edX Capstone Unbalanced Blades Submittal

Eduardo Caballero

3/5/2022

Executive Summary

When troubleshooting a mechanical system it is often helpful to have an idea of where the problem may be occuring. This project attempts to utilize machine learning to determine the type of imbalance that has occured on a spinning fan blade.

The data utilized was from a study "Prediction of Motor Failure Time Using An Artificial Neural Network" , doi:10.3390/s19194342, https://www.mdpi.com/1424-8220/19/19/4342 done by Scalabrini Sampaio, Gustavo, Vallim Filho, Arnaldo Rabello de Aguiar, Santos da Silva, Leilton and Augusto da Silva, Leandro.

In the study magnetic weights were placed on the blades to create an imbalance. 3 configurations were utilized 1. the magnets were placed on adjacent blades, 2 the magnets placed on blades 90 degs apart and 3. magnets placed on blades 180 degrees from each other.

Vibration data was collected from the fan at varying rpms.The data can be found at https://archive.ics.uci.edu/ml/machine-learning-databases/00611/accelerometer.csv.

By using the rpms that the fan is spinning at and the vibration levels caused by the imbalance, a model was created to predict the type of imbalance in the fan blades. This would be useful for technicians to know so that they can find the blades that are imbalanced much quicker.

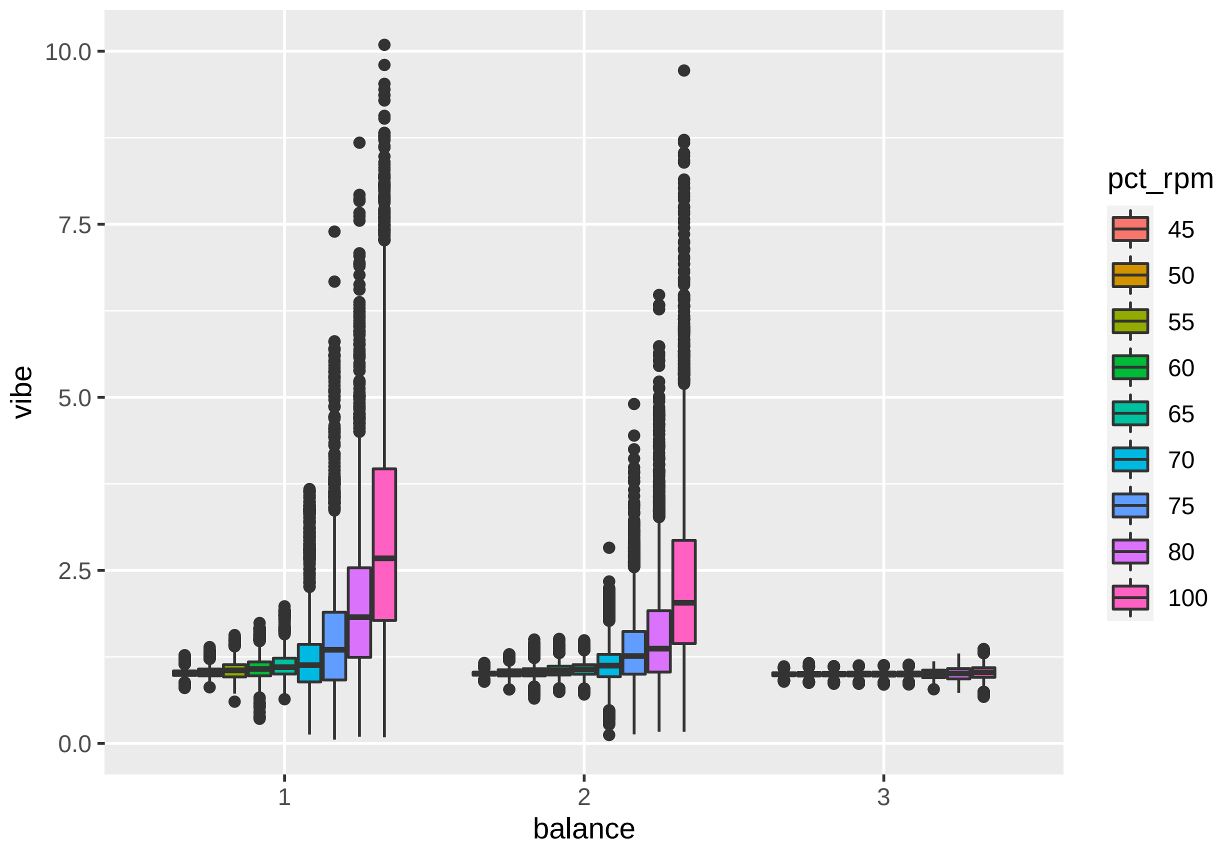
Various models were tested using the caret library. Models tested included knn, random forest, rpart and ranger. The root mean squared error (RMSE) was utilized to determine the best model. The knn model gave the best results of the models tested.

Analysis

The downloaded data was reviewed. It consisted of 5 variables. wconfid - an integer either 1,2 or 3 that represented the type of blade imbalance. pctid - an integer giving the percent of max rpms that the fan was spinning at x, y, z - vibration levels

3 additional columns were added to the data to aid in training and displaying graphics vibe - the magnitude of vibration calculated as sqrt(x^2 +y^2 +z^2) balance - conversion of wconfid to a factor instead of integer pct_rpm - conversion of pctid to a factor instead of an integer.

