

Образовательный центр МГТУ им. Н.Э. Баумана

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

к выпускной квалификационной работе
по курсу «Аналитик данных (Data scientist)»

Выполнил:

Альчаков Василий Викторович
слушатель гр. AD11872/3

2022

СОДЕРЖАНИЕ

1 Описание кейса «Прогнозирование размеров сварного шва при электронно-лучевой сварке тонкостенных конструкций аэрокосмического назначения»	3
2 Анализ входного набора данных и создание синтетического набора данных для обучения	4
3 Разведочный анализ данных	5
4 Построение моделей машинного обучения и нейронных сетей ...	7
5 Экспорт модели.....	10
6 Разработка программного обеспечения	13
7 Структура репозитория проекта.....	16
8 Заключение	17
Ссылки на ресурсы	17
Контакты для связи	17

1 Описание кейса «Прогнозирование размеров сварного шва при электронно-лучевой сварке тонкостенных конструкций аэрокосмического назначения»

В качестве исходных данных были взяты результаты экспериментальных исследований, проводимых в целях улучшения технологического процесса электронно-лучевой сварки изделия, сборка которого состоит из элементов, состоящих из разнородного материала. Установка электронно-лучевой сварки, на которой проводились исследования, предназначена для сварки электронным лучом в глубоком вакууме деталей сборочных единиц из нержавеющей сталей, титановых, алюминиевых и специальных сплавов. Существующая установка электронно-лучевой сварки обеспечивает повторяемость режимов в рамках возможностей реализованной системы управления. Работы по сварке выполнялись на образцах-имитаторах, соответствующих технологическому изделию. Для уменьшения вложения энергии при сварке:

1. Снижалась величина сварочного тока (IW);
2. Увеличивался ток фокусировки электронного пучка (IF);
3. Увеличивалась скорость сварки (VW);
4. Менялось расстояние от поверхности образцов до электронно-оптической системы (FP).

По совокупности параметров технологических режимов обеспечивались минимально возможные размеры сварных швов: глубина шва (Depth) и ширина шва (Width).

В процессе выполнения работ была произведена электронно-лучевая сварка 18-ти единиц образцов. Результаты металлографического контроля по размерам сварного шва для каждого образца проводились в 4-х поперечных сечениях сварного шва. Ускоряющее напряжение было постоянным в диапазоне 19,8 – 20 кВ. Набор полученных данных собраны в составе режимов сварки, размеров сварочных швов в поперечных сечениях всех образцов.

Требуется: провести прогнозирование глубины (Depth) и ширины (Width) сварного шва в зависимости от параметров технологического процесса (IW, IF, VW, FP).

2 Анализ входного набора данных и создание синтетического набора данных для обучения

Первичный анализ входного набора в Excel показывает, что исходный набор данных содержит повторяющиеся комбинации для набора входных параметров (IW, IF, VW, FP), которым соответствуют различные значения целевых параметров (Depth, Width) (рис. 2.1).

	A	B	C	D	E	F
1	IW	IF	VW	FP	Depth	Width
2	47	139	4.5	80	1.6	2.54
3	47	139	4.5	80	1.62	2.5
4	47	139	4.5	80	1.68	2.6
5	47	139	4.5	80	1.58	2.52
6						
7	45	140	4.5	80	1.76	2.48
8	45	140	4.5	80	1.72	2.48
9	45	140	4.5	80	1.64	2.52
10	45	140	4.5	80	1.64	2.56
11						

Рис. 2.1 – Исходные набор данных

В таком виде данные не могут быть использованы для обучения модели, поскольку возникает неоднозначность – одинаковой комбинации входов, соответствуют различные значения выходов. Для решения данной проблемы предлагается выполнить группировку и использовать усредненные значения целевых признаков для каждой уникальной комбинации входных параметров (рис. 2.2).

