Automation Integration Via Machine Learning: Dermason Bean Classification

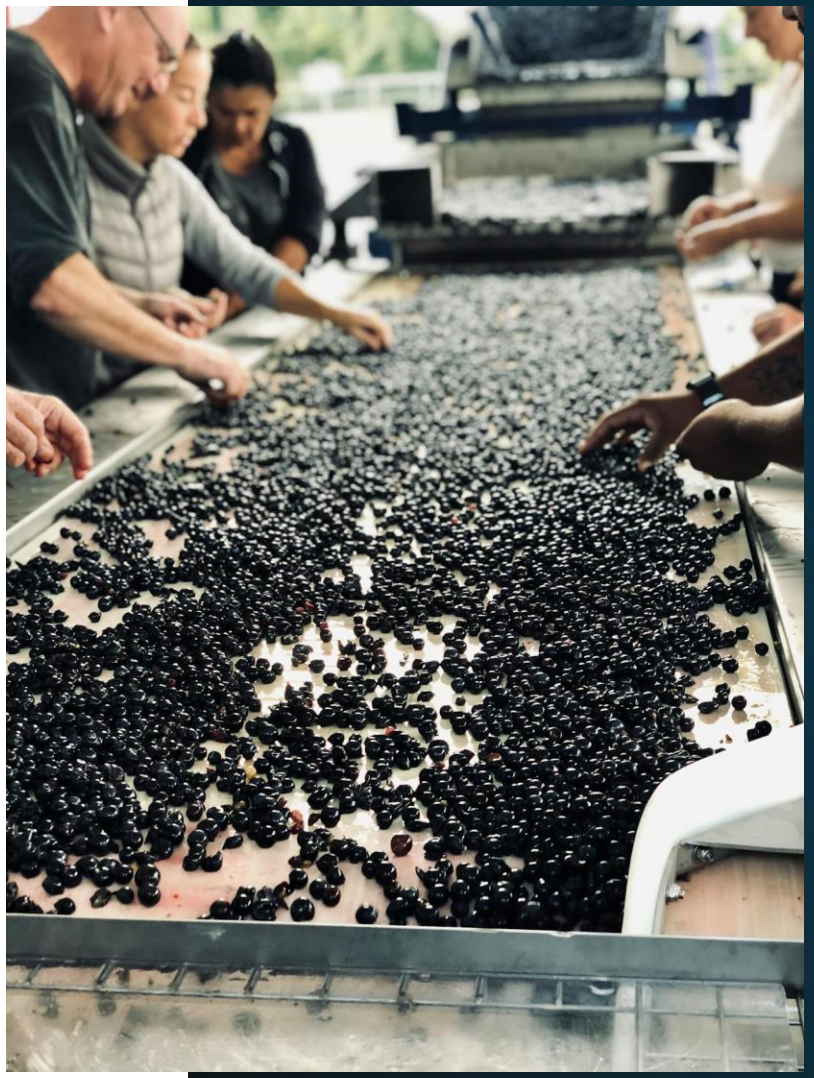
By: Andrew Reusche

Outline

- Business Problem
- The Data
- Methodology
- Baseline Results
  - Logistic Regression
  - Decision Tree
- Tuned Results
  - Logistic Regression
  - Decision Tree
- Final Results
- Conclusions
- Next Steps

Business Problem

- **Area of concern:** Can machine learning assisted automation help improve manufacturing efficiency?
- **Metric of success**: Minimize false positive rate of classification
- **Data extraction:** Bean's dimensional data taken from computer vision program
- **Technologies utilized:** Machine Learning classification models paired with Stratified-K-Fold cross-validation and grid search optimization



The Data

Source: "Dry Bean." UCI Machine Learning Repository, 2020, https://doi.org/10.24432/C50S4B.

