1. Polymer Classification 2. Mass of Gas Prediction

Ryan Mokarian, 1/17/2023

Table of Contents

Q1. Classification

Question

The file classification_density.csv contains two kinds of laboratory measurements. The bulk density (lb/ft3) includes empty spaces between particles. In contrast, the particle density (g/cm3) includes only the polymer itself. Each polymer sample has a unique sample identifier, but the same sample may have been tested multiple times.

For most of the samples, the file classification_labels.csv identifies the physical form of the sample, such as powder or pellets. However, some of the samples do not appear in classification_labels.csv, so please predict the physical form of the unlabeled samples.

Solution

Data Pre-Processing

File	Shape	Primary Key	Unique samples	NaN Values
classification_density.csv	(3547, 3)	sample	1769	0
classification_labels.csv	(1609, 2)	sample	1609	0

From above information, it is concluded that there are 1769 samples. Out of them, 1609 includes label and can be used as training data in a supervised learning model. The rest of the samples are unlabeled and can be used as testing data to predict their physical form.

Note each of the 1769 unique samples in the "classification_density.csv" file normally should have two entries. However, 1769 X 2 = 3538 is different than the total number of entries, 3547. That means a few samples have entries different than 2. For those samples, average of those values is aggregated on their respective parameters.

Below table shows sample Ids when number of entries are different than 2.

Sample's number of entries	Sample Id
1	N/A
2	The rest
3	'22-0002184', '22-0002348', '22-0004617', '22-0007477', '22-0007478'
4	'21-0004611', '21-0004612'
More than 4	N/A

The data frame obtained from "classification_density.csv" file need to be restructured to a dataframe with unique sample entries and two columns presenting the mean values for bulk density and particle density. The original and transformed dataframe shown below for a sample '21-0004611' with multiple entries.

Original table:

	sample	parameter	value
3478	21-0004611	bulk density	27.60000
3479	21-0004611	particle density	1.08970
3480	21-0004611	bulk density	27.50000
3481	21-0004611	bulk density	27.50000

Transformed table after applying groupby and pivot:

	value/bulk density	value/particle density
21-0004611	27.53333	1.08970

In the next step, training and testing dataframes are made. In order to identify the testing samples, two dataset are joined over their sample as their primary key. Shape of the training and testing datasets are shown below.

```
df_density  (1769, 3)
df_label  (1609, 2)
df_train (1609, 4)
      sample  bulk_density  particle_density finalform
0  02-03324          28.0            1.0908    Powder
1  02-05854          41.1            1.1031   Pellets
```

```
df_test (160, 3)
       sample  bulk_density  particle_density
1   02-03325          28.6            1.0883
7   02-08391          35.8            1.0730
```

Training and testing data frames are saved as csv file in the root. Below is snapshots from the beginning of the files.

Training data frame

sample	bulk_density	particle_density	finalform
02-03324	-0.879476799	-0.018101881	Powder
02-05854	1.365875467	0.385059165	Pellets
02-06141	1.228754717	0.489946591	Pellets
02-06142	1.263034905	0.470280199	Pellets
02-06143	0.286049568	0.50305752	Pellets
02-06144	0.371750036	0.50305752	Pellets
02-08392	0.560291066	-0.611371388	Pellets

Testing data frame

sample	bulk_density	particle_density
02-03325	-0.993625672	-0.182158488
02-08391	0.454898818	-1.29004415
03-03680	-1.456348773	-0.927990012
03-08660	1.098687481	0.418851381
03-08761	-1.536822356	-0.022854667
03-08762	-1.214928025	-0.037336833
03-16541	-0.812560111	-1.601410709
03-18590	0.957858711	-1.528999881

Distribution of physical forms and balancing the data

First the data is scaled using **StandardScaler**.

Now, training data need to be checked if it is balanced over its class, "finalform". Five classes of the final form and their frequencies are shown in the following table.

Final form class	Granular	Pellets	Powder	Blown	Cast
Frequency	930	378	261	30	10

Training dataset is imbalanced in a range of maximum 930 entries for Granular to minimum 10 entries for Cast physical form. Initial plan was to study on two following balancing methods. However due to the shortage of time only the first one studied.

