0.193933
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	0.371909	0.368184	0.305188	0.378514	0.366571	0.386228	0.204335	0.353512	0.373447	0.374647	0.000000	0.372844	0.376869
50%	0.495189	0.511396	0.451377	0.506382	0.488852	0.516931	0.354161	0.483718	0.501481	0.510143	1.000000	0.506414	0.504310
75%	0.629774	0.624719	0.587193	0.638735	0.623046	0.646553	0.538397	0.617568	0.624299	0.642511	1.000000	0.626112	0.630842
max	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000

Визуализируем полученный результат

```
plt.figure(figsize=(15, 6))
sns.set(context='notebook', style='whitegrid')
sns.kdeplot(data=df_norm_minmax)
plt.show()
```



ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Создание и обучение моделей

Разбиваем на тестовую, тренировочную выборки

```
[ ]  #1) Выборка для прогноза Прочность при растяжении, МПа
     X_train_1, X_test_1, y_train_1, y_test_1 = train_test_split(
         df_norm.loc[:, df_norm.columns != 'Прочность при растяжении, МПа'],
         df_norm[['Прочность при растяжении, МПа']],
            test_size = 0.3,
         random_state = 42)

     #2) Выборка для прогноза Модуль упругости при растяжении, ГПа
     X_train_2, X_test_2, y_train_2, y_test_2 = train_test_split(
         df_norm.loc[:, df_norm.columns != 'Модуль упругости при растяжении, ГПа'],
         df_norm[['Модуль упругости при растяжении, ГПа']],
            test_size = 0.3,
         random_state = 42)
```

Созданные модели :

- Метод К-ближайших соседей
- Линейная регрессия
- Случайный лес
- Метод опорных векторов
- Градиентный бустинг
- Дерево решений
- Lasso регрессия.

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Создание и обучение моделей

1) K-nearest neighbors

```
[ ]  # 1) Прочность при растяжении, МПа
     knn = KNeighborsRegressor(n_neighbors=5)
     knn.fit(X_train_1, y_train_1)
     y_pred_knn = knn.predict(X_test_1)
     mae_knr = mean_absolute_error(y_pred_knn, y_test_1)
     mse_knn_elast = mean_squared_error(y_test_1,y_pred_knn)
     Score_KNN_1 = knn.score(X_test_1, y_test_1)

     # 2) Модуль упругости при растяжении, ГПа
     knn2 = KNeighborsRegressor(n_neighbors=5)
     knn2.fit(X_train_2, y_train_2)
     y_pred_knn2 = knn2.predict(X_test_2)
     mae_knr2 = mean_absolute_error(y_pred_knn2, y_test_2)
     mse_knn_elast2 = mean_squared_error(y_test_2,y_pred_knn2)
     Score_KNN_2 = knn2.score(X_test_2, y_test_2)
```

```
K-nearest neighbors train data results:
Train score: 0.23
K-nearest neighbors test data results:
MAE:  0.17
MAPE: 6526587660746.40
MSE: 0.04
RMSE: 0.21
Test score: -0.18
==================================================
K-nearest neighbors train data results:
Train score: 0.18
K-nearest neighbors test data results:
MAE:  0.18
MAPE: 0.68
MSE: 0.05
RMSE: 0.23
Test score: -0.22
```

