RED WIND QUALITY ANALYSIS AND PREDICTION

DAVID LIU

RED WINE QUALITY PROBLEM AND OBJECTIVE

Problem Statement:

A wine company wants to improve its quality control process by implementing a machine learning model that can accurately classify red wines into good/bad quality categories based on chemical properties.

Objective:

To develop a machine learning model that can accurately classifies red wines into quality categories (eg, bad, good) based on their physicochemical properties.

TABLE OF CONTENTS

EDA

1. Dataset Overview
2. Correlations
3. Distribution plot for each feature
4. Feature Engineering

DATA OVERVIEW

11 Feature columns plus one quality columns

All features are positive numerical value

1599 rows of data

No missing values

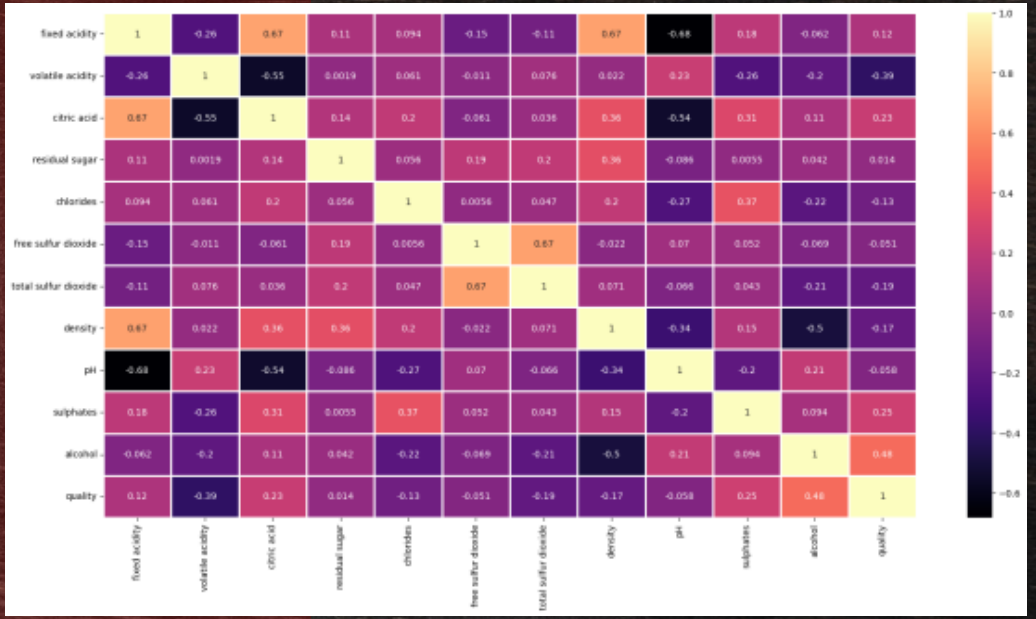
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

CORRELATION

Relatively high positive correlation(0.67) between fixed acidity acid and density.

Relatively high negative correlation (-0.68) between pH and fixed acidity acid.

We need to consider multicollinearity when building model.

