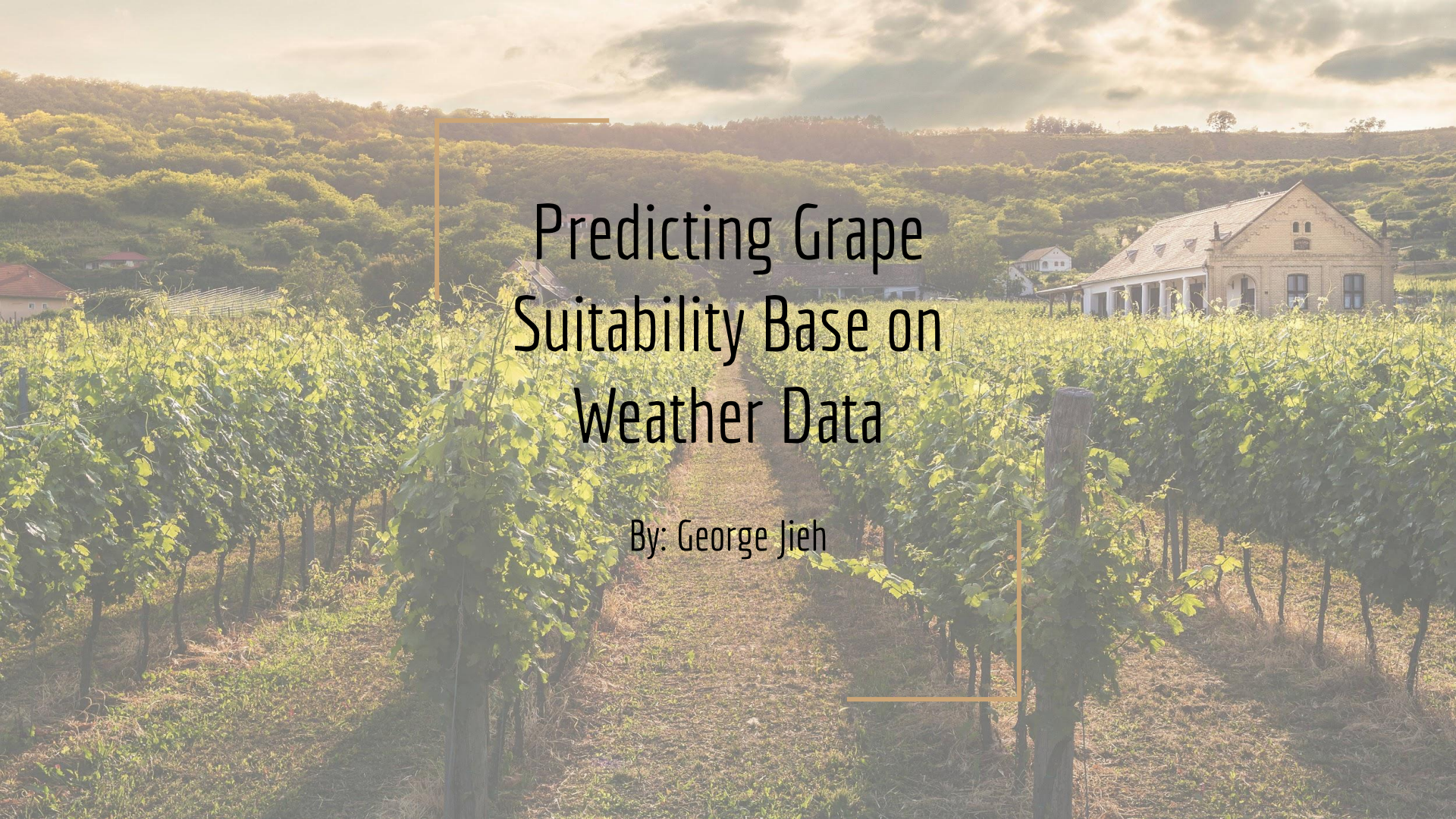


Predicting Grape Suitability Base on Weather Data

By: George Jieh

Non-Technical Overview

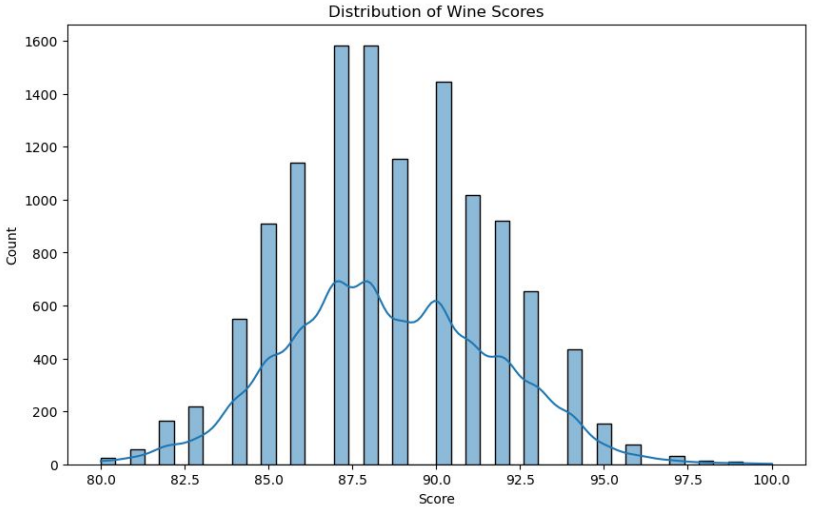
- **Problem Statement**
  - **Climate change is heavily impacting the wine industry by causing locations that are fit to traditionally produce certain wines to be no longer suitable for those grapes. This causes decrease in quality and make wine grape growth cycles increasing hard to get right by vintners. This results in crop waste.**
- **Solution**
  - **Develop a model that can predict a location's suitability for certain wine grape types.**
- **Impact**
  - **Reduce crop waste.**
  - **Discover new areas for viticulture.**
  - **Increase supply of quality wine, which in turn can make them more accessible.**