Contains

(7) bean types

Converted to (1) Dermason binary classifier

13,611 data instances

16 dimensional variables

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13611 entries, 0 to 13610
Data columns (total 17 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Area             13611 non-null  int64
 1   Perimeter        13611 non-null  float64
 2   MajorAxisLength  13611 non-null  float64
 3   MinorAxisLength  13611 non-null  float64
 4   AspectRation     13611 non-null  float64
 5   Eccentricity     13611 non-null  float64
 6   ConvexArea       13611 non-null  int64
 7   EquivDiameter    13611 non-null  float64
 8   Extent           13611 non-null  float64
 9   Solidity         13611 non-null  float64
 10  roundness        13611 non-null  float64
 11  Compactness      13611 non-null  float64
 12  ShapeFactor1     13611 non-null  float64
 13  ShapeFactor2     13611 non-null  float64
 14  ShapeFactor3     13611 non-null  float64
 15  ShapeFactor4     13611 non-null  float64
 16  Class            13611 non-null  object
dtypes: float64(14), int64(2), object(1)
```

The Data

Limitations

- Unconfirmed Classifications
- Unconfirmed Measurements
- Lack of additional descriptive metrics like weight and color

Data and Model Manipulation Methodology

Data Preprocessing

- Distribution Normalization
- Value Scaling
- Class Imbalance Modification

Hyperparameter Tuning

- Logistic Regression
  - **C** : regularization strength
  - **solver**: weights algorithm
  - **fit_intercept**: include intercept bias?
  - **penalty**: type of regularization

Hyperparameter Tuning

- Decision Tree
  - **criterion**: method of split quality measurement
  - **max_depth** : max # of decision nodes
  - **min_samples_split**: min # of samples required to split node
  - **min_samples_leaf** : min # of samples required to split be in leaf
  - **max_features**: max # of variables considered for each node split

Baseline Model Results: **Logistic Regression**

Average Cross-Validation Score for

