Predicting ratings of the beer based on its profile

1 Introduction

We got the inspiration for the database from beer production at Aalto University. One of the participants in this project has experience in beer production. It was interesting for us to delve into the topic of the rating and understand by what criteria customers prefer one sort and manufacturer of beer to another.

The data integration aims to create a new data set that contains comprehensive consumer reviews (appearance, aroma, palate, taste, and overall review scores) for different brews, combined with their detailed tasting profiles.

2 Problem formulation

Given the data in the dataset, how accurately can we predict the ratings of a beer based on its profile?

The dataset was provided by integrating information from two existing datasets on Kaggle[1]. Dataset provides beer ratings with its profiles. This data set contains tasting profiles and consumer reviews for 3197 unique beers from 934 different breweries. It includes beer name (label), style, brewery name, complete beer name (brewery + brew Name - unique identifier for each beer), description (notes on the beer if available), ABV (alcohol content of beer, % by volume), min IBU (the minimum IBU value each beer can possess), max IBU (the maximum IBU value each beer can process); mouthfeel: astringency, body, alcohol; taste: bitter, sweet, sour, salty; flavor and aroma: fruits, hoppy, spices, malty; review: aroma, appearance, palate, taste, overall; several reviews. The types of data are continuous. We are going to use beer profiles as labels and ratings as features. Our goal is to understand their preference for a particular beer label based on reviews. To analyse the data, we will use the Regression methods. For determining the best fit model we will use both methods and compare accuracy. A loss function measures how well a given machine learning model fits the specific data set. We will use mean squared error as my loss function.

3 Methods

3.1 Dataset

The project's dataset is collected with a BeerAdvocate source and a Beer tasting profiles dataset. As well as 1.5 million beer reviews by Tanya Cashorali (uploaded by Datadoume). The dataset contains tasting profiles and consumer reviews for 3197 unique beers from 934 different breweries. As mentioned earlier the dataset as well contains information about the beer, for example, label, alcohol content, style, mouthfeel, and more. In addition to numerical data, the table also contains non-numerical data, but for the analysis, we used only numerical data. The beer-style column was described by a string. So to analyze the data, we need a numeric value which is why each style we assigned values 0-110. In the beer profile and rating dataset, we have multiple independent features based on them we are trying to predict results.

3.2 Linear regression

Linear regression establishes the linear relationship between two variables based on a line of best fit and we can define how the change in one variable impacts a change in the other. Linear regression uses mean squared error as its loss function. The mean squared error (MSE) tells you how close a regression line is to a set of points. We used Excel to visualise the data on a graph. In figure 1 we will present how the label will be classified according to the attributes of beer (in this case body). And in Figure 2 the label will be classified according to the style.