	A	B	C	D	E	F	G	H
1	IW	IF	VW	FP	Depth	Width	Depth	Width
2	47	139	4.5	80	1.6	2.54	1.62	2.54
3	47	139	4.5	80	1.62	2.5		
4	47	139	4.5	80	1.68	2.6		
5	47	139	4.5	80	1.58	2.52		
6								
7	45	140	4.5	80	1.76	2.48	1.69	2.51
8	45	140	4.5	80	1.72	2.48		
9	45	140	4.5	80	1.64	2.52		
10	45	140	4.5	80	1.64	2.56		

Рис. 2.2 – Группировка данных и получение усредненных значений целевых параметров

Данную операцию легко сделать с помощью стандартных функций `groupby` библиотеки `pandas`. Соответствующая операция выполняется в 15-й ячейке ноутбука `welding-analysis.ipynb` (рис. 2.3).

```
In [5]: train_data = raw_data.groupby(['IW', 'IF', 'VW', 'FP']).agg({'Depth': ['mean'],
train_data.columns = ['IW', 'IF', 'VW', 'FP', 'Depth', 'Width']
```

Рис. 2.3 – Группировка данных с помощью `groupby`

Таким образом, исходные 72 строки данных будут преобразованы в новый синтетический набор данных из 15 строк, который может быть использован для обучения. Исходная выборка будет использована для проверки точности построенных моделей. Таким образом, использование функции `train_test_split` библиотеки `sklearn` для разбиения исходного набора данных на обучающую и тестовую выборки не требуется.

3 Разведочный анализ данных

Разведочный анализ сгенерированной синтетической выборки показывает, что пропуски в наборе данных отсутствуют. Корреляционная матрица (рис. 3.1) свидетельствует о том, что все параметры входного набора имеют влияние на целевые параметры. Параметр `VW` (скорость сварки) в большей степени влияет и на глубину, и на ширину шва, параметр `FP` (расстояние от поверхности образцов) имеет влияние на глубину шва, остальные параметры практически в равной степени влияют и на глубину, и на ширину. Поэтому в модели будут учтены все четыре входных параметра.

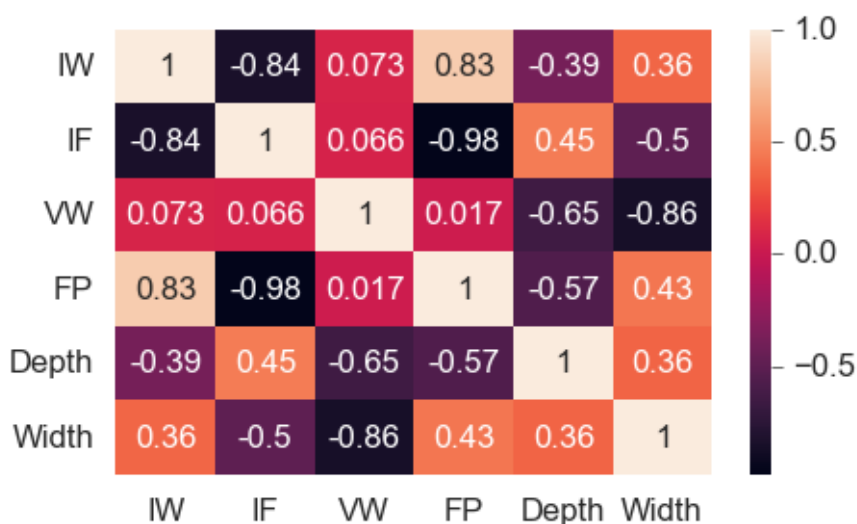


Рис. 3.1 – Корреляционная матрица в виде тепловой карты

Исходное распределения параметров близко к нормальному, однако со смещением, о чем свидетельствуют соответствующие графические

зависимости параметров, полученные с помощью функций `pairplot` и `distplot` библиотеки `seaborn` (рис. 3.2).