(1) SMOTE oversampling to balance number of entries for each climate type to 930. Therefore, total number of training data is increased to 5X930 = 4650.

Final form class	Granular	Pellets	Powder	Blown	Cast
Frequency after oversampling	930	930	930	930	930

(2) SMOTE oversampling on Blown and Cast and under-sampling on Granular

Model, Analysis and Results

Two models were used, k-nearest neighbors (KNN) and Artificial Neural Network ANN. Meanwhile, impact of imbalanced and balanced data on KNN model was investigated.

KNN Model – Imbalanced data

As it is shown below, the model performed well on Granular, Pellets, and Power forms and failed to predict on Cast and Blown forms. It is obvious as in the training the amount of data for these two forms are only 2% of total data (Total entries = 1609; Cast and Blown entries = 40).

```
Frequency of forms before balancing
 Granular    930
Pellets     378
Powder      261
Blown        30
Cast         10
```

KNN - Imbalanced Data:

	precision	recall	f1-score	support
Blown	0.00	0.00	0.00	8
Cast	0.00	0.00	0.00	2
Granular	0.93	0.96	0.95	233
Pellets	0.87	0.88	0.87	95
Powder	0.87	0.85	0.86	65
accuracy			0.90	403
macro avg	0.53	0.54	0.54	403
weighted avg	0.88	0.90	0.89	403

Note validated over 25% of the training data.

KNN Model – balanced data

After oversampling the minority classes using SMOTE method, the metrics values for minority classes significantly improved, as shown below. The problem of too much oversampling for the minority class with few samples is that the model is overfit on the minority classes and wouldn't generalized well on new data. For this reason, the best tradeoff is to increase minority class and decrease the majority class close to classes with average frequencies, in this case to a frequency close to number of entries for pallets and powder forms.

Frequency of forms after balancing	
Powder	930
Pellets	930
Granular	930
Blown	930
Cast	930

KNN - Balanced Data:

	precision	recall	f1-score	support
Blown	0.90	0.91	0.91	238
Cast	0.94	0.98	0.96	222
Granular	0.90	0.89	0.90	234
Pellets	0.88	0.83	0.85	230
Powder	0.93	0.95	0.94	239
accuracy			0.91	1163
macro avg	0.91	0.91	0.91	1163
weighted avg	0.91	0.91	0.91	1163

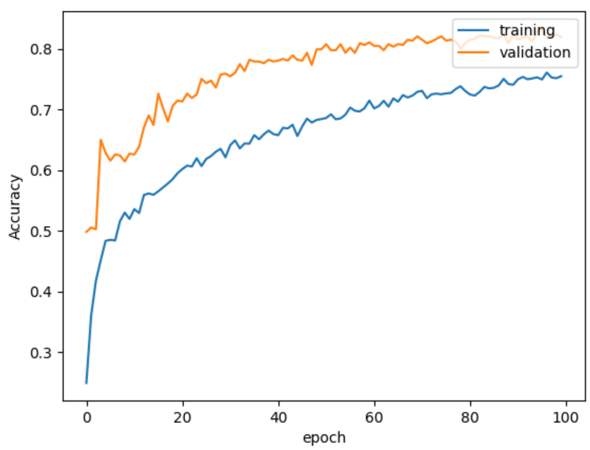
Note training data split 75-25 % for train and validation.

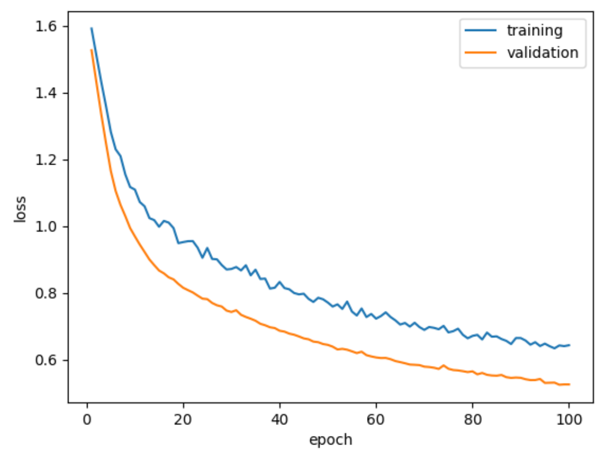
ANN – balanced data

ANN model structure:

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 16)                48

 dense_1 (Dense)             (None, 32)                544

 dropout (Dropout)           (None, 32)                0

 dense_2 (Dense)             (None, 16)                528

 dropout_1 (Dropout)         (None, 16)                0

 dense_3 (Dense)             (None, 5)                 85

=================================================================
Total params: 1,205
Trainable params: 1,205
Non-trainable params: 0
```

Accuracy and loss plots:





Note training data split 80-20 % for train and validation.

Classification report:

```
ANN accuracy_score:  0.8177128116938951
              precision    recall  f1-score   support

           0       0.81      0.67      0.73       233
           1       0.80      1.00      0.89       232
           2       0.85      0.88      0.87       233
           3       0.72      0.59      0.65       232
           4       0.89      0.94      0.91       233

   micro avg       0.82      0.82      0.82      1163
   macro avg       0.81      0.82      0.81      1163
weighted avg       0.81      0.82      0.81      1163
 samples avg       0.82      0.82      0.82      1163
```

Please note due to shortage of time, hyper-parameter studies were skipped.

Prediction of unlabeled samples

Since KNN on balanced data provided the highest accuracy, it was selected and trained on all the balanced training data to take advantage of maximum information. Then the model used to predict physical form of 160 unlabeled samples. The script saves the result in a csv file format in the root, portion of that is shown below.

	A	B	C	D
1	sample	bulk_density	particle_density	finalform
2	02-03325	-0.993625672	-0.182158488	Powder
3	02-08391	0.454898818	-1.29004415	Pellets
4	03-03680	-1.456348773	-0.927990012	Granular
5	03-08660	1.098687481	0.418851381	Pellets
6	03-08761	-1.536822356	-0.022854667	Powder
7	03-08762	-1.214928025	-0.037336833	Powder
8	03-16541	-0.812560111	-1.601410709	Granular
9	03-18590	0.957858711	-1.528999881	Pellets
10	03-25031	-1.919071874	-1.442106888	Powder

Conclusion

In this study, the original dataset restructured to a dataset with two features for each entry, bulk density and particle density. For entries with more than one value for each feature the average values was considered. From the restructured dataset, entries with known labels were stored in a training data frame (dt_train) and those with unknown labels were saved in a testing data frame (df_test).

One classic model (KNN) and one neural network model were used. KNN trained and validated on imbalanced and balanced data. Minority class oversampled. It was noticed the model accuracy improved on the prediction of minority class. The balanced data were used on ANN model using Keras API.

The KNN model selected due to its higher performance and trained on all training data to take advantage of all the information. The trained model was used to predict physical form test data, data without known labels.

Disclaimer

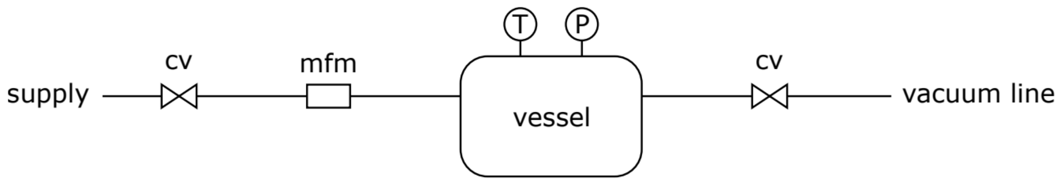
Most of the disclaimers below are due to my shortage of time and they would be improved if I had more time to spend on the assignment.

- More AI models could be tried and compared.