Training Data

Specificity: 96.146%

Precision: 88.582%

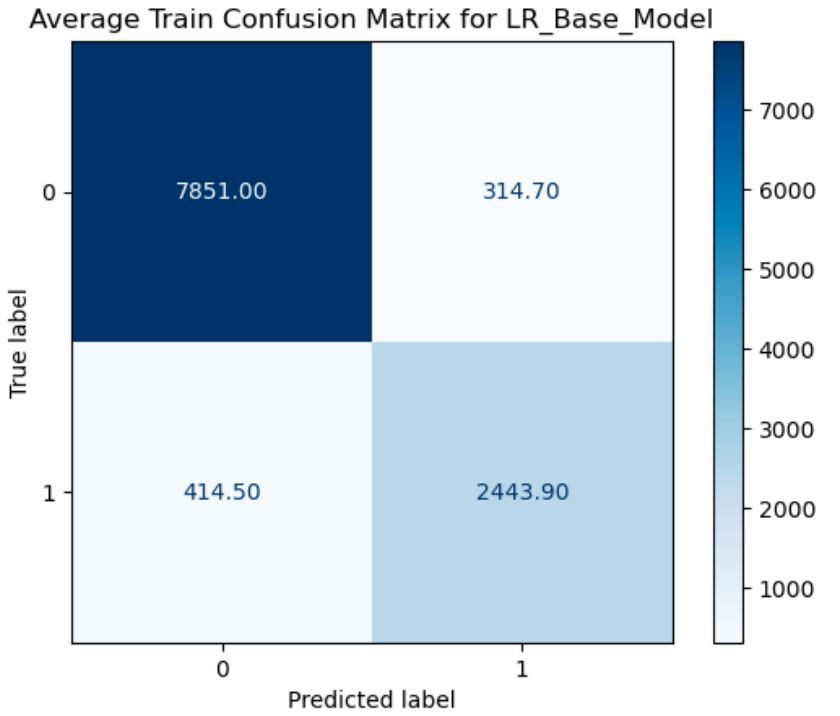
Recal: 85.499%

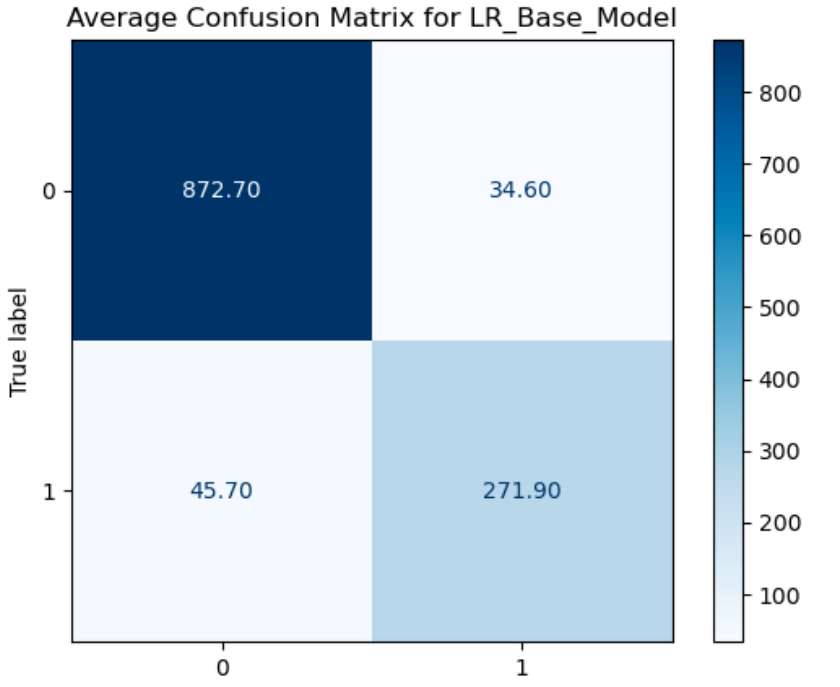
Validation Data

Specificity: 96.187%

Precision: 88.769%

Recal: 85.613%





Baseline Model Results: **Decision Tree**

Average Cross-Validation Score for

Training Data

Specificity: 100%

Precision: 100%

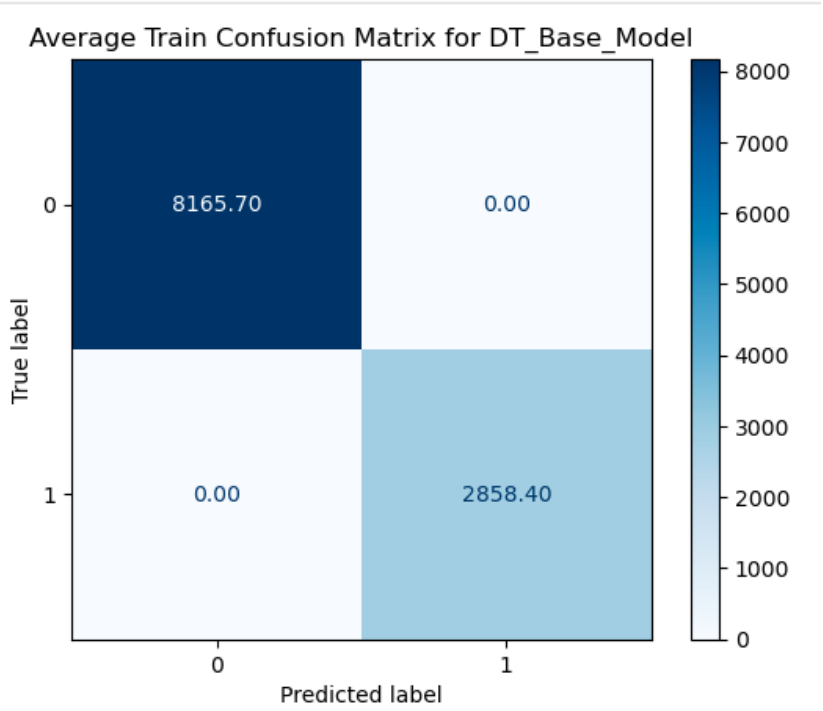
Recal: 100%

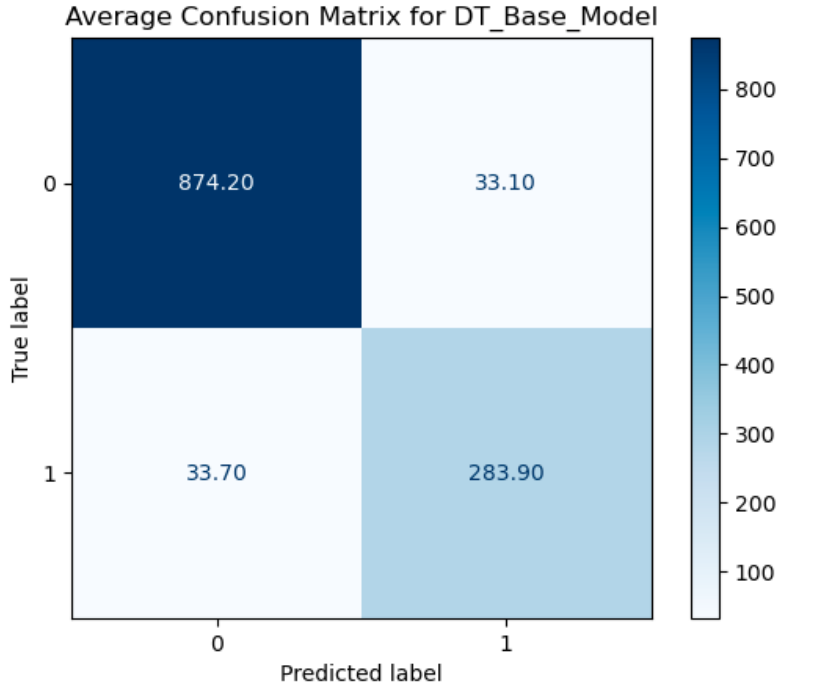
Validation Data

Specificity: 96.352%

Precision: 89.575%

Recal: 89.389%





TUNING TIME

Tuned Model Results: **Logistic Regression**

Average Cross-Validation Score for

Training Data

Specificity: 98.004%

Precision: 93.440%

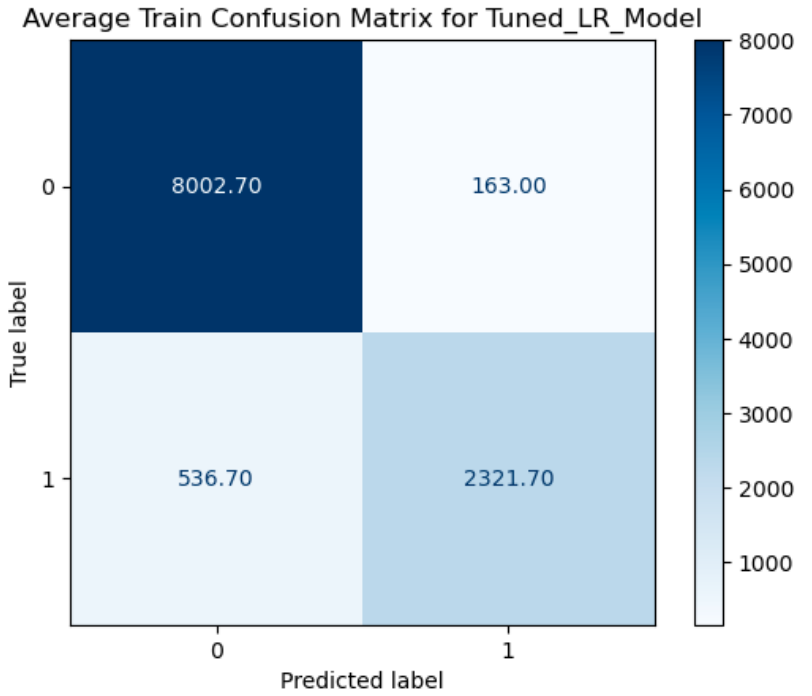
Recal: 81.224%

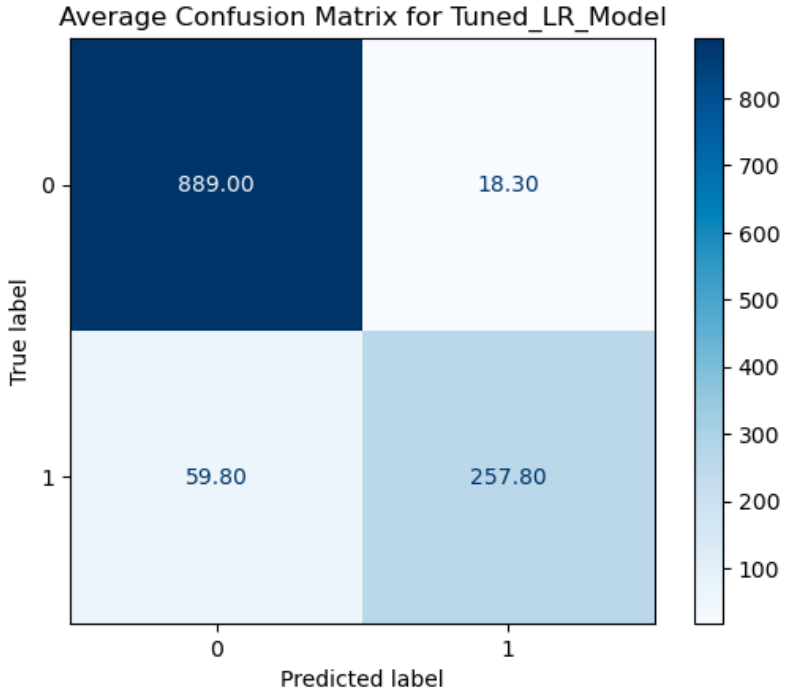
Validation Data

Specificity: 97.983%

Precision: 93.396%

Recal: 81.172%





	v_avg_spec	v_avg_prec	v_avg_rec	v_Kwargs_hyp_para	scaler	v_SM_RU	normalized