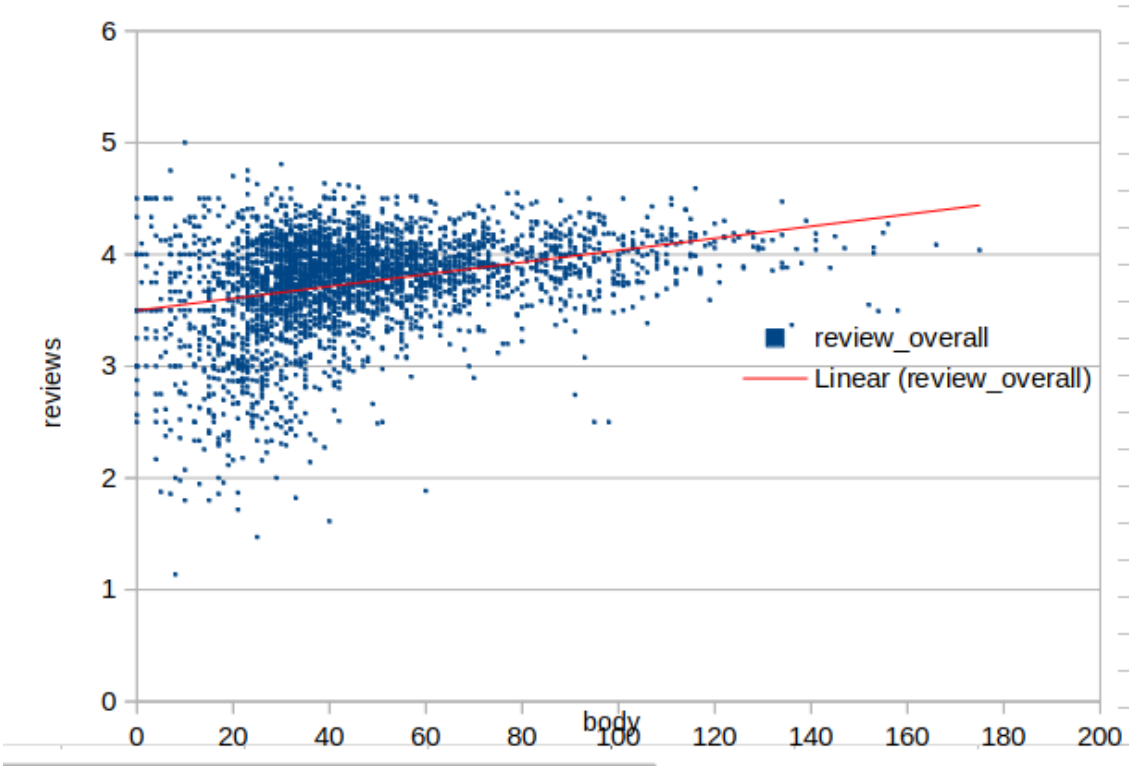

Figure 1. Beer label vs body.

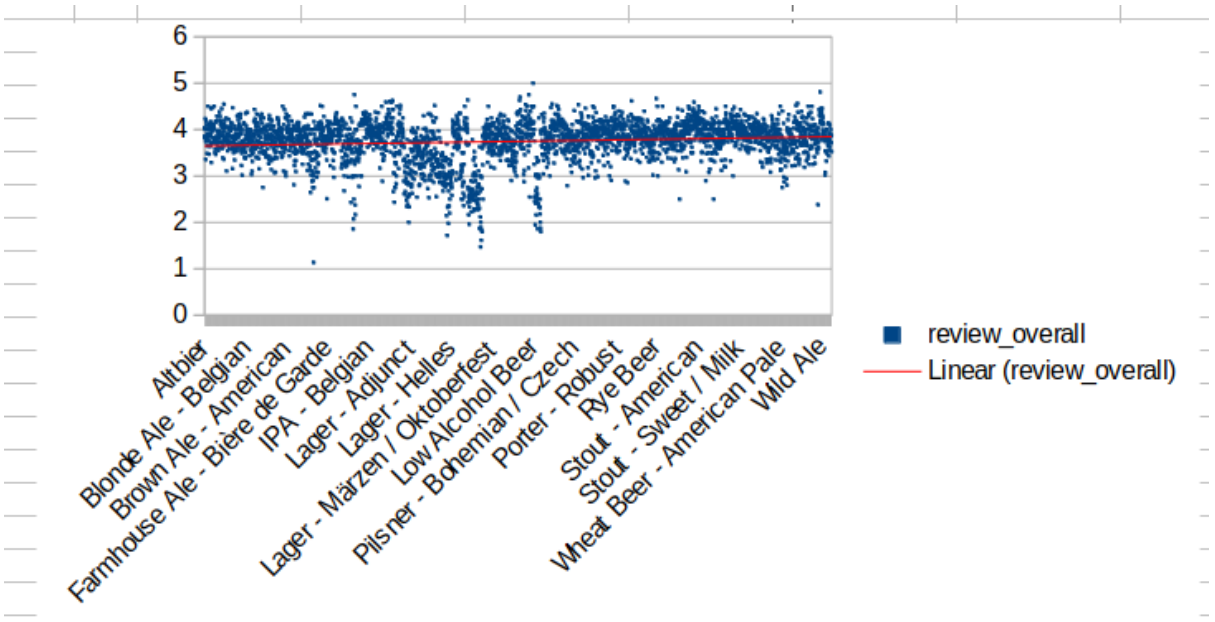

Figure 2. Beer label vs style.