- While SMOTE is good oversampling method to learn the topological properties of the neighborhood, due to few samples of Cast and Blown compared to the Granular form, a combination method of under-sampling the Granular and over-sampling the Cast and Blown forms could provide a better generalization.
- In the training-validation split of the training data, a cross-validation method would provide more representative value.
- Hyper-parameter optimization could be performed on both models through a grid search.
- The code could be more organized and follow an object-oriented approach.

Q2. Regression

Question

Batches of gas are accumulated in the vessel shown below. The gas passes through a control valve (cv) and a mass flow meter (mfm) on its way into the vessel, which is instrumented to measure temperature and pressure. The temperature and pressure vary as the vessel is filled, depending on the temperature and flow rate of the incoming gas. At the end of each batch, the outlet control valve is opened and the vessel is rapidly emptied into a vacuum line. The outlet valve is then closed so a new batch can be processed.



Operators can use the data from the mass flow meter to estimate the mass of gas accumulated in the vessel during a batch, but their current method is somewhat manual, so they don't do it very often. Instead, they have asked if it is possible to predict the mass of gas in the vessel at any moment in time based on the other process measurements. This will eliminate their manual analysis and give them a continuous indicator of the process.

Please use the process data in the file regression.csv to predict the mass of gas in the vessel for each of the following conditions:

• 50 °C and 800 kPa-a

• 100 °C and 2000 kPa-a

• 200 °C and 500 kPa-a

The measurement units in the process data are kg/s for flow rate, °C for temperature and kPa-a for absolute pressure.

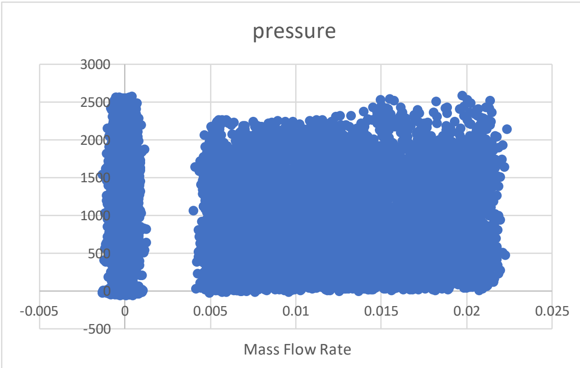
Solution

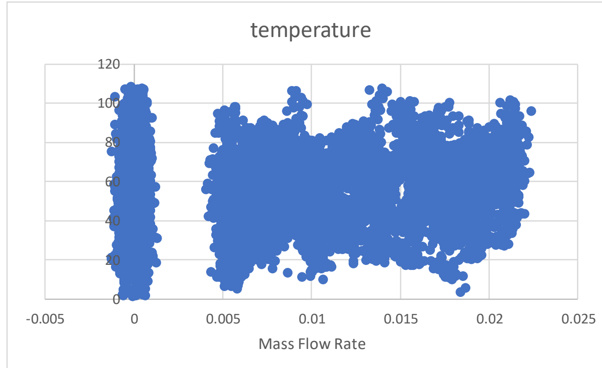
In this question, pressure and temperature are independent variables/predictors to predict mass flow as dependent variable. In the following, both classical and deep neural network models are trained and used for prediction of the mass of gas.

Classic Regressor's Training and Prediction

Data Pre-Processing

From the pressure and temperature plots below, clear patterns are not identified.





Regressor Models

The below regressor models trained on 80% of data and tested on 20%.

Model	Mean absolute error	R2 score
Linear Regression	0.0048	0.11
Polynomial Regression (n=3)	0.0043	0.18
KNeighbors Regressor	0.0042	0.12
Gradient Boosting Regressor	**0.0042**	**0.21**
Extra Trees Regressor	0.0040	0.13
Random Forest Regressor	0.0039	0.17
Ridge	0.0048	0.11

Coefficient of determination of above regressors indicates that the regressor are not powerful enough to accurately model the data.

Regressor Prediction of three inquired conditions

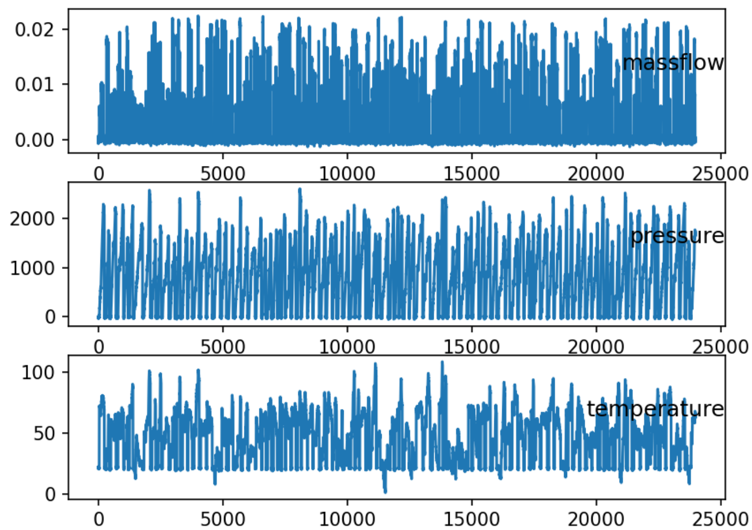
Prediction of Gradient Boosting Regressor model for the three conditions are shown below.

Pressure (Kpa)	Temperature (C)	Predicted mass flow rate (Kg/s)
800	50	0.00578625
2000	100	0.01163568
500	200	0.01513317

Neural Network Training and Prediction