82	0.979831	0.933961	0.811720	{'C': 10, 'solver': 'liblinear', 'fit_intercept': True, 'penalty': 'l2'}	False	False	X_train_Norm
56	0.976966	0.929386	0.861462	{'C': 1, 'solver': 'liblinear', 'fit_intercept': False, 'penalty': 'l2'}	MinMaxScaler()	False	X_train
57	0.976966	0.929386	0.861462	{'C': 1, 'solver': 'saga', 'fit_intercept': False, 'penalty': 'l2'}	MinMaxScaler()	False	X_train
24	0.976855	0.928979	0.860518	{'C': 1, 'solver': 'liblinear', 'fit_intercept': True, 'penalty': 'l2'}	MinMaxScaler()	False	X_train
25	0.976745	0.928565	0.859885	{'C': 1, 'solver': 'saga', 'fit_intercept': True, 'penalty': 'l2'}	MinMaxScaler()	False	X_train
114	0.978949	0.927953	0.772675	{'C': 10, 'solver': 'liblinear', 'fit_intercept': False, 'penalty': 'l2'}	False	False	X_train_Norm
125	0.975533	0.927688	0.892632	{'C': 10000, 'solver': 'saga', 'fit_intercept': False, 'penalty': 'l2'}	MinMaxScaler()	False	X_train_Norm
127	0.975533	0.927672	0.892318	{'C': 100000, 'solver': 'saga', 'fit_intercept': False, 'penalty': 'l2'}	MinMaxScaler()	False	X_train_Norm
