CS 668 PROJECT
Ventilator Pressure Detection

Sunanda Singareddy, Data Science, sunandasreddy@gmail.com
https://github.com/Sunanda1710

Abstract

The COVID-19 pandemic statistic count is 146,122 cases per million people and there is an incremental trend across the globe.

This project allows me to address this situation and automatically predict the right level flow of air pressure based on the actual pressure values.

As a result, predicting the pressure on ventilation before hand will help in increasing survival rates. The early estimate on **recovery rate is 97 to 99.5%**

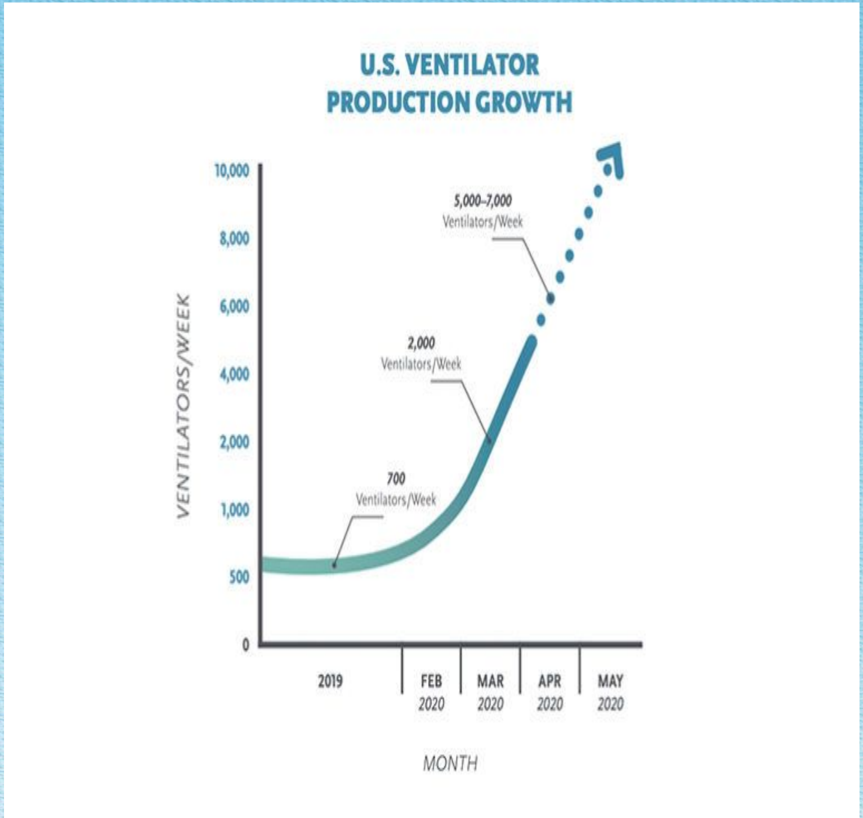


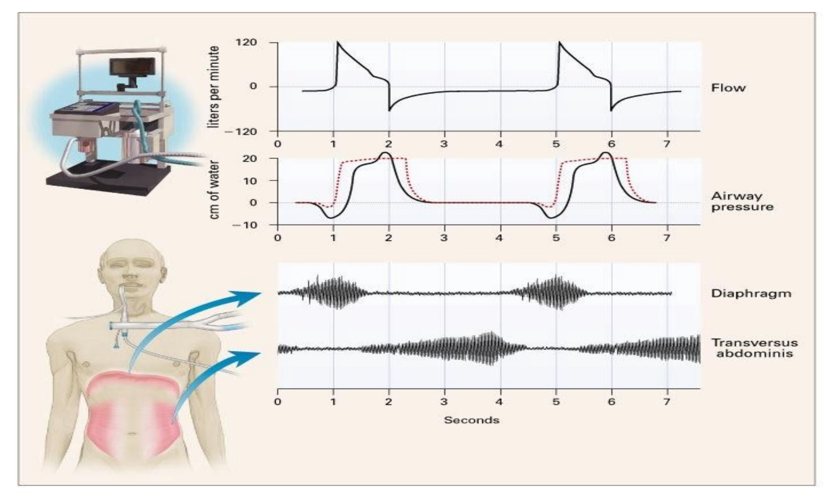
Image Reference : https://www.massdevice.com/u-s-ventilator-manufacturing-is-rapidly-expanding-heres-how/