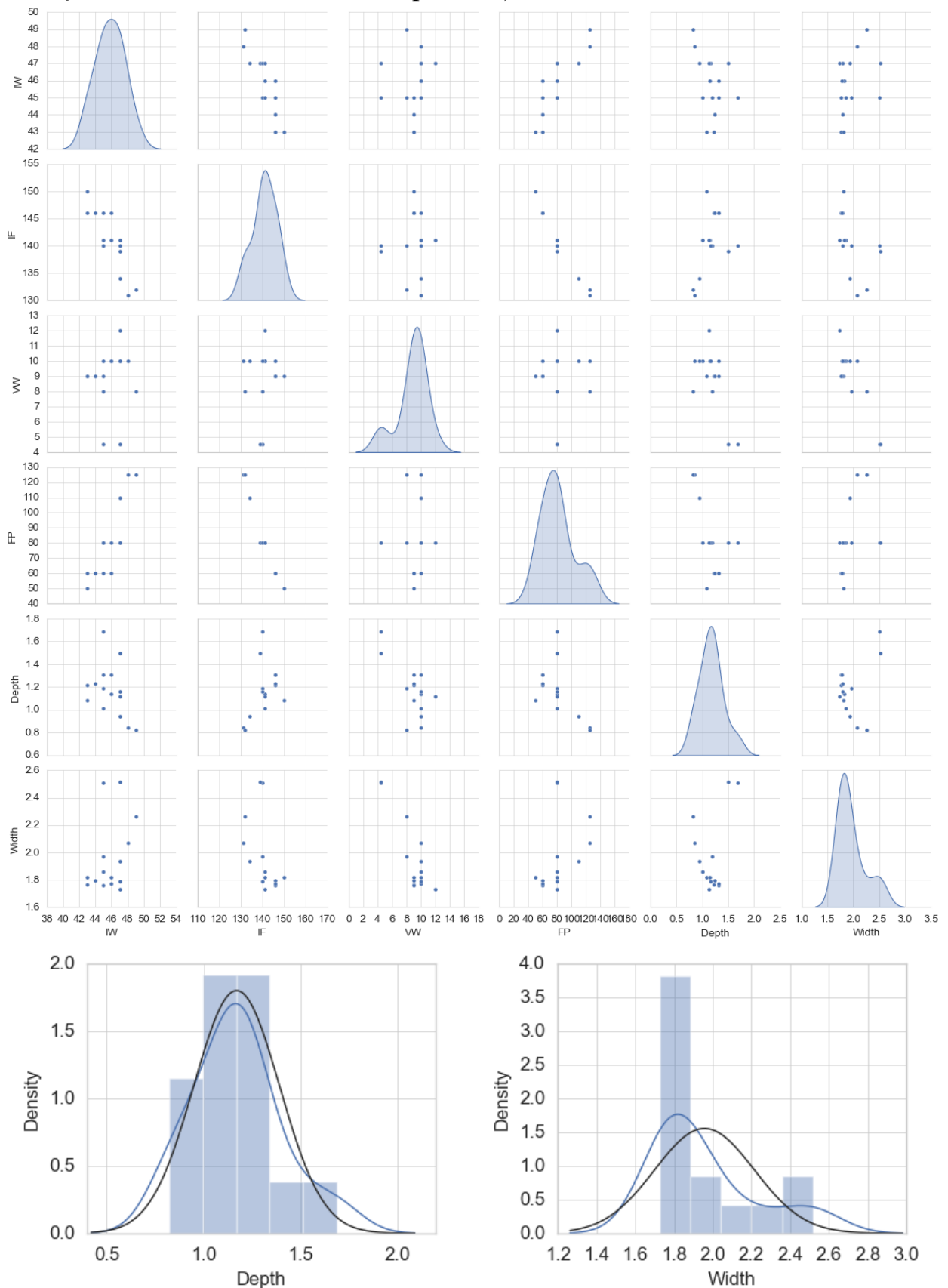


Рис. 3.2 – Статистическое распределение параметров набора данных

Аналогичные выводы позволяют сделать графики распределения вероятностей (probability plot) – присутствует отклонение данных от нормального распределения (рис. 3.3), следовательно, необходимо будет выполнить нормализацию выборки перед построением модели