124	0.975092	0.927163	0.902389	{'C': 10000, 'solver': 'liblinear', 'fit_intercept': False, 'penalty': 'l2'}	MinMaxScaler()	False	X_train_Norm

Tuned Model Results: **Decision Tree**

Average Cross-Validation Score for

Training Data

Specificity: 98.668%

Precision: 98.390%

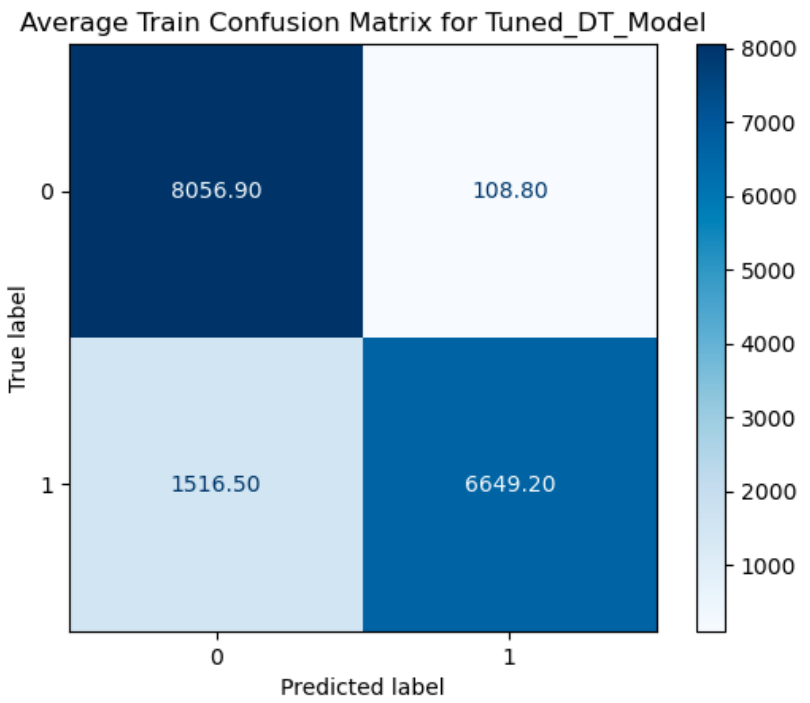
Recal: 81.428%

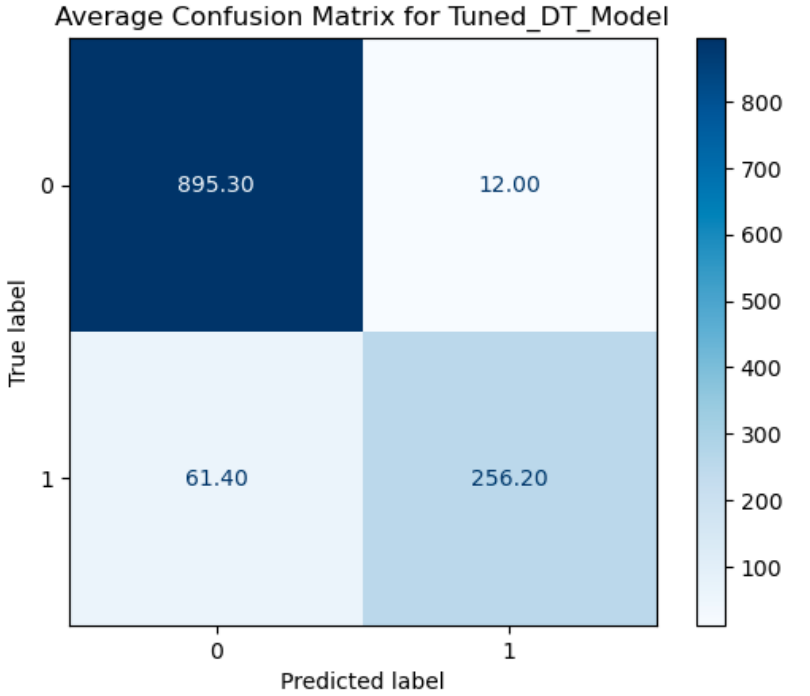
Validation Data

Specificity: 98.677%

Precision: 95.563%

Recal: 80.668%





	v_avg_spec	v_avg_prec	v_avg_rec	v_hyp_para	scaler	v_SM_RU	normalized
10	0.986774	0.955633	0.806682	{'max_depth': 2, 'min_samples_split': 0.05, 'criterion': 'gini', 'min_samples_leaf': 0.1425, 'max_features': 4}	False	True	X_train
22	0.986774	0.955633	0.806682	{'max_depth': 2, 'min_samples_split': 0.05, 'criterion': 'gini', 'min_samples_leaf': 0.1425, 'max_features': 4}	False	True	X_train
34	0.986774	0.955633	0.806682	{'max_depth': 2, 'min_samples_split': 0.1, 'criterion': 'gini', 'min_samples_leaf': 0.1425, 'max_features': 4}	False	True	X_train