A boxplot of the data was done to see the distribution of vibration levels for varying rpm levels and balance configurations.



based on this graph it was determined to utilize an rpm level of 50% or greater since the vibration levels did not show much of a difference at slower speeds. This would mean that the fan would have to be brought up to 50% max speed for our prediction model to operate. When blades are unbalanced, it is best to try and keep the rpms below max level to ensure a dangerous failure does not occur.

The data was then divided into a validation set and a training/test set. 10% of the data was used for the validation set the remaining data was the divided into a train and a test set with a 80% 20% split.

Results

Various machine learning models were tested to determine which one gave the best predicted results. Models tested were knn, random forest, rpart and ranger. The predictors used to train the model were vibe and pct_rpm.

The knn model was run with cross validation. The model was run with the train record set. Once a trained model was established the model was tested on the test data set. The RMSE results were 0.632061 The model information is as shown below:

```
## k-Nearest Neighbors
##
## 71280 samples
##     2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 64152, 64152, 64152, 64152, 64152, 64152, ...
```

```
## Resampling results across tuning parameters:
##
##   k   RMSE       Rsquared   MAE
##   30  0.6281428  0.4080532  0.5304963
##   32  0.6274695  0.4092328  0.5302432
##   34  0.6269072  0.4102147  0.5300983
##   36  0.6264478  0.4110015  0.5301691
##   38  0.6260741  0.4116477  0.5301452
##   40  0.6255876  0.4125056  0.5299757
##   42  0.6252285  0.4131316  0.5299522
##   44  0.6249138  0.4136825  0.5299322
##   46  0.6246563  0.4141389  0.5298939
##   48  0.6243874  0.4146093  0.5299020
##   50  0.6240996  0.4151187  0.5298434
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 50.
```

The Random Forest model was run with cross validation. The RMSE results were 0.6346502

```
## Random Forest
##
## 71280 samples
##     2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 64152, 64152, 64152, 64152, 64152, 64152, ...
## Resampling results across tuning parameters:
##
##   mtry  RMSE       Rsquared   MAE
##    1    0.6268215  0.4122242  0.5452744
##    5    0.7127430  0.2958149  0.5492621
##   10    0.7131556  0.2949734  0.5502264
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was mtry = 1.
```

The rpart model was run with cross validation. The RMSE results were 0.637228. The model information is shown below.

```
## CART
##
## 71280 samples
##     2 predictor
##
## No pre-processing
## Resampling: Cross-Validated (10 fold)
## Summary of sample sizes: 64152, 64152, 64152, 64152, 64152, 64152, ...
## Resampling results across tuning parameters:
##
##   cp           RMSE       Rsquared   MAE
##   0.001072852  0.6288691  0.4059442  0.5445227
```

```
##   0.001103805  0.6290217  0.4056491  0.5449568
##   0.001170336  0.6294315  0.4048695  0.5456300
##   0.001300685  0.6303798  0.4030708  0.5475553
##   0.001663184  0.6317837  0.4004047  0.5492017
##   0.001880446  0.6332828  0.3975621  0.5521097
##   0.001905143  0.6341081  0.3959985  0.5537010
##   0.001960617  0.6350285  0.3942377  0.5547235
##   0.002121308  0.6368805  0.3907176  0.5569141
##   0.002675731  0.6385175  0.3875679  0.5598873
##   0.002779048  0.6393889  0.3859012  0.5618521
##   0.006639697  0.6422404  0.3804161  0.5646475
##   0.009066397  0.6473970  0.3704149  0.5682519
##   0.009352346  0.6506290  0.3641060  0.5720209
##   0.019377553  0.6602989  0.3449932  0.5848704
##   0.020267552  0.6729436  0.3196450  0.5973789
##   0.027359905  0.6883857  0.2880772  0.6092977
##   0.031706147  0.7058039  0.2516229  0.6199213
##   0.044014906  0.7235771  0.2134035  0.6445275
##   0.199021388  0.7743771  0.1934471  0.6636267
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was cp = 0.001072852.
```

The ranger model. The RMSE results were 0.6760426. The model information is shown below.

```
## Random Forest
##
## 71280 samples
##     2 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 71280, 71280, 71280, 71280, 71280, 71280, ...
## Resampling results across tuning parameters:
##
##   splitrule   RMSE      Rsquared   MAE
##   variance    0.733806  0.2751745  0.5544207
##   extratrees  0.703024  0.3055021  0.5477386
##
## Tuning parameter 'mtry' was held constant at a value of 2
## Tuning
##  parameter 'min.node.size' was held constant at a value of 5
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were mtry = 2, splitrule = extratrees
##  and min.node.size = 5.
```

Since the knn model ran the lowest RMSE in the test, it was run with the validation model and a final RMSE calculated at 0.6303947. The second lowest RMSE was random forest and it was also tested with the validation data which gave a RMSE of 0.6320287

Conclusion

Based on the models run the knn model performed the best. In order to make this model useful, additional tests would need to be run for different fans and different weight imbalances to determine if these results could be useful in predicting the imbalance in any fan or is it very specific for the fan that was used to gather the initial data. The more generalized these results can be, the more useful they would be in real world applications.