Project Machine Learning
Task 2: Data Preparation
9 group

Tarelkin Evgenii
Steba Oxana

2021

.

1 Task 1

b. Added new column "Year" (vintage of wine). The year is taken from the "title" column.
c. A histogram for each column that visualizes the distribution of its values.

Column "country". Most wines are produced in the USA. The second and the third place on frequencies are France and Italy (Figure 1).
Column "country" has 63 nan values.

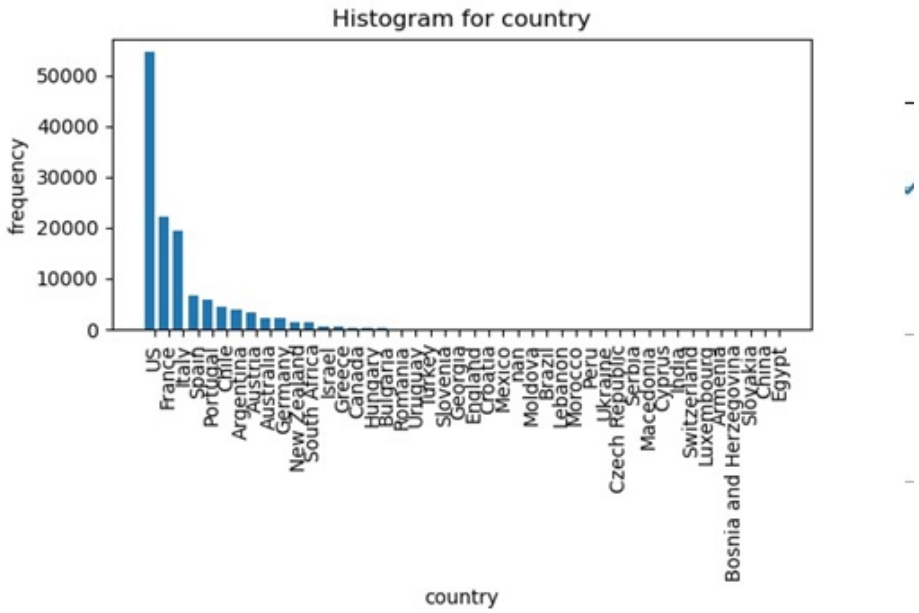

Figure 1: Histogram with values from the "Country" column

Column "points". In this dataset, the column "points" have values from 80.0 to 100.0. The most frequent values are between 86 and 92. Here we use bins equal to 10. Values are divided into 10 groups. Column "points" has 0 nan values (Figure 2).

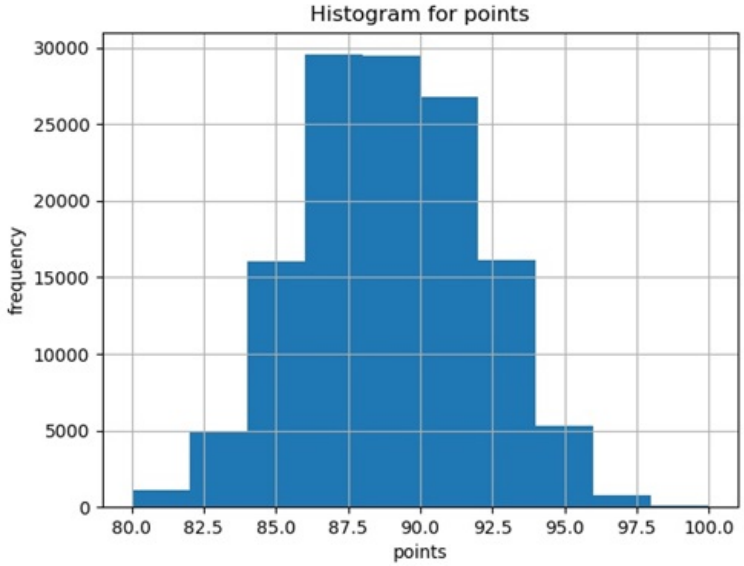

Figure 2: Histogram with values from the "Points" column

Column "price". Most of the values in the price column are less than 250. Column "price" has 8395 nan values (Figure 3).