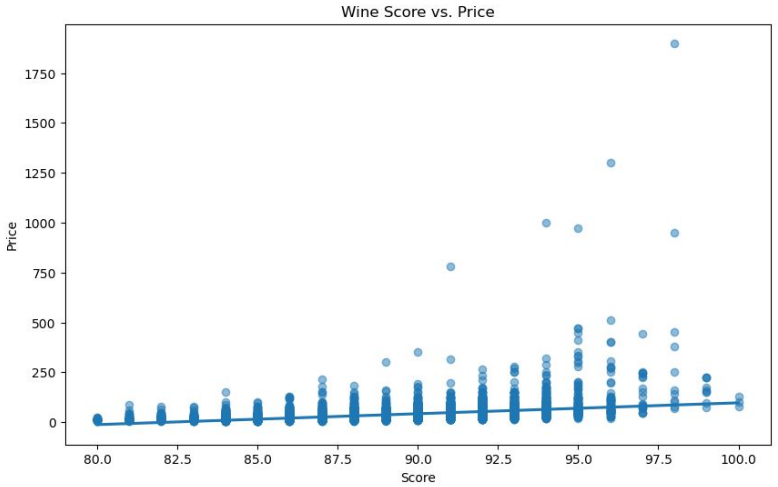
The Dataset

- Dataset is a combination of two datasets.
- The wine data is a combination of a data set from Kaggle that someone scraped from Wine Enthusiast website and from my own scraping of data from Wine Enthusiast with my own custom scraper.
- The weather data is fetched from World Weather Online API. These weather APIs all need latitude and longitude information, so that was fetched through Bing Maps API.
- The combined dataset contains name of wine, name of winery, price, wine score, location details, temperature data, precipitation, sun exposure data, and humidity.
- All climate data is in metric system.
- During data cleaning process I reorganize column order, change column names, drop bad data, denote North and South hemisphere, and combine months into grape growth phases to reduce column amounts.
- The grape growth phases are Budburst, Early Growth, Flowering, Fruit Set, Verasion, Harvest, Early Dormancy, and Late Dormancy. Some of the months for each phase overlap, so I combined some of these phases into 4 main phases.

Exploratory Data Analysis

- Detailed labeling seems to be associated with higher quality and price.
- Wine score distribution gathers around 87 to 88 points.
- While there is a trend of higher prices for higher-scoring wines, there is a substantial number of reasonably priced wines with high scores, indicating that quality is accessible at various price points.
- High scoring wines have ample amount of rain during BEG and FFV and tends to have the least amount of rain during LD and HED.
- Temperature for high quality wines during FFV tends to go the not too hot but not to cold route, but tends to have the lowest temperatures in all other cycles. The same could be said for sun exposure as well.
- Humidity seems to be all over the place too, with Riesling being able to withstand high humidity the most. However this is most likely because Riseling have many different styles and one of the most expensive styles is late harvest, which encourages Botrytis, a type of grape fungal growth, that requires high humidity right before sunrise and enough heat to dry off the dew quickly during the rest of the day.





Models

- The models that I have tested where I fully built from the ground up are linear regression, random forest, gradient boosting, and neural network.
- We are building the model with 24 features and after filtering, 20 classes.
- After playing around with the parameters random forest style models ended up being the most accurate, albeit accurate is relative, since it was still in the low 50%.
- During testing neural network it came to my attention that the dataset have a very imbalance sample size for each class tested. So I started using SMOTE (Synthetic Minority Oversampling Technique) for classes that have to few samples and random undersampling for classes that have way too many samples.
- Tried PCA, but in the grander scheme of things, it doesn't really work. Albeit it says I can cut my features by half, however "scientifically" all these climate data supposedly matters.
- In the end to attempt to have a higher accuracy, i played around with three AutoML libraries:
    - H2O AutoML
    - TPOT
    - AutoKeras
- H2O and TPOT both recommended a random forest model. TPOT was the standard random forest model, while H2O came up with a Distributed Random Forest Model.
- AutoKeras was able to train a deep learning model to eventually get to a relatively high accuracy (around 80%), but I suspect it for being auto fitted since it had to run up to 950 epochs to get to that range.
- Ultimately in the end the Distributed Random Forest model tuned by H2O was the model chosen since it had the highest average accuracy (56%) and lowest errors and losses among all the models (5 manually by me and almost 80 different models from the AutoML libraries) tested.


Web App Demo