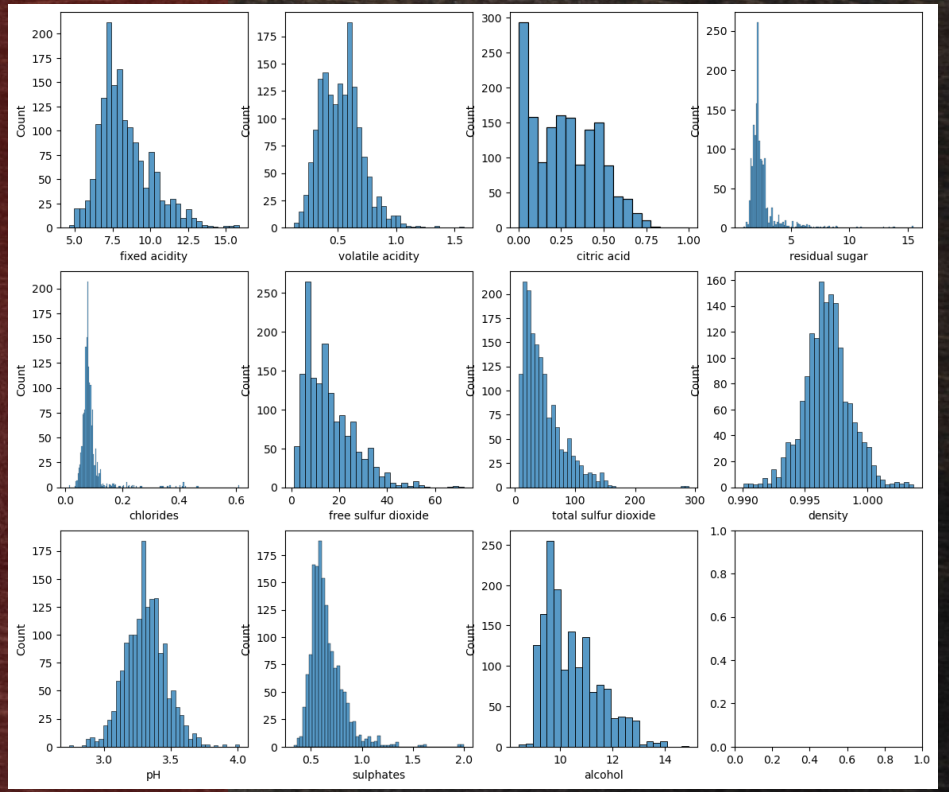


DISTRIBUTION FOR EACH FEATURE

Density, pH are more likely to be normally distributed

Other features are right skewed.

Need to apply standard scale for all feature



FEATURE ENGINEERING

Encode Y variable to 1/0(good/bad quality)

Train-Test Split

Standarad Scale features

ASSUMPTION

1. Data contain patterns that can be learne through hierarchical representation.
2. Ensemble learning
3. Gradient descent optimization

MODEL BUILT

KNN Classifier

Gradient Boosting Machines

Random Forest Classification

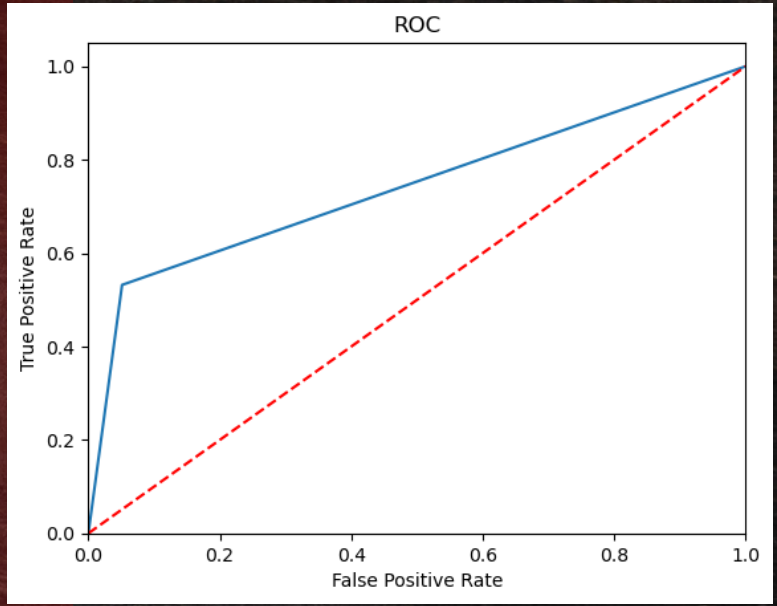
Deep Learning with Regularization

KNN CLASSIFIER

KNN Classifier with uniform weights has training accuracy 0.94 and test accuracy 0.875, may overfitted. With distance weights has training accuracy 1 and test accuracy 0.89, result looks improved than uniform weights, but definitely overfitted.

GBM MODEL

Using GridSearchCV, we tuned the hyperparameters to find the best parameters, and the final result looks better than KNN. The training accuracy is 0.906 and test accuracy is 0.8875, close enough for not overfitting and accuracy.



RANDOM FOREST CLASSIFICATION

Random Forest Classifier with training accuracy 1 and test accuracy 0.9125, overfitted.