Introduction

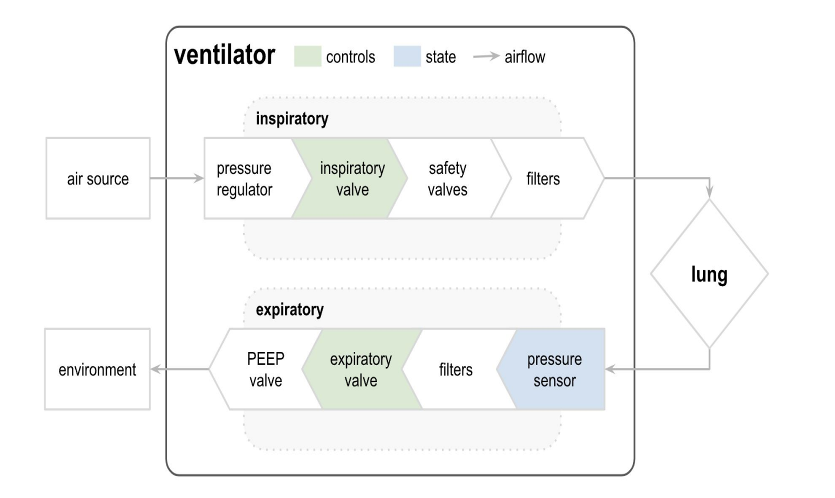
Covid-19 is the respiratory disease which have an impact on respiratory tract including lungs.

Lungs and airways swell and become inflamed and this infection starts in one part of lung and spreads to the other.

As the body starts to fight it, the lungs become more inflated and fill with fluid which can make it harder for patients to swap oxygen and carbon dioxide



DOI: https://doi.org/10.4187/respcare.07404



https://www.kaggle.com/c/ventilator-pressure-prediction/data

A ventilator mechanically helps pump oxygen into body and can be set to take a certain number of breaths per minute.

It can also hold positive end-expiratory pressure (PEEP) - constant amount of low pressure to keep air sacs in lung from collapsing.

The pressure plays a vital role and it is dependent on features such as R (resistance through with air is passed), C (compliance of volume per pressure).

Prediction of pressure must not be beyond the PIP (target pressure) and must not be below PEEP (positive end-expiratory pressure) value.

Personal Motivation

- ❖ Help doctors by automatically making the system predict the pressure required, and save Covid-19 patients from death.
- ❖ Gain ample amount of knowledge in Data Analysis, Data Visualization and get good job in Data Science field.
- ❖ Compare various models and their performance, accuracy rate and check which model is better fit.
- ❖ The final score calculated on the Kaggle competition website must be nearly equal to those on top of the leaderboard.

Literature Review : Introduction

I have done my research on Ventilator Pressure Prediction connected to a sedated patient's lung. The ground work includes, checking for the best predictive model available for pressure prediction and enhance it further, so that the system will adapt itself and automatically predict the right level of pressure through respiratory circuit.

Research Question:

What is the predicted pressure to pass through the respiratory circuit so that the patient can survive on the ventilator?

Why I chose this research question?

> It will increase the chance of patients to survive who are on ventilator during Covid-19 pandemic situation.
> Expand my knowledge of enhancing the performance of model and use this expertise in various domain.

Summary of the Literature Review

In research papers published by authors we can see the Machine Learning models used are:

For Classification:

> Artificial Neural Network (ANN)
> Decision Trees (Bootstrap Aggregation)

For Regression:

> Ensemble Model : XGBoost, LightGBM
> Gradient Boosting : catBoost
> Recurrent Neural Network - LSTM Model
> Random Forest
> Linear Regression

Optimization:

The optimization used in the research paper is k-fold cross validation for enhancing the performance of model.