Data Pre-Processing

With consideration of timestamps, the pressure, temperature and mass flow are plotted as shown in the following.



Data is first MinMax scaled between 0 and 1 and then is reframed for the LSTM, as shown below.

Original dataset

	massflow	pressure	temperature
2022-04-04 09:00:00	-0.00016	-15.19360	21.69464
2022-04-04 09:01:00	-0.00069	13.76619	21.44937
2022-04-04 09:02:00	0.00023	-3.35813	21.48880
2022-04-04 09:03:00	-0.00028	9.46699	20.74979
2022-04-04 09:04:00	0.00044	13.08061	20.59433

Scaled dataset

	0	1	2
0	0.04721	0.01611	0.19262
1	0.02475	0.02702	0.19034
2	0.06372	0.02057	0.19070
3	0.04210	0.02540	0.18381
4	0.07256	0.02676	0.18236

Reframed dataset

	massflow(t-1)	pressure(t-1)	temperature(t-1)	massflow(t)
4/4/2022 9:00	0.047215	0.016106	0.192625	0.024747
4/4/2022 9:01	0.024747	0.027022	0.190337	0.063724
4/4/2022 9:02	0.063724	0.020567	0.190705	0.042103
4/4/2022 9:03	0.042103	0.025401	0.18381	0.072563
4/4/2022 9:04	0.072563	0.026764	0.18236	0.056765

The data is from 4/4/2022 to 4/21/2022, including 23,977 entries. The first 80% of dataset is used for training and the rest for testing.

Long Short-Term Memory (LSTM) Model

It is thought that consideration of time sequence accompanying with the features and mass flow may bring added value to the data to capture patterns that could only be considered when long-term dependencies are considered, even at a level of one timestamp. For this reason, LSTM model is planned to be used.

The plan is to frame a supervised learning problem to predict the mass flow rate at the current timestamp given the temperature and pressure measurement conditions at the prior time step.

In the data pre-processing section, it was explained that dataset is reframed as shown below to be used in LSTM mode.

	massflow(t-1)	pressure(t-1)	temperature(t-1)	massflow(t)
4/4/2022 9:00	0.047215	0.016106	0.192625	0.024747
4/4/2022 9:01	0.024747	0.027022	0.190337	0.063724
4/4/2022 9:02	0.063724	0.020567	0.190705	0.042103
4/4/2022 9:03	0.042103	0.025401	0.18381	0.072563
4/4/2022 9:04	0.072563	0.026764	0.18236	0.056765

The data is from 4/4/2022 to 4/21/2022, including 23,977 entries. The first 80% of dataset is used for training and the rest for testing. Below shows the format ready for the LSTM model.

train_X.shape = (19181, 1, 3) → (number of samples, time window, features)

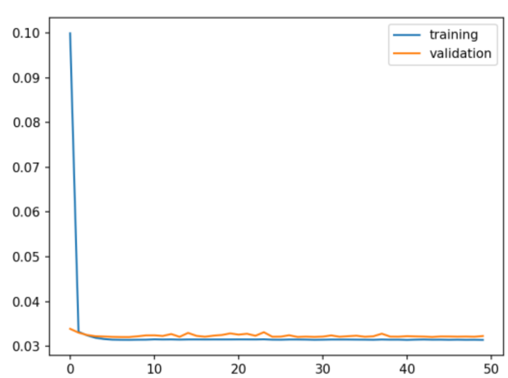
train_y.shape = (19181,)

test_X.shape = (4796, 1, 3)

test_y.shape = (4796,)

Model Fitting

LSTM is defined with 50 neurons in the first hidden layer and 1 neuron in the output layer for predicting mass flow rate. The input shape will be 1 time step with 3 features. Mean Absolute Error (MAE) loss function and the Adam version of stochastic gradient descent are used. The model will be fit for 50 training epochs with a batch size of 72. Training and test loss are tracked and plotted.



Model Evaluation

We combine the forecast with the test dataset is forecasted and invert scaled to bring the mass flow rate to its original scale. For the error score, Root Mean Squared Error (RMSE) is calculated in the same units as the variable itself.

RMSE: 0.003

R2 score: 0.84

LSTM Prediction of three inquired conditions

Average mass flow at (t-1)