- Пример модели

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Создание и обучение моделей

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Создание и обучение моделей

	target_var	model_name	MAE	MAPE	MSE	RMSE	R2
0	Модуль упругости при растяжении, ГПа	K-NN	0.17	6526587660746.40	0.04	0.21	-0.18
1	Прочность при растяжении, МПа	K-NN	0.18	0.68	0.05	0.23	-0.22
2	Модуль упругости при растяжении, ГПа	Linear_Regression	0.16	9232724774322.15	0.04	0.19	-0.01
3	Прочность при растяжении, МПа	Linear_Regression	0.17	0.65	0.04	0.21	-0.03
4	Модуль упругости при растяжении, ГПа	Random_Forest	0.16	8769553567468.01	0.04	0.20	-0.09
5	Прочность при растяжении, МПа	Random_Forest	0.17	0.67	0.04	0.21	-0.06
6	Модуль упругости при растяжении, ГПа	Support_Vector_Regression	0.15	8176304369191.76	0.04	0.19	-0.00
7	Прочность при растяжении, МПа	Support_Vector_Regression	0.17	0.68	0.04	0.20	-0.00
8	Модуль упругости при растяжении, ГПа	Gradient_Boosting	0.16	8395674120401.24	0.04	0.20	-0.04
9	Прочность при растяжении, МПа	Gradient_Boosting	0.17	0.68	0.05	0.21	-0.09
10	Модуль упругости при растяжении, ГПа	Decision Tree Regression	0.19	8198168168923.67	0.06	0.25	-0.72
11	Прочность при растяжении, МПа	Decision Tree Regression	0.24	0.83	0.09	0.30	-1.11
12	Модуль упругости при растяжении, ГПа	Lasso_Regressor	0.15	8183534682026.43	0.04	0.19	-0.00
13	Прочность при растяжении, МПа	Lasso_Regressor	0.17	0.65	0.04	0.21	-0.01

- Таблица с метриками моделей, где данные преобразованы с помощью MinMaxScaler

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Создание и обучение моделей

	target_var	model_name	MAE	MAPE	MSE	RMSE	R2
0	Модуль упругости при растяжении, ГПа	K-NN	0.01	0.02	0.00	0.02	0.94
1	Прочность при растяжении, МПа	K-NN	0.00	0.05	0.00	0.00	0.72
2	Модуль упругости при растяжении, ГПа	Linear_Regression	0.01	0.02	0.00	0.01	0.96
3	Прочность при растяжении, МПа	Linear_Regression	0.00	0.04	0.00	0.00	0.79
4	Модуль упругости при растяжении, ГПа	Random_Forest	0.01	0.02	0.00	0.02	0.94
5	Прочность при растяжении, МПа	Random_Forest	0.00	0.04	0.00	0.00	0.78
6	Модуль упругости при растяжении, ГПа	Support_Vector_Regression	0.07	0.10	0.01	0.09	-0.45
7	Прочность при растяжении, МПа	Support_Vector_Regression	0.00	0.11	0.00	0.00	-0.17
8	Модуль упругости при растяжении, ГПа	Gradient_Boosting	0.01	0.01	0.00	0.01	0.98
9	Прочность при растяжении, МПа	Gradient_Boosting	0.00	0.04	0.00	0.00	0.78
10	Модуль упругости при растяжении, ГПа	Decision Tree Regression	0.01	0.02	0.00	0.02	0.93
11	Прочность при растяжении, МПа	Decision Tree Regression	0.00	0.06	0.00	0.00	0.54
12	Модуль упругости при растяжении, ГПа	Lasso_Regressor	0.06	0.08	0.01	0.07	-0.00
13	Прочность при растяжении, МПа	Lasso_Regressor	0.00	0.09	0.00	0.00	-0.00
14	Соотношение матрица/наполнитель	Model 1	0.92	0.39	1.32	1.15	-0.86
15	Соотношение матрица/наполнитель	Model 2	1.88	0.62	4.22	2.05	-4.95

- Таблица с метриками моделей, где данные преобразованы с помощью Normalizer

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Создание и обучение нейросетей