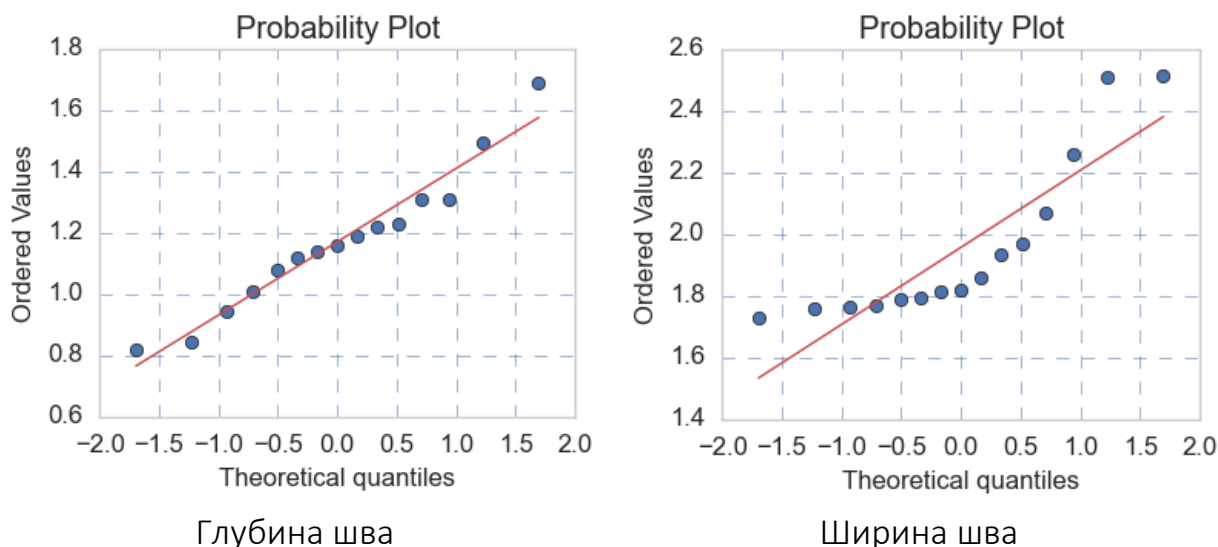


Рис. 3.2 – График распределения вероятностей

4 Построение моделей машинного обучения и нейронных сетей

В качестве моделей машинного обучения были выбраны три метода:

- линейная регрессия (LinearRegression);
- полиномиальная регрессия (PolynomialFeatures);
- метод опорных векторов (SupportVectorRegression).

Реализация методов выполнена с помощью библиотеки `scikit-learn`.

Перед применением одного из указанных методов построения модели выполняется преобразование данных на основе применения `StandardScaler`.

Для сокращения кода, повторяющиеся вызовы команд выведены в отдельные функции:

`plt_model(x_index, y_test, y_predict, title_txt)` – функция отрисовки результатов моделирования. Входные параметров:

`x_index` – вектор значений индексов по оси абсцисс;

`y_test` – вектор значений целевого параметра тестовой выборки;

`y_predict` – вектор предсказанных значений целевого параметра;

`title_txt` – текстовая метка для отрисовки заголовка.

`fit_model(pipe, x_train, y_train, x_test, y_test)` – функция обучения модели. Входные параметры:

`pipe` – конвейер для обучения модели, полученный с помощью функции `make_pipeline`;

`x_train` – матрица входных параметров обучающей выборки;

`y_train` – вектор значений целевого параметра обучающей выборки;

`x_test` – матрица входных параметров тестовой выборки;

`y_test` – вектор значений целевого параметра тестовой выборки.

`run_model(pipe, x_train, y_train, x_test, y_test, model_type, model_name)` – функция сборки и прогона модели. Входные параметры:

`pipe` – конвейер для обучения модели, полученный с помощью функции `make_pipeline`;

`x_train` – матрица входных параметров обучающей выборки;

`y_train` – вектор значений целевого параметра обучающей выборки;

`x_test` – матрица входных параметров тестовой выборки;

`y_test` – вектор значений целевого параметра тестовой выборки;

`model_type` – тип модели { `Width`; `Depth` };

`model_name` – название модели для отображения в заголовке;

Для оценки точности модели использована метрика MAE

$$MAE = \frac{1}{m} \sum_{i=1}^m |a_i - y_i|$$

a_i – предсказанные значения целевого признака;

y_i – истинные значения целевого признака;

m – количество элементов в выборке.

В результате построения и прогона модели выполняется визуализация, на которой в одних осях отображаются предсказанные и истинные значения целевого признака, а также выводятся значения метрики MAE в заголовке графика (рис. 4.1)