Research Papers Conclusion: **Ensemble or Boosting** is the best predictive model

Data

Dataset: https://www.kaggle.com/c/ventilator-pressure-prediction/data

Dataset Description:

- id - unique time step (Globally)
- breath_id - unique time step for breaths (Globally)
- R - lung attribute indicating resistance (in cmH2O/L/S).
- C - lung attribute indicating compliance (in mL/cmH2O).
- time_step - actual time stamp.
- u_in - the control input for the inspiratory solenoid valve. Ranges from 0 to 100.
- u_out - the control input for the exploratory solenoid valve. Either 0 or 1.
- pressure - the airway pressure measured in the respiratory circuit, measured in cmH2O.

Dataset Rows and Columns: **6 036 000** rows × 8 columns

Train Dataset Datatype

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6036000 entries, 0 to 6035999
Data columns (total 8 columns):
 #   Column     Dtype
---  ------     -----
 0   id         int64
 1   breath_id  int64
 2   R          int64
 3   C          int64
 4   time_step  float64
 5   u_in       float64
 6   u_out      int64
 7   pressure   float64
dtypes: float64(3), int64(5)
memory usage: 368.4 MB
None
```

Loading Train Dataset

[2]:
```
Train_set = pd.read_csv('train.csv')
display(HTML('<p style="font-size: 12px;"">Train dataset contains below rows and columns</b></p>'))
print(Train_set.shape,'\n')
display(HTML('<p style="font-size: 12px;"">Top 5 rows of Train Dataset</b></p>'))
print(Train_set.head())
```

Train dataset contains below rows and columns