3.3 Polynomial regression

Polynomial regression is a form of regression analysis in which we can see that we are trying to estimate the relationship between coefficients and y. In this method for data analysis, we used beer style, the alcohol content of beer (ABV), min IBU, max IBU, mouthfeel: astringency, and body. Polynomial regression uses mean squared error as its loss function. The mean squared error (MSE) tells you how close a regression line is to a set of points. We tried different degrees of polynomial regression i.e. from 1 to 4, as you can see in Figure 3, to get a better one for our data. Polynomial regression uses mean squared error as its loss function.

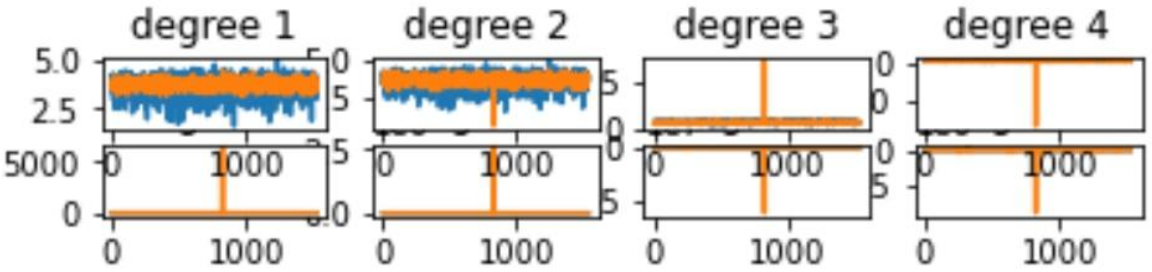

Figure 3. 1 to 4 degrees of polynomial regression.

3.4 Training, testing and validation

We split the dataset into train and validation sets. Using the train_test_split() method the training set and validation set are split by 0.7 and 0.3 respectively. It gives the best results in terms of high accuracy and low loss.

4 Results

The linear regression algorithm works when the relationship between the data is linear. You can see the linear regression results on the graph, which do not perform well means they do not come close to reality. Hence, we should introduce polynomial regression to overcome this problem. We used 70% of the dataset for training and 30% for verifying and noticed the difference between the training accuracy and validation accuracy is lowest for the Polynomial regression model. Thus, this is the model we choose here. In comparison with using different degrees of polynomial regression, we came to the result that degree 2 showed the best model. Figure 4 shows the distribution of our dataset using the polynomial regression model.