DEEP LEARNING

6 Dense Layer with ‘relu’ activation function

Batch Normalization for each layer

0.2 Dropout for each layer

Total train parameters 23,109

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 64)	768
batch_normalization (BatchNormalization)	(None, 64)	256
dropout (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 32)	2,080
batch_normalization_1 (BatchNormalization)	(None, 32)	128
dropout_1 (Dropout)	(None, 32)	0
dense_2 (Dense)	(None, 32)	1,056
batch_normalization_2 (BatchNormalization)	(None, 32)	128
dropout_2 (Dropout)	(None, 32)	0
dense_3 (Dense)	(None, 32)	1,056
batch_normalization_3 (BatchNormalization)	(None, 32)	128
dropout_3 (Dropout)	(None, 32)	0
dense_4 (Dense)	(None, 32)	1,056
batch_normalization_4 (BatchNormalization)	(None, 32)	128
dropout_4 (Dropout)	(None, 32)	0
dense_5 (Dense)	(None, 32)	1,056
batch_normalization_5 (BatchNormalization)	(None, 32)	128
dropout_5 (Dropout)	(None, 32)	0
dense_6 (Dense)	(None, 1)	33

MODEL EVALUATION

Run model for train and test for modal stability

Check overfitting

Pick the one most stable, higher accuracy and no overfitting.

MODEL SUMMARY TABLE

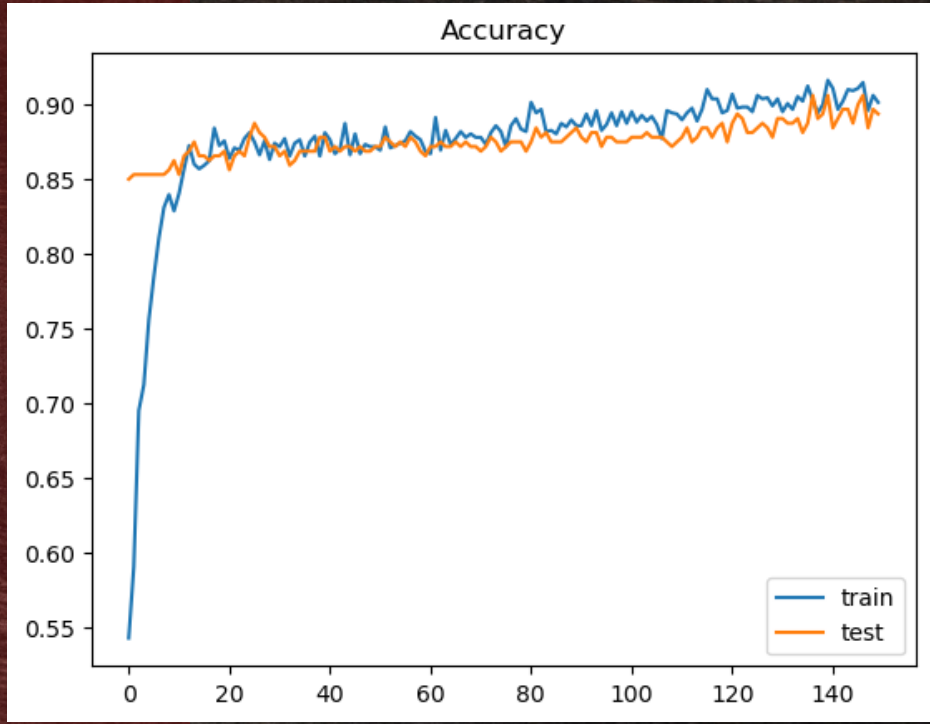
	TRAIN ACCURACY	TEST ACCURACY	OVERFITTING
KNN	1	0.875	Yes
GBM	0.905	0.888	No
RANDOM FOREST	1	0.913	Yes
DEEP LEARNING	0.901	0.894	No

