PURDUE UNIVERSITY | The Data Mine

Yamaha - KPVs of Manufacturing Precision Propellers

Aadit Sangwan, Caleb Cox, Dhruv Arora, Josiah Linneman, Kaiden Krenek and Abolee Diwate (TA)

YAMAHA Revs your Heart | Precision Propeller Industries, Inc.

Introduction

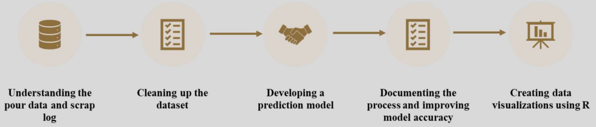
About Yamaha: Yamaha Precision Propellers is one of the leading investment casting propeller manufacturers in the United States.

Problem: The company aims at reducing the amount of product scraped to less than 2%. The successful to defective pour ratio at the start of the project was too high, due to investment casting being a tedious process with multiple variable factors.

Motivation: To create a data prediction model that would evaluate the data and predict which factors lead to a failed pour.

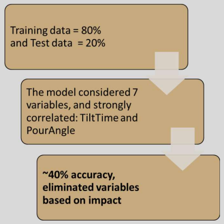
Goal: To clean the dataset and identify key factors impacting scrap production.

Project Workflow



Understanding the pour data and scrap log → Cleaning up the dataset → Developing a prediction model → Documenting the process and improving model accuracy → Creating data visualizations using R

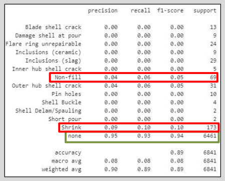
Decision Tree Approach



Training data = 80% and Test data = 20%

The model considered 7 variables, and strongly correlated: TiltTime and PourAngle

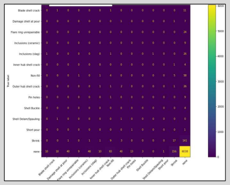
~40% accuracy, eliminated variables based on impact



```
                         precision    recall  f1-score   support

      Blade shell crack       0.00      0.00      0.00        13
    Damage shell at pour      0.00      0.00      0.00         9
Flare ring unrepairable       0.00      0.00      0.00        24
    Inclusions (ceramic)      0.00      0.00      0.00         9
      Inclusions (slag)       0.00      0.00      0.00        29
    Inner hub shell crack     0.00      0.00      0.00         5
               Non-fill       0.04      0.06      0.05        69
    Outer hub shell crack     0.04      0.06      0.05        31
              Pin holes       0.00      0.00      0.00        10
           Shell Buckle       0.00      0.00      0.00         4
    Shell Delam/Spauling      0.00      0.00      0.00         2
             Short pour       0.00      0.00      0.00         2
                 Shrink       0.09      0.10      0.10       173
                   none       0.95      0.93      0.94      6461

               accuracy                           0.89      6841
              macro avg       0.08      0.08      0.08      6841
           weighted avg       0.90      0.89      0.89      6841
```

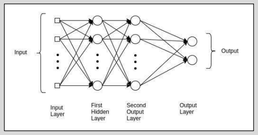


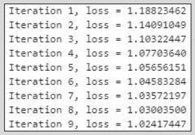
The number of **None** (no defects predicted) heavily outweighs the number of shrink and non-fill, hence the decision model algorithm struggled to find enough training data points.

Prediction Model Approach

MLP Classifier Model:

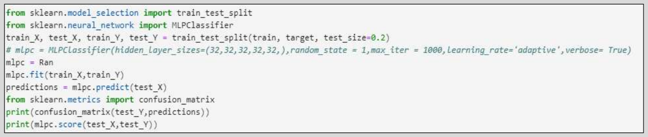
- Model based on a neural network that predicts qualitative labels using features that are both qualitative and quantitative features.
- Manipulates the weights toward features to find the best prediction routes.





```
Iteration 1, loss = 1.18823462
Iteration 2, loss = 1.14091049
Iteration 3, loss = 1.10322447
Iteration 4, loss = 1.07703640
Iteration 5, loss = 1.05656151
Iteration 6, loss = 1.04583284
Iteration 7, loss = 1.03572197
Iteration 8, loss = 1.03003500
Iteration 9, loss = 1.02417447
```