Создание модели и слоёв

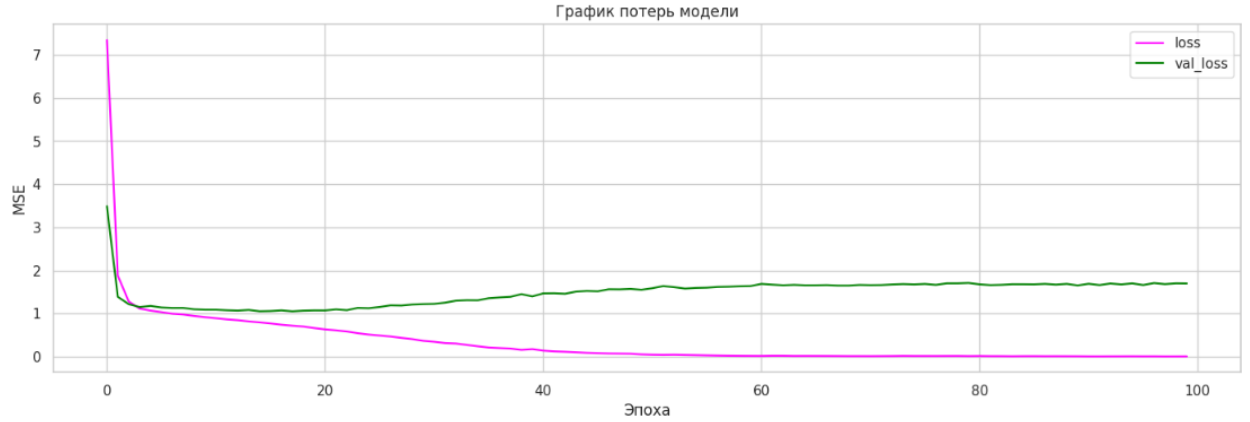
```
[357] X_train_norm = X_train_normalizer

      model = tf.keras.Sequential([X_train_norm, layers.Dense(128, activation='relu'),
                                                 layers.Dense(64, activation='relu'),
                                                 layers.Dense(32, activation='relu'),
                                                 layers.Dense(16, activation='relu'),
                                                 layers.Dense(1)
                                                ])

      model.compile(optimizer=tf.keras.optimizers.Adam(0.001), loss='mean_squared_error')
```

Архитектура модели

```
[358] model.summary()

Model: "sequential_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 normalization_2 (Normalizat  (None, 12)               25
 ion)

 dense_26 (Dense)            (None, 128)               1664

 dense_27 (Dense)            (None, 64)                8256

 dense_28 (Dense)            (None, 32)                2080

 dense_29 (Dense)            (None, 16)                528

 dense_30 (Dense)            (None, 1)                 17

=================================================================
Total params: 12,570
Trainable params: 12,545
Non-trainable params: 25
_________________________________________________________________
```

Визуализация ошибок

```
[361] model_loss_plot(model_hist)
```



Результат работы модели

```
[362] pred_plot(y_test.values, model.predict(X_test.values), 'Соотношение матрица/наполнитель', 'Keras_neuronet')

10/10 [==============================] - 0s 2ms/step
```

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Создание и обучение нейросетей

	target_var	model_name	MAE	MAPE	MSE	RMSE	R2
0	Модуль упругости при растяжении, ГПа	K-NN	0.17	6526587660746.40	0.04	0.21	-0.18
1	Прочность при растяжении, МПа	K-NN	0.18	0.68	0.05	0.23	-0.22
2	Модуль упругости при растяжении, ГПа	Linear_Regression	0.16	9232724774322.15	0.04	0.19	-0.01
3	Прочность при растяжении, МПа	Linear_Regression	0.17	0.65	0.04	0.21	-0.03
4	Модуль упругости при растяжении, ГПа	Random_Forest	0.16	8769553567468.01	0.04	0.20	-0.09
5	Прочность при растяжении, МПа	Random_Forest	0.17	0.67	0.04	0.21	-0.06
6	Модуль упругости при растяжении, ГПа	Support_Vector_Regression	0.15	8176304369191.76	0.04	0.19	-0.00
7	Прочность при растяжении, МПа	Support_Vector_Regression	0.17	0.68	0.04	0.20	-0.00
8	Модуль упругости при растяжении, ГПа	Gradient_Boosting	0.16	8395674120401.24	0.04	0.20	-0.05
9	Прочность при растяжении, МПа	Gradient_Boosting	0.17	0.68	0.05	0.21	-0.10
10	Модуль упругости при растяжении, ГПа	Decision Tree Regression	0.20	8286233635861.67	0.06	0.25	-0.68
11	Прочность при растяжении, МПа	Decision Tree Regression	0.25	0.85	0.09	0.30	-1.22
12	Модуль упругости при растяжении, ГПа	Lasso_Regressor	0.15	8183534682026.43	0.04	0.19	-0.00
13	Прочность при растяжении, МПа	Lasso_Regressor	0.17	0.65	0.04	0.21	-0.01
14	Соотношение матрица/наполнитель	Model 1	0.94	0.41	1.44	1.20	-1.02
15	Соотношение матрица/наполнитель	Model 2	2.78	0.96	8.47	2.91	-10.94
16	Соотношение матрица/наполнитель	Model 3	0.80	0.34	0.95	0.98	-0.34
17	Соотношение матрица/наполнитель	Model 4	0.95	0.39	1.34	1.16	-0.90
18	Соотношение матрица/наполнитель	Model 5	0.68	0.30	0.71	0.84	-0.00