46	0.986774	0.955633	0.806682	{'max_depth': 2, 'min_samples_split': 0.1, 'criterion': 'gini', 'min_samples_leaf': 0.1425, 'max_features': 4}	False	True	X_train
58	0.986774	0.955633	0.806682	{'max_depth': 2, 'min_samples_split': 0.15, 'criterion': 'gini', 'min_samples_leaf': 0.1425, 'max_features': 4}	False	True	X_train
70	0.986774	0.955633	0.806682	{'max_depth': 2, 'min_samples_split': 0.15, 'criterion': 'gini', 'min_samples_leaf': 0.1425, 'max_features': 4}	False	True	X_train

FINAL TEST

Final Model: **Decision Tree**

Preprocessing

Scaler: None

Normalization: False

SMOTE/RandUnd: True

Hyperparameters

max_depth: 2

min_samples_split: 0.05

criterion: gini

min_samples_leaf: 0.1425

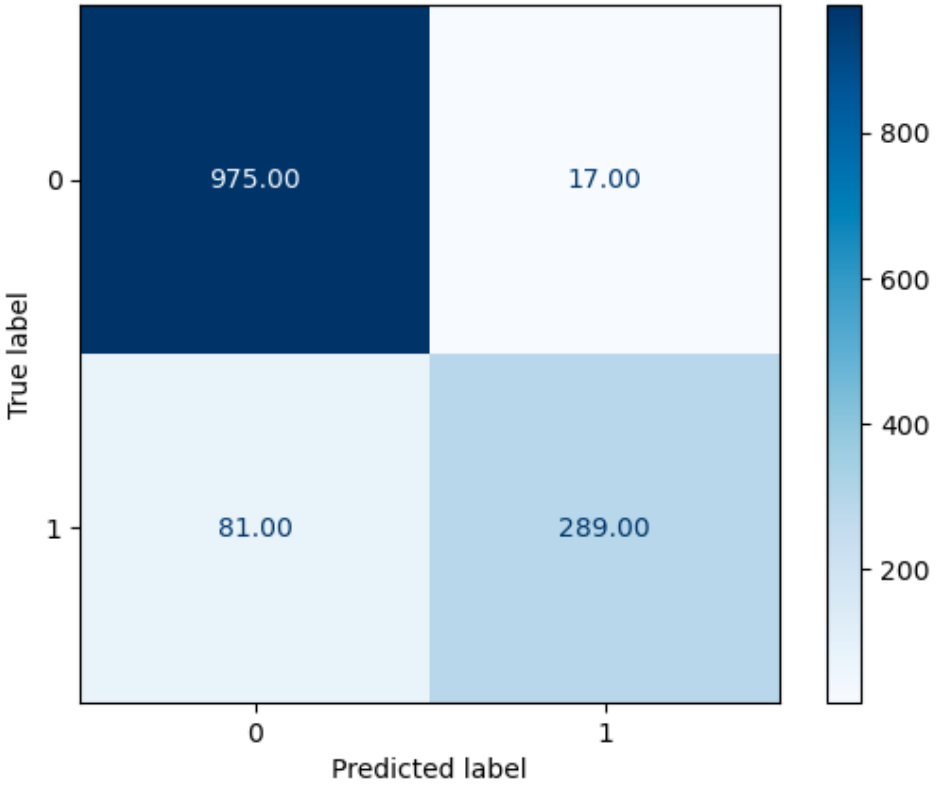
max_features: 4

Performance Scores:

Specificity: 98.286%

Precision: 94.444%

Recal: 78.108%



Conclusion

Classification via Supervised Machine Learning can help implement process automation.

Up to 78% of the target beans could be classified, and the rest can still be sold for profit.