```
(6036000, 8)
```

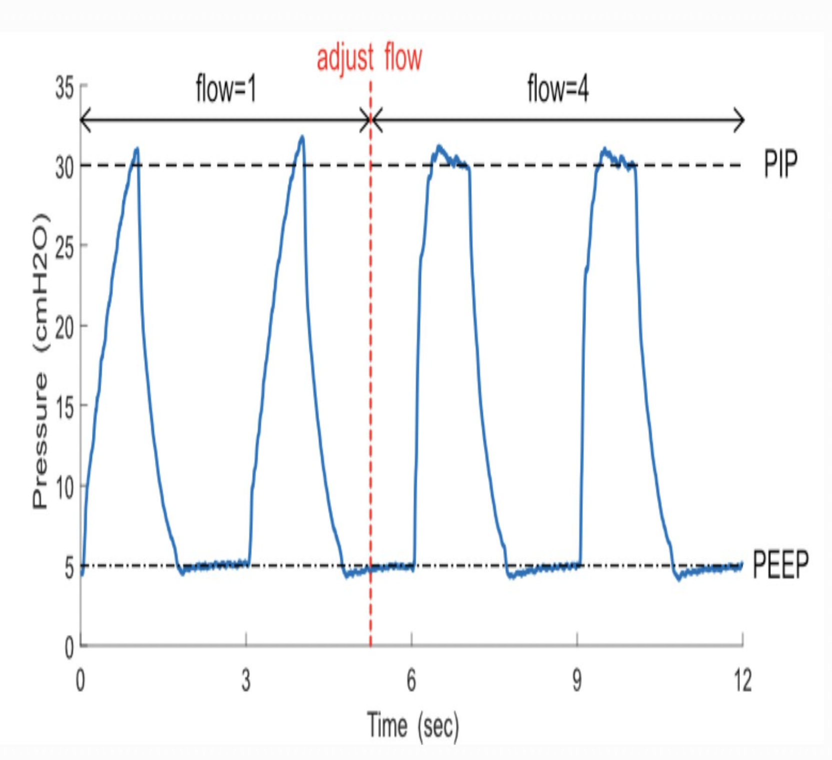
Top 5 rows of Train Dataset

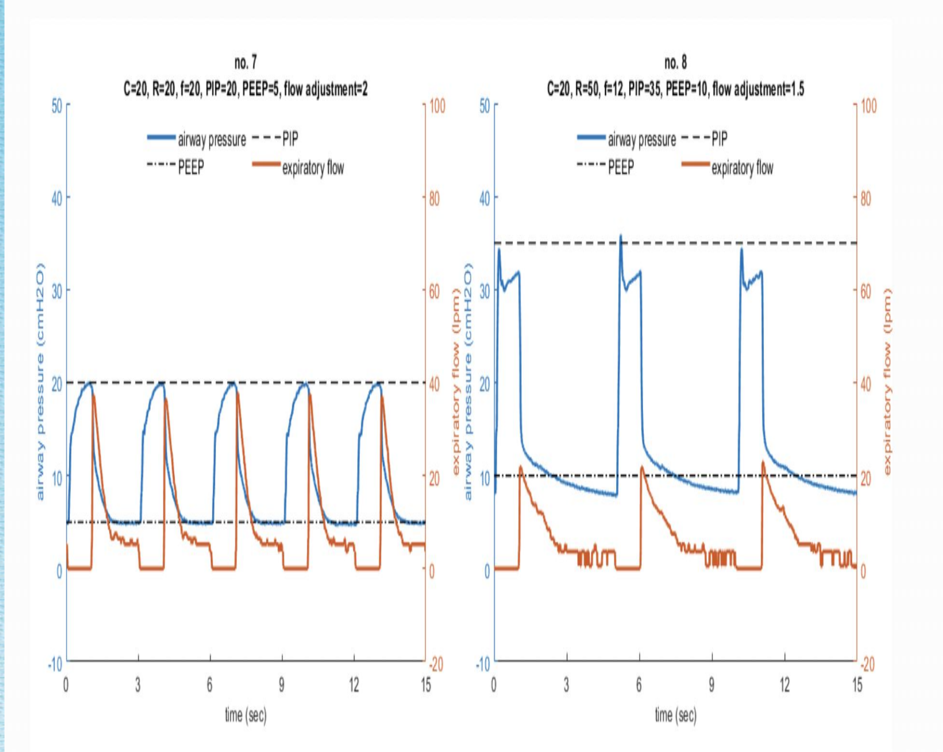
	id	breath_id	R	C	time_step	u_in	u_out	pressure
0	1	1	20	50	0.000000	0.083334	0	5.837492
1	2	1	20	50	0.033652	18.383041	0	5.907794
2	3	1	20	50	0.067514	22.509278	0	7.876254
3	4	1	20	50	0.101542	22.808822	0	11.742872
4	5	1	20	50	0.135756	25.355850	0	12.234987

[7]:
```
Train_set.isna().any()
```

[7]:
```
id           False
breath_id    False
R            False
C            False
time_step    False
u_in         False
u_out        False
pressure     False
dtype: bool
```

Data Description Continued...





Problems / Issues

- **Problem 1:** *<u>Relationship</u>* has to be established in the dataset provided between time step, resistance, compliance and breaths.

- **Problem 2:** The dataset has time step information which is a time-series data and *<u>feature engineering</u>* has to be done in order to get accurate results.

- **Problem 3:** the features provided are minimal, and hence *<u>additional features</u>* to be created using Pandas for checking various combinations between R and C where R feature have settings as R=5, 20, and 50 cm H2O/L/s and C feature have settings C=5, 20, and 50 mL cm H2O

Data Preparation

Fig1: Analyzing the dataset

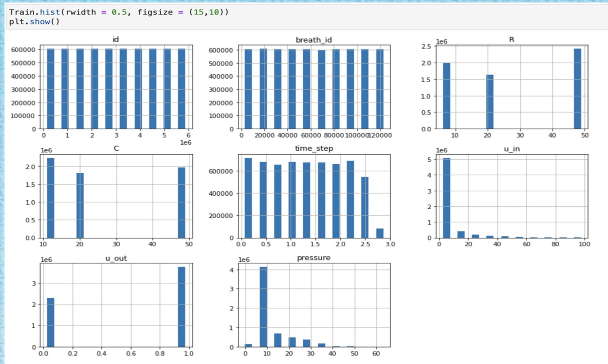
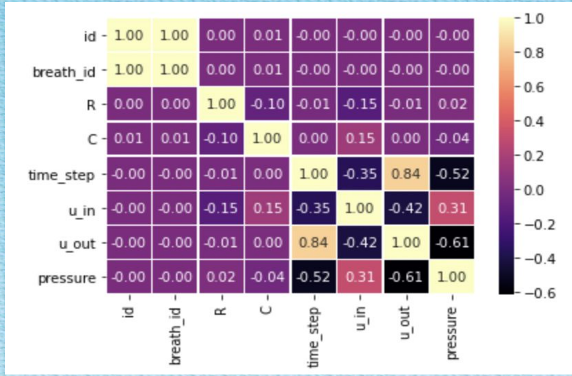


Fig2: Check in case there are any missing values or any null values