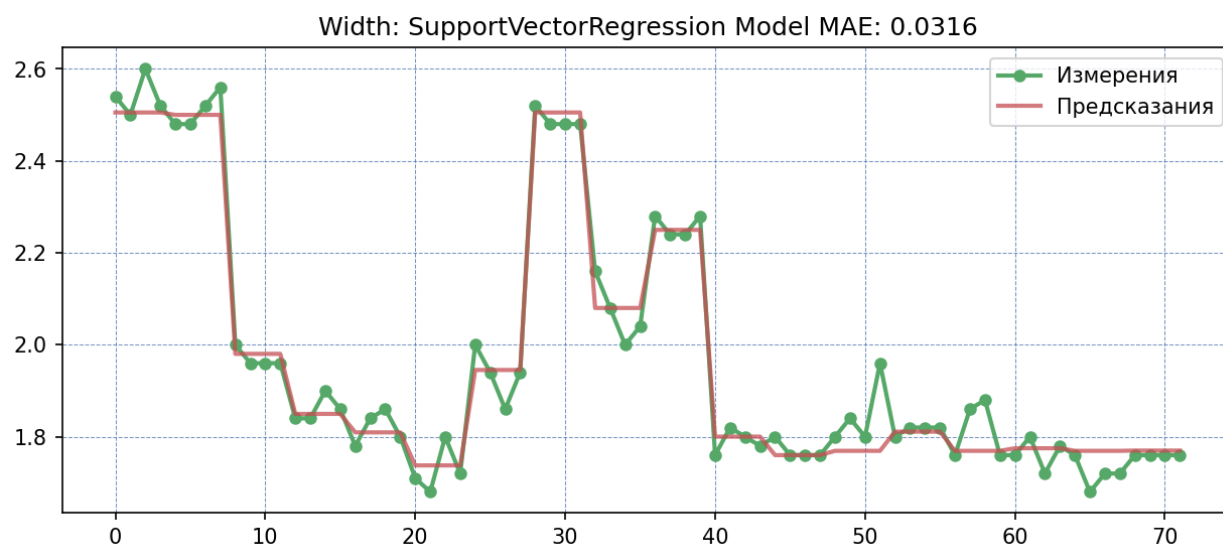


Рис. 4.1 – Результаты построения регрессионной модели с помощью метода SVR для ширины сварного шва MAE = 0,0316

Сравнение метрик MAE моделей приведены в табл. 4.1.

Табл. 4.1 – MAE метрики полученных моделей для целевых параметров

	LinearRegression	PolynomialFeatures	SupportVectorRegression
Width	0.0544	0.0306	0.0316
Depth	0.0746	0.0428	0.045

Как видно из табл. 4.1 наилучшие параметры показала модель PolynomialFeatures, наихудшие – LinearRegression

Следующая модель построена с помощью нейронной сети, состоящей из входного слоя – нормализатора, двух скрытых слоев и выходного слоя с двумя отдельными выходами для каждого из целевого параметров (рис. 4.2). В качестве активационной функции скрытых слоев использована активационная функция `relu`. Обучение модели выполнялось на 1000 эпохах без валидационной выборки. В качестве метрики также использовалась метрика MAE.

Model: "sequential"

Layer (type)	Output Shape	Param #
normalization (Normalization)	(None, 4)	9
dense (Dense)	(None, 64)	320
dense_1 (Dense)	(None, 32)	2080
dense_2 (Dense)	(None, 2)	66
Total params: 2,475		
Trainable params: 2,466		
Non-trainable params: 9		

Рис. 4.2 – Структура нейронной сети

По точности модель нейронной сети уступает модели **PolynomialFeatures** что может быть обусловлено ограниченным размером обучающей выборки.

Табл. 4.2 – MAE метрики полученных моделей для целевых параметров

	Linear Regression	Polynomial Features	Support Vector Regression	Neural Network
Width	0.0544	0.0306	0.0316	0.031
Depth	0.0746	0.0428	0.0449	0.043

5 Экспорт модели

Для экспорта модели был использован формат ONNX (Open Neural Network Exchange), который представляет собой открытый формат моделей машинного обучения и позволяет перемещать и использовать модели между разными платформами и инструментами машинного обучения. ONNX разрабатывается и поддерживается совместно компаниями Microsoft, Facebook, Amazon и другими партнерами как проект с открытым исходным кодом.

Для экспорта будет использоваться модель, построенная на базе нейронной сети. У модели будет 4 входа – вектор значений параметров (IW, IF, VW, FP) и 2 выхода – (Depth, Width).

Экспорт осуществляется с помощью методов библиотек `onnx` и `tf2onnx`. Проверка загрузки модели выполняется с помощью библиотеки `onnxruntime`. Созданная модель сохраняется в файле `welding-model.onnx` в папке `model` проекта.