<2% of non-target beans are falsely classified, and <6% of beans classified as target are falsely classified.

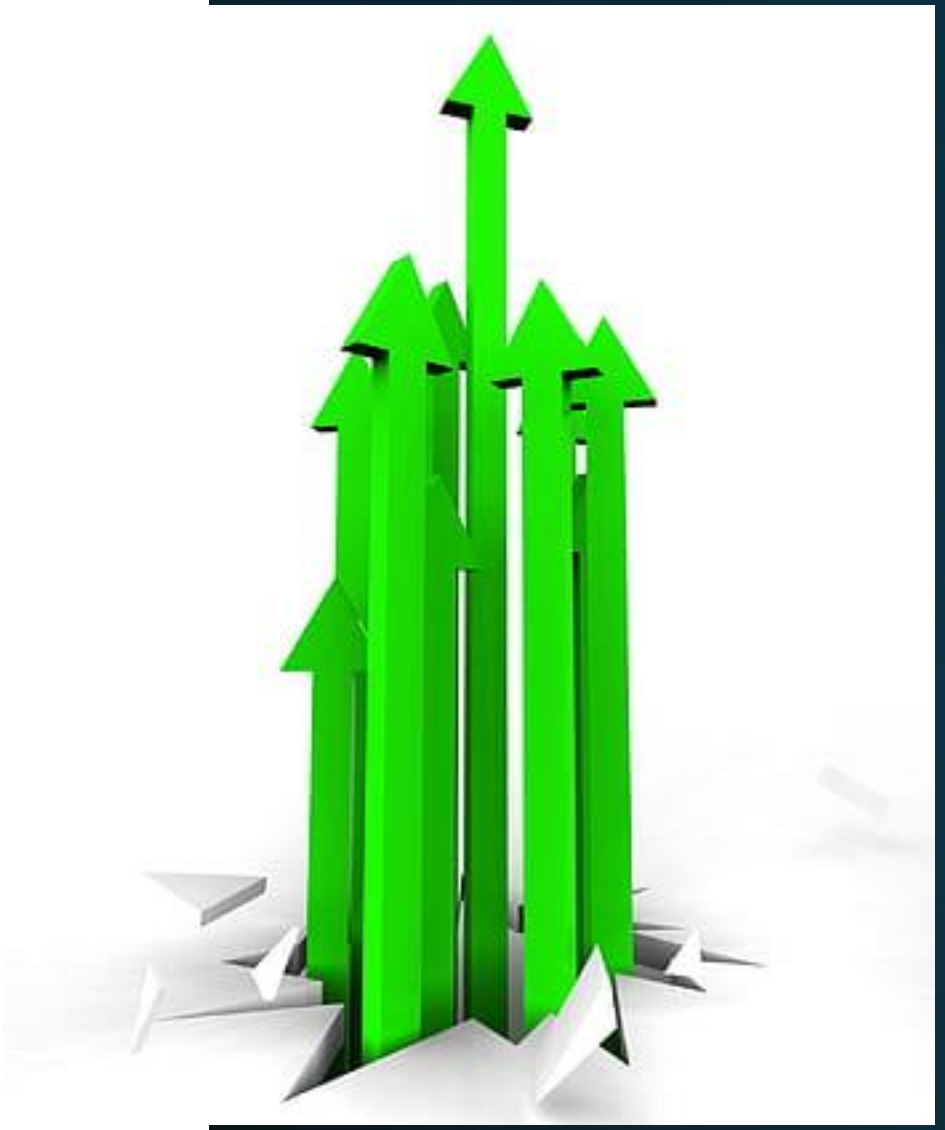
The need for manual bean filtering could be significantly reduced, but maybe not completely eliminated.

Next Steps

More Sensors = Better Results?

New Models = Better Results?

New Metrics = Grade and Quality Classifications



Thank you!

- Email: A.J.Reusche@gmail.com
- Github: https://github.com/AndrewReusche
- LinkedIn: https://www.linkedin.com/in/andrew-reusche-1397bb311/