```
[7]: Train_set.isna().any()

[7]: id            False
     breath_id     False
     R             False
     C             False
     time_step     False
     u_in          False
     u_out         False
     pressure      False
     dtype: bool
```

Fig3: Check the correlation in features amongst itself for feature selection



Conclusion:

- we can see thaFrom histogram in fig1 t id and breath id have the unique values, R and C values are 5, 20, 50, input value varies between 0 to 100 and u_out value is either 0 or 1
- In fig2, a check is performed to see if any missing values or null values are available. The result says false, meaning no attributes have null values
- Correlation in features are checked amongst itself and all the features are highly correlated and hence no attributes are dropped.

Experimentation

Linear Regression:

The model score given by Linear Regression is 0.38 which is not a best model for prediction.

```
score=r2_score(y_test,y_prediction)
print('r2 socre is',score)
```

```
r2 socre is 0.3834632804905116
```

```
print(LR.score(x_train, y_train))
```

```
0.38415233773264024
```

```
print(LR.score(x_test, y_test))
```

```
0.3834632804905116
```

Get the score for the model fit ¶

```
[12]: regressor.score(X_test, y_test)#0.7328009958642798 (
      #0.7340095203341319 (randomstate 0)
      #0.7347170806238494(sample 10400)
      #0.7355095986669042(sample 11000)
      #0.7386705994342928
      #0.7413771292353316
```

```
[12]: 0.7411017900742535
```

Random Forest:

The model score given by RF is 0.74 which is still less to say a best predictor but better than Linear Regression.

Experimentation Continued

XGBoost:

The model score given by XGBoost is 0.74 but the model performance if GPU is not used then it is poor. (Note: I was not able to improve as I was not able to use system GPU)

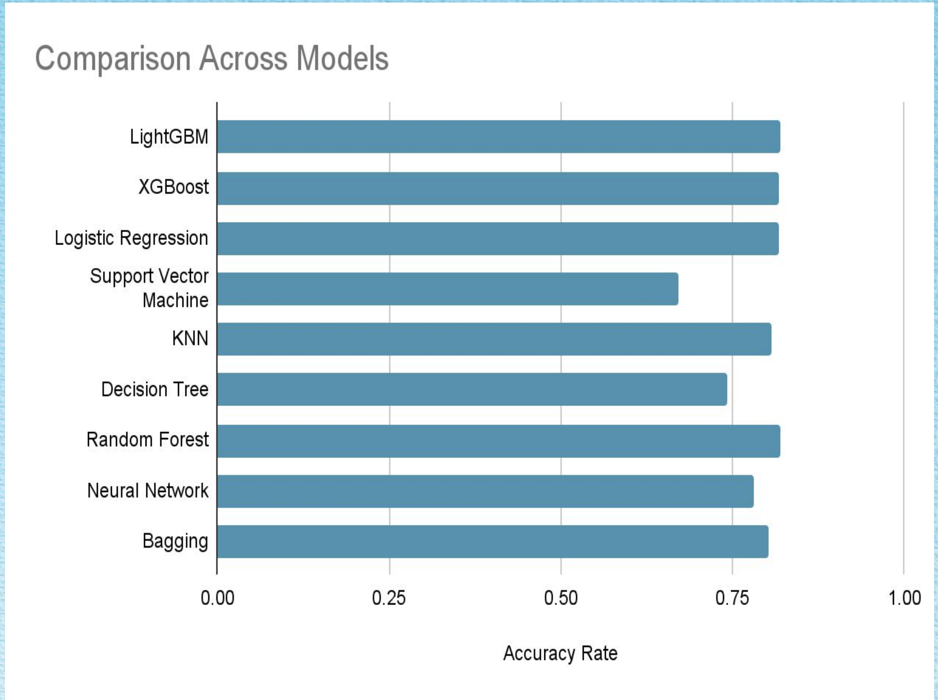
```
[29]: %time regressor.fit(X_train, y_train)