DEEP LEARNING

Train and test accuracy relatively follows each other, no overfitting issue

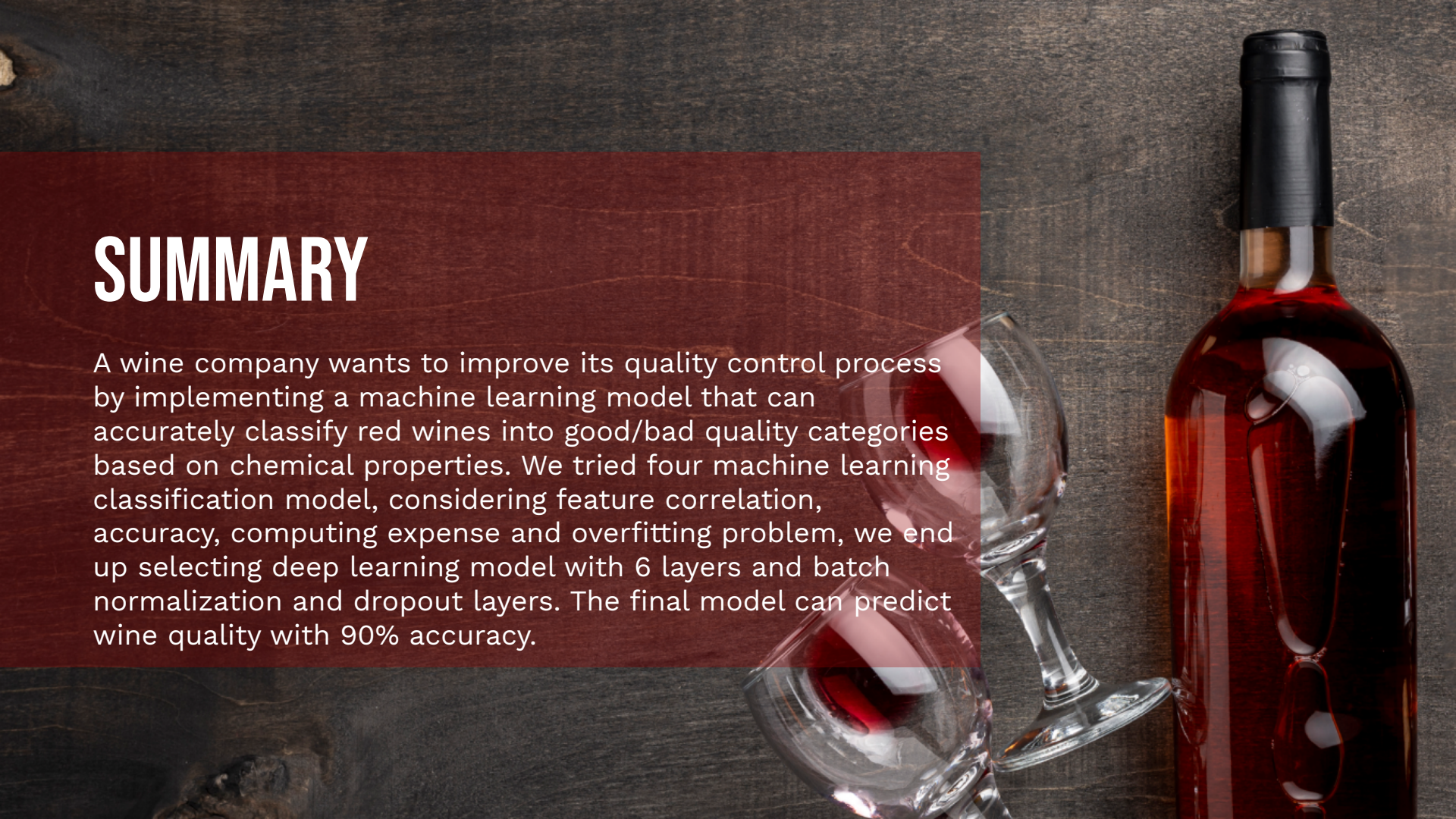
Model provide relatively high accuracy(0.9)

Not too expensive computing power



SUMMARY

A wine company wants to improve its quality control process by implementing a machine learning model that can accurately classify red wines into good/bad quality categories based on chemical properties. We tried four machine learning classification model, considering feature correlation, accuracy, computing expense and overfitting problem, we end up selecting deep learning model with 6 layers and batch normalization and dropout layers. The final model can predict wine quality with 90% accuracy.



FUTURE IMPROVEMENT OF MODEL:

Continuing collect more data, and retrain the model, if model efficiency start dropping with more data came in, then we can try different layers, or break down the outcome to muti-classification model instead of good/bad classification.

THANKS FOR WATCHING!

CREDITS: This presentation template was created by **Slidesgo,** including icons by **Flaticon** and infographics & images by **Freepik**