- Таблица с метриками моделей и нейросетей

ЦЕНТР ДОПОЛНИТЕЛЬНОГО ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

Разработка приложения

```
Приложение даёт прогноз соотношение матрица-наполнитель
Введите "да" для прогноза, "нет" для выхода
да
Введите данные
Плотность, кг/м3: 2030
Модуль упругости, ГПа: 738
Количество отвердителя, м.%: 30
Содержание эпоксидных групп,%_2: 22
Температура вспышки, С_2: 100
Поверхностная плотность, г/м2: 210
Модуль упругости при растяжении, ГПа: 70
Прочность при растяжении, МПа: 3000
Потребление смолы, г/м2: 220
Угол нашивки: 0
Шаг нашивки: 4
Плотность нашивки: 57
1/1 [==============================] - 0s 46ms/step
Прогноз значения соотношение матрица-наполнитель:
79.90848
Введите "да" для прогноза, "нет" для выхода
нет
```

```python
def app_model():
    #nn_model = load_model('C:/Users/serzh/OneDrive/Bureau//model_1/')
    application_model = model
    data_x = load('/content/df_x_minmax.pkl')
    data_y = load('/content/df_y_minmax.pkl')

    print('Приложение даёт прогноз соотношение матрица-наполнитель')
    for i in range(1000):
        try:
            print('Введите "да" для прогноза, "нет" для выхода')
            check = input()

            if check == 'да':
                print('Введите данные')
                X = input_variable()
                X =  data_x.transform(np.array(X).reshape(1,-1))
                prediction = application_model.predict(X)
                output = data_y.inverse_transform(prediction)
                print('Прогноз значения соотношение матрица-наполнитель: ')
                print(output[0][0])

            elif check == 'нет':
                break
            else:
                print('Повторите выбор')

        except Exception as e:
            print(e)
            print('Данные неверны. Повторите ввод')

app_model()
```

Введите значение

x1
x2
x3
x4
x5
x6
x7
x8
x9
x10
x11
x12

Soumettre

app.py 2 | model_vkr.pkl | main.html

C: > Users > serzh > OneDrive > Bureau > BKP > Application > app.py > ...

```python
import flask
from flask import render_template
import pickle
import pandas as pd
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras import initializers

app = flask.Flask(__name__, template_folder = 'templates')

@app.route('/', methods = ['POST', 'GET'])

@app.route('/index', methods = ['POST', 'GET'])
def main():
    if flask.request.method == 'GET':
        return render_template('main.html')

    if flask.request.method == 'POST':

        x_list = []
        for x in range(1,13,1):
            x_list.append(float(flask.request.form[f'x{x}']))
        print(x_list)
        with open ('model_vkr.pkl', 'rb') as f:
            loaded_model = pickle.load(f)

        y_pred = loaded_model.predict(np.array(x_list).reshape(1, -1))

        return render_template('main.html', result = y_pred)

if __name__== '__main__':
    app.run()
```

ЦЕНТР
ДОПОЛНИТЕЛЬНОГО
ОБРАЗОВАНИЯ
МГТУ им. Н.Э. Баумана

do.bmstu.ru