CPU times: user 19min 46s, sys: 49.7 s, total: 20min 36s
Wall time: 2min 40s

[29]: XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
                   colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
                   importance_type='gain', interaction_constraints='',
                   learning_rate=0.9, max_delta_step=0, max_depth=7,
                   min_child_weight=1, missing=nan, monotone_constraints='()',
                   n_estimators=100, n_jobs=8, num_parallel_tree=1, random_state=42,
                   reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
                   tree_method='approx', validate_parameters=1, verbosity=None)

[30]: print('Training accuracy {:.4f}'.format(regressor.score(X_train,y_train)))
      print('Testing accuracy {:.4f}'.format(regressor.score(X_test,y_test)))

Training accuracy 0.7450
Testing accuracy 0.7415
```

```
37]: model = lgb.LGBMRegressor(learning_rate=0.09,max_depth=5,random_state=42)
     model.fit(x_train,y_train,eval_set=[(x_test,y_test),(x_train,y_train)], verbose=20)
     #LGBMRegressor(boosting_type='gbdt', class_weight=None, colsample_bytree=1.0,
      #      importance_type='split', learning_rate=0.1, max_depth=-1,
       #     min_child_samples=20, min_child_weight=0.001, min_split_gain=0.0,
        #    n_estimators=100, n_jobs=-1, num_leaves=31, objective=None,
         #   random_state=None, reg_alpha=0.0, reg_lambda=0.0, silent=True,
          # subsample=1.0, subsample_for_bin=200000, subsample_freq=0)

     [20]    training's l2: 20.03     valid_0's l2: 20.0408
     [40]    training's l2: 17.5855   valid_0's l2: 17.6013
     [60]    training's l2: 17.0731   valid_0's l2: 17.0915
     [80]    training's l2: 16.7872   valid_0's l2: 16.8069
     [100]   training's l2: 16.5995   valid_0's l2: 16.6229

37]: LGBMRegressor(learning_rate=0.09, max_depth=-5, random_state=42)

38]: print('Training accuracy {:.4f}'.format(model.score(x_train,y_train)))
     print('Testing accuracy {:.4f}'.format(model.score(x_test,y_test)))