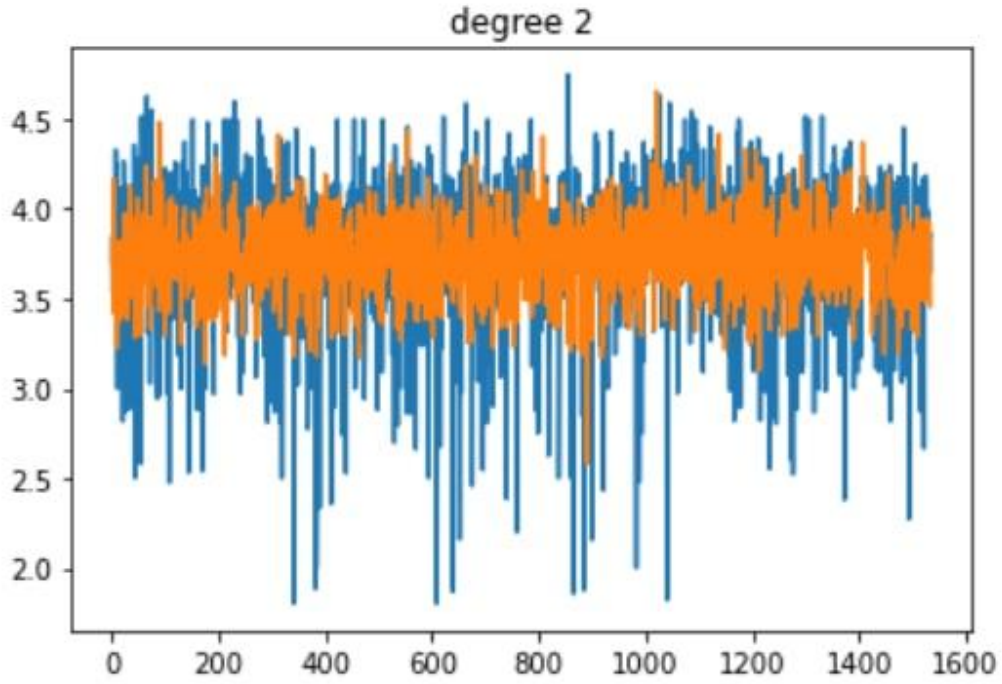

Figure 4. Polynomial regression model, degree 2.

5 Conclusion

Since there are a lot of nuances in the brewing process that cannot be represented by statistics, as well as beer is an extremely complex drink, has more extensive characteristics, and is highly dependent on individual preferences, the graph cannot be considered accurate, since it is very difficult to predict the beer rating based only on the presented small list of characteristics, namely the data that is displayed in the dataset we used.

References

[1] https://www.kaggle.com/datasets/ruthgn/beer-profile-and-ratings-data-set

Appendix

```
import numpy as np
import pandas as pd
from scipy import constants
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
from sklearn.preprocessing import PolynomialFeatures
data = pd.read_csv('beer_profile_and_ratings.csv')
X = np.asarray(data[["Style_n","ABV", "Min IBU", "Max IBU", "Astringency","Body" ]])
y = np.asarray(data["review_overall"])
X_train, X_rem, y_train, y_rem = train_test_split(X, y, test_size=0.6)
X_val, X_test, y_val, y_test = train_test_split(X_rem, y_rem, test_size=0.2)
lin_reg_mod = LinearRegression()
lin_reg_mod.fit(X_train, y_train)
pred_train = lin_reg_mod.predict(X_train)
tr_error = mean_squared_error(y_train, y_pred_train)############
pred_val = lin_reg_mod.predict(X_val)
val_error = mean_squared_error(y_val, y_pred_val)############
print(tr_error, val_error)
plt.plot(y_val , label = "validation")
plt.plot(y_pred_val, label = "predicted")
plt.legend()
plt.show()
tr_errors, val_errors = [], []

for n in range(8, 0, -1):

    poly = PolynomialFeatures(degree = n)
    X_train_poly = poly.fit_transform(X_train)
    regr = LinearRegression(fit_intercept=False)
    regr.fit(X_train_poly, y_train)
    y_pred_train = regr.predict(X_train_poly)
    #tr_error = (np.sqrt(mean_squared_error(y_train, y_pred_train))
    tr_error =mean_squared_error(y_train, y_pred_train)##########

    X_val_poly = poly.fit_transform(X_val)
    y_pred_val = regr.predict(X_val_poly)

    #val_error = (np.sqrt(mean_squared_error(y_val, y_pred_val))
    val_error = mean_squared_error(y_val, y_pred_val)##################
    tr_errors.append(tr_error)
    val_errors.append(val_error)
    if n == 2:
        plt.title("degree {}".format(n))
        plt.plot(y_val , label = "validation")
        plt.plot(y_pred_val, label = "predicted")

plt.show()
print("tr: ",tr_errors)
print("vl: ",val_errors )
```