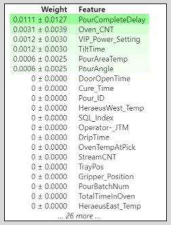
Python implementation



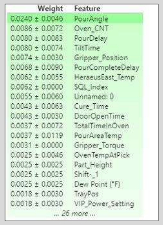
```
from sklearn.model_selection import train_test_split
from sklearn.neural_network import MLPClassifier
train_X, test_X, train_Y, test_Y = train_test_split(train, target, test_size=0.2)
# mlpc = MLPClassifier(hidden_layer_sizes=(32,32,32,32,32,),random_state = 1,max_iter = 1000,learning_rate='adaptive',verbose= True)
mlpc = Ran
mlpc.fit(train_X,train_Y)
predictions = mlpc.predict(test_X)
from sklearn.metrics import confusion_matrix
print(confusion_matrix(test_Y,predictions))
print(mlpc.score(test_X,test_Y))
```

Findings

- Implemented random forest algorithm multiple times and established permutation importance on the model.

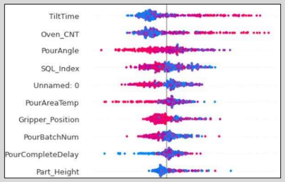


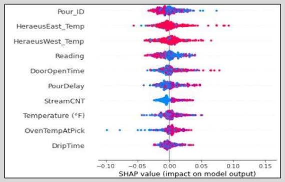
Weight	Feature
0.0111 ± 0.0127	PourCompleteDelay
0.0031 ± 0.0039	Oven_CNT
0.0012 ± 0.0030	VIP_Power_Setting
0.0012 ± 0.0030	TiltTime
0.0006 ± 0.0025	PourAreaTemp
0.0006 ± 0.0025	PourAngle
0 ± 0.0000	DoorOpenTime
0 ± 0.0000	Cure_Time
0 ± 0.0000	Pour_ID
0 ± 0.0000	HeraeusWest_Temp
0 ± 0.0000	SQL_Index
0 ± 0.0000	Operator-_JTM
0 ± 0.0000	DripTime
0 ± 0.0000	OvenTempAtPick
0 ± 0.0000	StreamCNT
0 ± 0.0000	TrayPos
0 ± 0.0000	Gripper_Position
0 ± 0.0000	PourBatchNum
0 ± 0.0000	TotalTimeInOven
0 ± 0.0000	HeraeusEast_Temp
… 26 more …	



Weight	Feature
0.0240 ± 0.0046	PourAngle
0.0086 ± 0.0072	Oven_CNT
0.0080 ± 0.0083	PourDelay
0.0080 ± 0.0074	TiltTime
0.0074 ± 0.0030	Gripper_Position
0.0068 ± 0.0090	PourCompleteDelay
0.0062 ± 0.0055	HeraeusEast_Temp
0.0062 ± 0.0000	SQL_Index
0.0055 ± 0.0060	Unnamed: 0
0.0043 ± 0.0063	Cure_Time
0.0043 ± 0.0030	DoorOpenTime
0.0037 ± 0.0072	TotalTimeInOven
0.0037 ± 0.0119	PourAreaTemp
0.0031 ± 0.0000	Gripper_Torque
0.0025 ± 0.0046	OvenTempAtPick
0.0025 ± 0.0025	Part_Height
0.0025 ± 0.0025	Shift-_1
0.0025 ± 0.0025	Dew Point (°F)
0.0018 ± 0.0030	TrayPos
0.0018 ± 0.0030	VIP_Power_Setting
… 26 more …	

The most weighted features from running the random forest algorithm are PourComplete Oven_CNT, Tiltime, PourAngle, PourDelay, PourAreaTemp





Conclusion

There was **no singular factor** that led to a failed pour.

The main processing factors that led to a failed pour are; (Put this in later).

Identifying Operator error has reduced scrap production by ~8%

Future Scope

Use tenser flow to re-create the prediction model

Identify additional parameters that could be affecting the scrap production

Provide Yamaha engineering with the tools and training documentation

Acknowledgement and References

- Thank You to the Yamaha Team for their continuous support.
- The creators of all the different data analytics tools that were used.
- The Data Mine team for their guidance.

scikit learn | NumPy | pandas

The Data Mine Corporate Partners Symposium 2023