     Training accuracy 0.7476
     Testing accuracy 0.7472
```

LightGBM:

This model gives the better accuracy than XGBoost and the performance is better so further analysis and predictions is done using LightGBM Ensemble Model in this project.

Methodology



Results from the literature

- ➢ Pressure is a continuous variable and hence for prediction **regression** is used instead of classification.
- ➢ Also, **ensemble model** is better in terms of prediction as it combines the prediction from multiple models.
- ➢ The bar chart is a consolidated report from the results obtained from literature review.
- ➢ **LightGBM and XGBoost** provide the best accuracy rate as compared to other models.

Train and Test Accuracy Comparison between Models

Model	Parameters	Train Accuracy	Test Accuracy
Linear Regression	Random state = 42	38.41	38.34
Random Forest	Random state = 42, Sample = 11000	74.11	73.89
XGBoost	Random state = 42, learning_rate = 0.9	74.50	74.15
LightGBM	Random state = 42, learning_rate = 0.9	74.76	74.72
LightGBM with Added features (Categorical Values)	**Random state = 42, learning_rate = 0.25 Feature Engineering of time series**	**98.58**	**98.51**

Results

LightGBM with features added:

Additional features are added using feature engineering and also converted R (5, 20, 50) and C (5, 20, 50) to numerical values. This improvised the performance and the test and train score are nearly equivalent to 98%

```
[39]: model = lgb.LGBMRegressor(learning_rate=0.35,max_depth=27,random_state=42, num_leaves = 106)
      model.fit(x_train,y_train,eval_set=[(x_test,y_test),(x_train,y_train)], verbose=30)

      [30]    training's l2: 1.34234  valid_0's l2: 1.368
      [60]    training's l2: 1.08865  valid_0's l2: 1.12528
      [90]    training's l2: 0.970549 valid_0's l2: 1.01532

[39]: LGBMRegressor(learning_rate=0.35, max_depth=27, num_leaves=106, random_state=42)

[40]: print('Training accuracy {:.4f}'.format(model.score(x_train,y_train)))
      print('Testing accuracy {:.4f}'.format(model.score(x_test,y_test)))
      #Training accuracy 0.9655 5 depth and 0.09 and 0.33 test size
      #Testing accuracy 0.9653
      #Training accuracy 0.9711 0.10 and 7 depth and 0.35
      #Testing accuracy 0.9709
      #Training accuracy 0.9741 0.15 and 7
      #Testing accuracy 0.9739
      #Training accuracy 0.9748 11 depth 0.15
      #Testing accuracy 0.9746
      #Training accuracy 0.9793 num leaves 56
      #Testing accuracy 0.9790
      #Training accuracy 0.9840 rate 0.25 depth 13 leaves 86
      #Testing accuracy 0.9835