Для проверки правильности работы экспортированной модели выполнено сравнение результатов, полученных по первой строке входного массива данных.

Для нейронной сети после обучения получены результаты

```
In [40]: test_predictions = seq_model.predict(test_features[:1], verbose = 0)
print(f'Параметры: {test_features[:1]}')
print(f'Прогноз: {test_predictions}')
print(f'Истинное значение: {test_labels[:1]}')
```

Параметры: [[47. 139. 4.5 80.]]
Прогноз: [[1.485 2.518]]
Истинное значение: [[1.6 2.54]]

Для нейронной сети, загруженной из файла ONNX получен такой же результат

```
In [44]: # Создание переменной сессии для работы с моделью (загрузка модели)
session = rt.InferenceSession('model\welding-model.onnx')
# Создание вектора для параметров
X = np.array([[47.0, 139.0, 4.5, 80.0]], dtype = np.float32)
inputDetails = session.get_inputs()
# Получение прогноза
y = session.run(None, {inputDetails[0].name: X})
print(y)
```

[array([[1.485, 2.518]], dtype=float32)]

Для исследования структуры полученной модели в ONNX формате может быть использована программа Netron, которая позволяет представить структуру полученной сети в виде схемы (рис. 5.1). Также можно увидеть названия входных и выходных переменных, что будет использовано при написании кода в рамках разработки программного обеспечения.



Рис. 5.1 – Структура модели в программе Netron

6 Разработка программного обеспечения

В качестве платформы для разработки программного обеспечения выбрана платформа Microsoft .NET WinForms. Данный выбор обусловлен тем, что многие промышленные предприятия (для нужд которых в дальнейшем планируется использовать знания, полученные при освоении курса) в целях безопасности отключают доступ во внешнюю сеть, в т.ч. сеть Internet. Поэтому нужны именно локально работающие приложение с интуитивно-понятным интерфейсом. Кроме того, компания Microsoft в последние годы занимается разработкой и поддержкой библиотеки машинного обучения ML.NET, которая поддерживает формат ONNX. Таким образом, становится возможным использование моделей, построенных и исследованных в Python (в т.ч. Jupyter Notebook)

Внешний вид главного окна приложения представлен на рис. 6.1.

Welding Predictor 1.0

Путь к файлу модели
D:\SourceCode\jupyter-notebook\bmstu-data-scientist-attestation\modelwelding-model.onnx

Путь к файлу данных
D:\SourceCode\jupyter-notebook\bmstu-data-scientist-attestation\data\ebw_data.csv

IW	IF	VW	FP	Depth	Width
47	139	4.5	80	1.6	2.54
47	139	4.5	80	1.62	2.5
47	139	4.5	80	1.68	2.6
47	139	4.5	80	1.58	2.52
45	140	4.5	80	1.76	2.48
45	140	4.5	80	1.72	2.48
45	140	4.5	80	1.64	2.52
45	140	4.5	80	1.64	2.56
45	140	8	80	1.2	2
45	140	8	80	1.16	1.96
45	140	8	80	1.2	1.96
45	141	10	80	0.98	1.84
45	141	10	80	1.02	1.84
45	141	10	80	1.04	1.9
45	141	10	80	1	1.86
46	141	10	80	1.2	1.78

Входные параметры

Ток сварочного тока: 47

Ток фокусировки: 139

Скорость сварки: 4.5

Скорость сварки: 80

Выходные параметры

Глубина шва: 1.6

Ширина шва: 2.54

Прогноз

Глубина шва:

Ширина шва:

ВЫБОРОЧНЫЙ ПРОГНОЗ

ПРОГОН ВСЕЙ ВЫБОРКИ

ВЫХОД

Статистика по всей выборке

	MAE	MSE	RMSE	R2	MAPE
Глубина					
Ширина					

Рис. 6.1 – Внешний вид главного окна приложения

На главной форме представлены элементы, позволяющие указать путь к файлу ONNX модели и путь к CSV файлу с тестовыми данными, для которых будет построен прогноз. Пути к файлу данных и файлу модели сохраняются в файле настроек приложения. В данной версии приложения подразумевается, что входной набор содержит также данные для целевых переменных. Это позволит вычислить точность получаемых прогнозов. Если такие данные отсутствуют, можно последние два столбца данных заполнить нулями, в этом случае прогноз по-прежнему будет рассчитан, однако значения метрик не будут соответствовать действительности.