Prediction of LSTM model for the three conditions are shown below. Note since mass flow rate at (t-1) has not been provided in the questions, the average mass flow rate of all entries, which is 0.004323049 kg/s, is used and considered the mass flow at previous timestamp. .

Mass flow rate at (t-1) (Kg/s)	Pressure (Kpa)	Temperature (C)	**Predicted mass flow rate (Kg/s)**
0.004323049	800	50	**-0.01417813**
0.004323049	2000	100	**-0.01418319**
0.004323049	500	200	**-0.02655255**

Zero mass flow at (t-1)

Below it is assumed that there is no mass flow rate at the previous timestamp.

Mass flow rate at (t-1) (Kg/s)	Pressure (Kpa)	Temperature (C)	**Predicted mass flow rate (Kg/s)**
0	800	50	**-0.02811318**
0	2000	100	**-0.0284315**
0	500	200	**-0.01638901**

Discussion/Question

By looking at the original dataset, it is noticed that for two similar entries, shown below for pressure = 800 kpa and temperature = 50 C, the mass flow rates have different values. This might be an indication that timestamp's context is an important factor to predict the mass flow rate.

	massflow	pressure	temperature	ambient_temperature
4/6/2022 6:07	0.005274506	745.78754	47.549377	20.673798
4/6/2022 6:08	0.005205465	766.21466	48.80896	19.775085
4/6/2022 6:09	0.004917877	762.5574	50.13452	18.882538
4/6/2022 6:10	0.005618066	798.28705	50.010437	20.50235
4/6/2022 6:11	0.005116373	819.2651	50.53299	19.924652
4/6/2022 6:12	0.004993486	810.2992	51.24533	19.817108
4/6/2022 6:13	0.005433605	843.45636	51.934418	19.946869
4/7/2022 11:00	0.000304618	808.74493	51.97293	21.24646
4/7/2022 11:01	-0.000202911	805.29675	51.82846	21.629211
4/7/2022 11:02	-0.000376684	795.16644	51.052856	18.727966
4/7/2022 11:03	-0.000116405	803.13245	50.434753	21.434906
4/7/2022 11:04	-0.000514135	828.5533	49.91336	20.60846
4/7/2022 11:05	0.016964179	845.4948	52.044464	21.15982

While it should be a direct relation between mass flow rate and pressure in a constant temperature, it is questionable how come there are examples that the trend does not follow the direct relation pattern. For example, below illustrates an example where temperature is 50C and when pressure goes lower or higher that 798 Kpa, the mass flow rate in both cases are decreased!

	massflow	pressure	temperature	ambient_temperature
4/6/2022 6:09	0.004917877	762.5574	50.13452	18.882538
4/6/2022 6:10	0.005618066	798.28705	50.010437	20.50235
4/6/2022 6:11	0.005116373	819.2651	50.53299	19.924652

Conclusion

The following two approached was tried.

(1) Using regressor models to train and predict the mass flow rate from pressure and temperature without consideration of the impact of the timestamps.
(2) Utilizing LSTM model to train and predict the mass flow rate at time t from mass flow rate, pressure, and temperature at time (t-1).

In the regressor approach, multiple regressor model tested and it was noticed regressor models are not powerful enough to accurately model and predict the data. The best coefficient of determination belongs to Gradient Boosting model with R2 score = 0.21

LSTM recurrent approach demonstrated a better performance with RMSE: 0.003 and **R2 score = 0.84**. For the prediction of the inquired gas conditions, refer to the respective sections.

Thank you for the interesting questions,

Ryan