      Training accuracy 0.9858
      Testing accuracy 0.9851
```

Kaggle Results

Overview Data Code Discussion Leaderboard Rules Team My Submissions **Submit Predictions** ...

8 submissions for Sunanda Reddy

Sort by Select...

All Successful Selected

Submission and Description	Public Score	Use for Final Score
Submission_LightGBM_features (2).csv 2 hours ago by Sunanda Reddy add submission details	1.1323	☐
Submission_LightGBM_features (1).csv 3 hours ago by Sunanda Reddy add submission details	1.4368	☐
Submission_LightGBM_features.csv 14 days ago by Sunanda Reddy add submission details	1.5968	☐
submission_1.csv 15 days ago by Sunanda Reddy Testing on Data	0.1502	☐
submission_LightGBM2.csv 16 days ago by Sunanda Reddy LightGBM Modification	4.1764	☐
Submission_LightGBM.csv 16 days ago by Sunanda Reddy LightGBM method	4.2082	☐

#	Team Name	Notebook	Team Members	Score	Entries	Last
1	ryomak			0.1095	62	7h
2	AmbrosM			0.1118	25	3h
3	Shujun, Kha, Zidmie, Gilles, B			0.1119	137	1d
4	waiwai			0.1129	104	4h
5	Y.Z.S.C			0.1138	165	7h
333	Kris			0.150	4	2d
334	Sunanda Reddy			0.150	5	1d
335	Jonver Oro			0.150	2	1d
336	ASHFAQUE			0.150	25	20h
337	Kuante			0.150	9	12h

Your Best Entry ↑

Your submission scored 0.150, which is an improvement of your previous score of 4.176. Great job!

Tweet this!

Discussion & Conclusion

- Predicting the pressure on ventilation before hand will help in increasing survival rates. The early estimate on recovery rate is 97 to 99.5%

- To avoid hyperventilation state of patient the prediction of pressure and the flow rate must be accurate.

- LightGBM is the best model in regression as compared to other models for predicting continuous variables.

Future Work

Enhance LightGBM features to improvise prediction. (My current score: 0.1502)

Use gridwise search parameters for boosting the accuracy percentage and optimize the model.

The team who won the competition with 0.0575 score, have used LSTM (Neural Network Model)

References

- Yu L, Halalau A, Dalal B, Abbas AE, Ivascu F, Amin M, et al. (2021) Machine learning methods to predict mechanical ventilation and mortality in patients with COVID-19. PLoS ONE 16(4): e0249285. https://doi.org/10.1371/journal.pone.0249285
- Ghazal, Sam & Sauthier, Michaël & Brossier, David & Bouachir, Wassim & Jouvet, Philippe & Noumeir, Rita. (2019). Using machine learning models to predict oxygen saturation following ventilator support adjustment in critically ill children: A single center pilot study. PLOS ONE. 14. e0198921. 10.1371/journal.pone.0198921. **https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0198921**
- Mamandipoor, B., Frutos-Vivar, F., Peñuelas, O. et al. Machine learning predicts mortality based on analysis of ventilation parameters of critically ill patients: multi-centre validation. BMC Med Inform Decis Mak 21, 152 (2021). https://doi.org/10.1186/s12911-021-01506-w
- T. Chen et al., "Prediction of Extubation Failure for Intensive Care Unit Patients Using Light Gradient Boosting Machine," in IEEE Access, vol. 7, pp. 150960-150968, 2019, doi: 10.1109/ACCESS.2019.2946980. **https://www.researchgate.net/publication/336453098_Prediction_of_Extubation_Failure_for_Intensive_Care_Unit_Patients_Using_Light_Gradient_Boosting_Machine**
- Sayed, M.; Riaño, D.; Villar, J. Predicting Duration of Mechanical Ventilation in Acute Respiratory Distress Syndrome Using Supervised Machine Learning. J. Clin. Med. 2021, 10, 3824. https://doi.org/10.3390/jcm10173824
- Zhu Y, Zhang J, Wang G, Yao R, Ren C, Chen G, Jin X, Guo J, Liu S, Zheng H, Chen Y, Guo Q, Li L, Du B, Xi X, Li W, Huang H, Li Y and Yu Q (2021) Machine Learning Prediction Models for Mechanically Ventilated Patients: Analyses of the MIMIC-III Database. Front. Med. 8:662340. doi: 10.3389/fmed.2021.662340 https://doi.org/10.3389/fmed.2021.662340
- Zhang, Zhongheng MD1; Liu, Jingtao MD2; Xi, Jingjing MD3; Gong, Yichun MD2; Zeng, Lin PhD4; Ma, Penglin MD2 Derivation and Validation of an Ensemble Model for the Prediction of Agitation in Mechanically Ventilated Patients Maintained Under Light Sedation, Critical Care Medicine: March 2021 - Volume 49 - Issue 3 - p e279-e290 doi: 10.1097/CCM.0000000000004821 https://pubmed.ncbi.nlm.nih.gov/33470778/
- Zhang, Z, Navarese, EP, Zheng, B, et al. Analytics with artificial intelligence to advance the treatment of acute respiratory distress syndrome. J Evid Based Med. 2020; 13: 301– 312. https://doi.org/10.1111/jebm.12418
- Raita, Y., Camargo, C.A., Macias, C.G. et al. Machine learning-based prediction of acute severity in infants hospitalized for bronchiolitis: a multicenter prospective study. Sci Rep 10, 10979 (2020). https://doi.org/10.1038/s41598-020-67629-8

Feedback

- Your justifications for each model you're using are very strong - Brian
- Your presentations and explanation of the topic was really good - Prachi
- Presentation was well prepared and presented. I like the motivation you gave first which gave a strong point on you doing the project. - Shefali