После загрузки CSV файла тестовая выборка будет представлена в табличном виде. При переходе по строкам таблицы, соответствующие значения параметров будут отображаться в текстовых полях групп **Входные параметры** и **Выходные параметры**. Группа **Прогноз** будет содержать прогнозируемые значения целевых переменных с указанием величины относительной ошибки предсказания, вычисленной по формуле

$$\delta = \frac{a - y}{y} 100\%$$

a – предсказанное значение;

y – истинное значение.

Для получения прогноза необходимо выбрать строку данных, убедиться, что соответствующие значения появились в текстовых полях (либо ввести значения вручную) и нажать кнопку **ВЫБОРОЧНЫЙ ПРОГНОЗ**. Результаты прогноза полностью совпадают с данными, полученными в Jupyter Notebook

Прогноз		
Глубина шва	1.49	6.88%
Ширина шва	2.52	0.79%

```
In [40]: test_predictions = seq_model.predict(test_features[:1], verbose = 0)
print(f'Параметры: {test_features[:1]}')
print(f'Прогноз: {test_predictions}')
print(f'Истинное значение: {test_labels[:1]}')
```

```
Параметры: [[ 47.  139.   4.5  80. ]]
```

```
Прогноз: [[1.485 2.518]]
```

```
Истинное значение: [[1.6  2.54]]
```

Точность по параметру глубина шва составила 6,88%, по параметру ширина шва – 0,79%.

Прогнозные значения полностью совпали (значения в приложении округлены до второго знака) $1,485 = 1,49$ и $2,518 = 2,52$ соответственно.

Также в программе предусмотрен режим прогона всей тестовой выборки, состоящей из 72 записей и подсчет различных метрик,

характеризующих точность модели. Для этого необходимо загрузить CSV файл с данными, и нажать на кнопку ПРОГОН ВСЕЙ ВЫБОРКИ, после чего дождаться окончания расчетов. В нижней части окна приложения будет выведена сводная статистика по выборке

44	146	9	60	1.2	1.68
44	146	9	60	1.24	1.72
44	146	9	60	1.2	1.72
45	146	9	60	1.36	1.76
45	146	9	60	1.28	1.76
45	146	9	60	1.28	1.76
45	146	9	60	1.32	1.76

Ширина шва 1.76

Прогноз
Глубина шва 1.31
Ширина шва 1.76

ВЫБОРОЧНЫЙ ПРОГНОЗ

ПРОГОН ВСЕЙ ВЫБОРКИ

ВЫХОД

Статистика по всей выборке

	MAE	MSE	RMSE	R2	MAPE
Глубина	0.04	0.0034	0.06	0.93	3.42%
Ширина	0.03	0.0019	0.04	0.98	1.61%

Для вычисления статистических метрик использовались соотношения

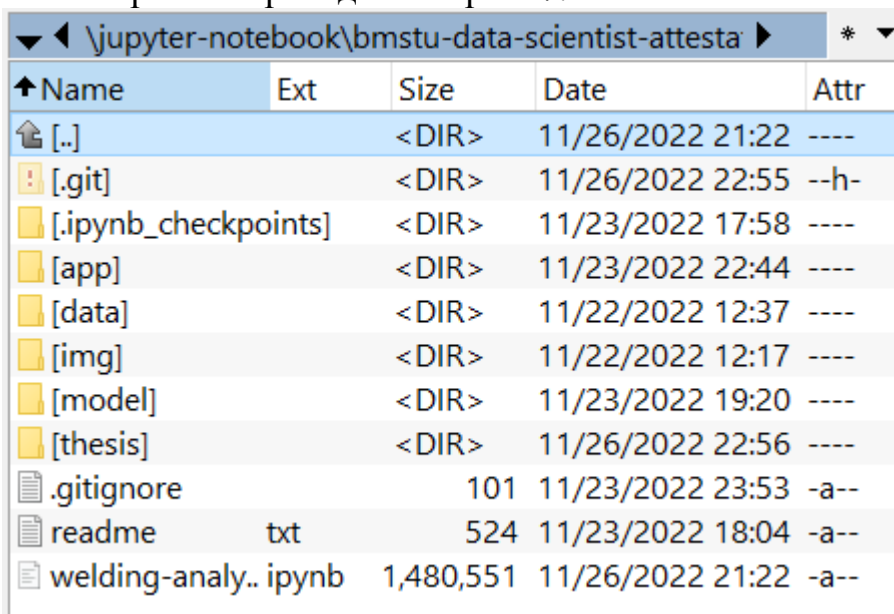
Табл. 6.1 – Метрики для оценки точности модели

Метрика	Формула	Значение	
		Depth	Width
MAE	$MAE = \frac{1}{m} \sum_{i=1}^m a_i - y_i $	0,03	0,04
MSE	$MSE = \frac{1}{m} \sum_{i=1}^m a_i - y_i ^2$	0,0019	0,0034
RMSE	$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m a_i - y_i ^2}$	0,04	0,06
R2	$R^2 = 1 - \frac{\sum_{i=1}^m a_i - y_i ^2}{\sum_{i=1}^m \bar{y} - y_i ^2}$ $\bar{y} = \frac{1}{m} \sum_{i=1}^m y_i$	0,98	0,93
MAPE	$MAPE = 100\% \frac{1}{m} \sum_{i=1}^m \frac{ a_i - y_i }{ y_i }$	1,61	3,42

7 Структура репозитория проекта

Все необходимые файлы для разворачивания проекта на локальной машине размещены в репозитории GitHub по адресу <https://github.com/VASoftLab/bmstu-data-scientist-attestation.git>

Структура папок проекта приведена на рис. 7.1



Name	Ext	Size	Date	Attr
[.]		<DIR>	11/26/2022 21:22	----
[.git]		<DIR>	11/26/2022 22:55	--h-
[.ipynb_checkpoints]		<DIR>	11/23/2022 17:58	----
[app]		<DIR>	11/23/2022 22:44	----
[data]		<DIR>	11/22/2022 12:37	----
[img]		<DIR>	11/22/2022 12:17	----
[model]		<DIR>	11/23/2022 19:20	----
[thesis]		<DIR>	11/26/2022 22:56	----
.gitignore		101	11/23/2022 23:53	-a--
readme	txt	524	11/23/2022 18:04	-a--
welding-analy.. ipynb		1,480,551	11/26/2022 21:22	-a--

Рис. 7.1 – Структура файлов и папок проекта

app – папка с исходными кодами десктопного приложения;

data – папка с файлами модели для обучения (исходный датасет в различных форматах);

img – папка с изображениями;

model – папка с моделью в формате ONNX, модель создается и сохраняется после выполнения скрипта `welding-analysis.ipynb` после чего данная модель читается десктопным приложением для получения прогнозной оценки параметров;

thesis – папка с файлами пояснительной записке к проекту;

`readme.txt` – краткая справка по структуре проекта

welding-analysis.ipynb – Jupyter notebook для анализа данных и построения модели

8 Заключение

В рамках выполнения выпускной квалификационной работы изучена возможность построения моделей машинного обучения и модели нейронной сети для предсказания значения целевых переменных для прогнозирования размеров сварного шва при электронно-лучевой сварке тонкостенных конструкций аэрокосмического назначения.

Наилучшую точность удалось получить с использованием простой регрессионной модели на базе полиномиальной аппроксимации. Точность полученной модели на базе нейронной сети несколько уступает полиномиальному представлению, что объясняется небольшим набором данных, доступных для исследования и построения модели.

Для упрощения разработки программного обеспечения, для экспорта была использована модель на базе нейронной сети. Для этого модель была экспортирована в формат ONNX, для загрузки модели, набора данных и получения прогноза было разработано настольное WinForm приложение с помощью языка программирования C# и библиотеки ML.NET.

Метрика R2 для полученной модели равна 0,98 для глубины и 0,93 для ширины сварного шва. Данные параметры очень близки к 1, что позволяет сделать заключение о хорошей точности полученной модели.

Ссылки на ресурсы

Репозиторий проекта

<https://github.com/VASoftLab/bmstu-data-scientist-attestation.git>

Видео демонстрация работы приложения

<https://youtu.be/c6v0IUnsBCo>

Ссылка для загрузки архива

<https://disk.yandex.ru/d/Vk64Ug3scoX13g>

Контакты для связи

Альчаков Василий Викторович

Моб.: +7-967-303-99-33

E-mail: alchakov@mail.ru

Telegram: @alchakov