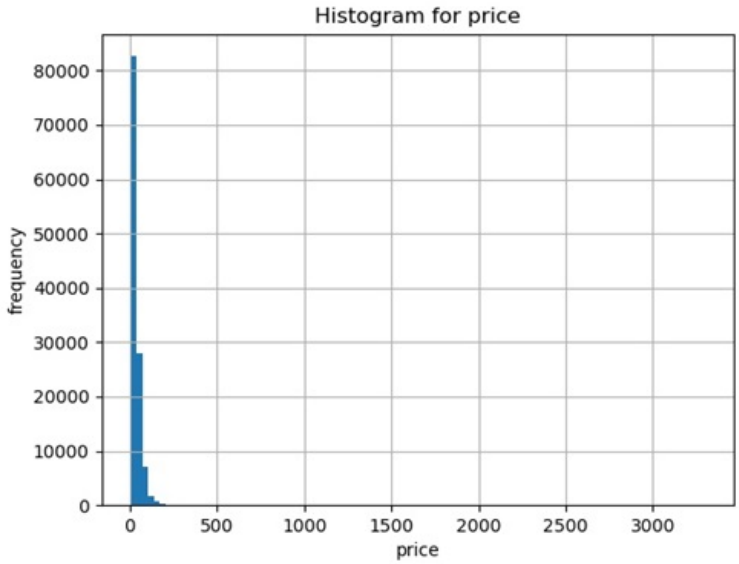

Figure 3: Histogram with values from the "Price" column

Column "province". Here we display provinces that are more than 60 times (Figure 4).

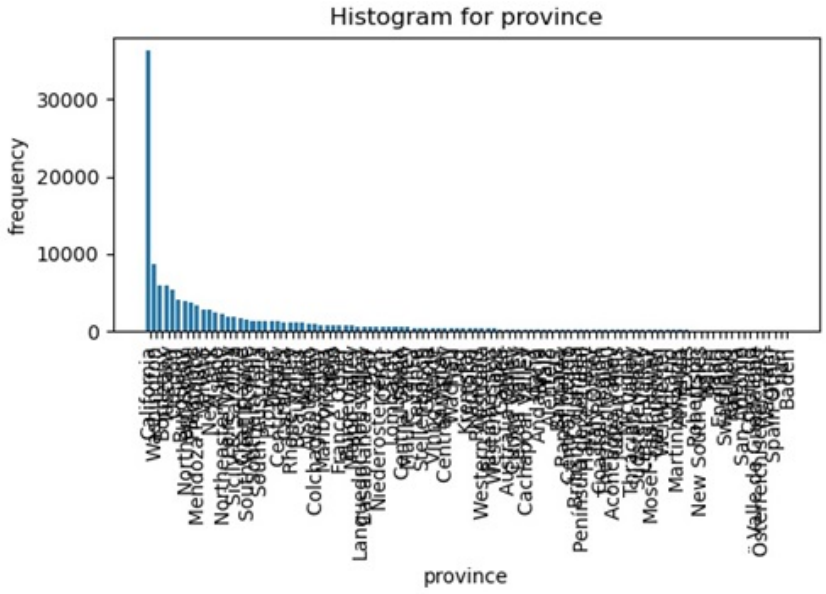

Figure 4: Histogram with values from the "Province" column

We change the value of the border, so now we show the provinces that greater than 560. We can't see the Nan value in the graph that means that empty values meet less than 560 times. The province with the greatest frequency value is California, more than 30 000 times. The next is Washington with almost 10 000 times (Figure 5).
Column "province" has 63 nan values.

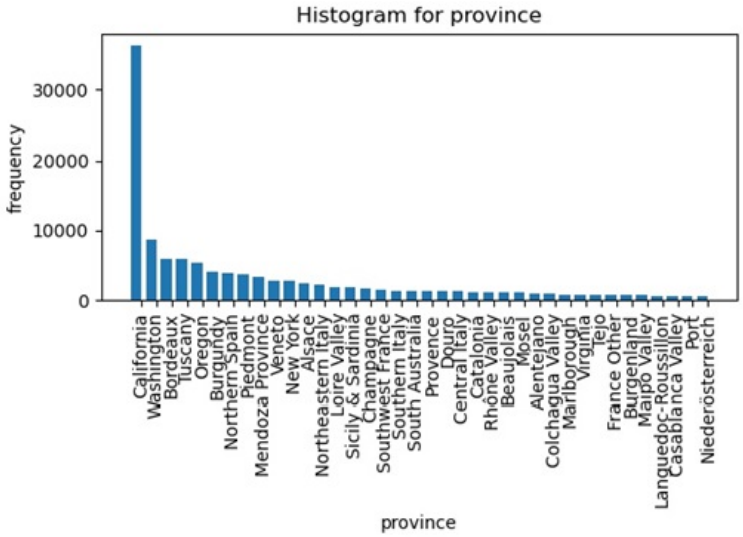

Figure 5: Histogram with values from the "Province" column

Column "region 1". Show the regions more frequent than 70 (Figure 6).

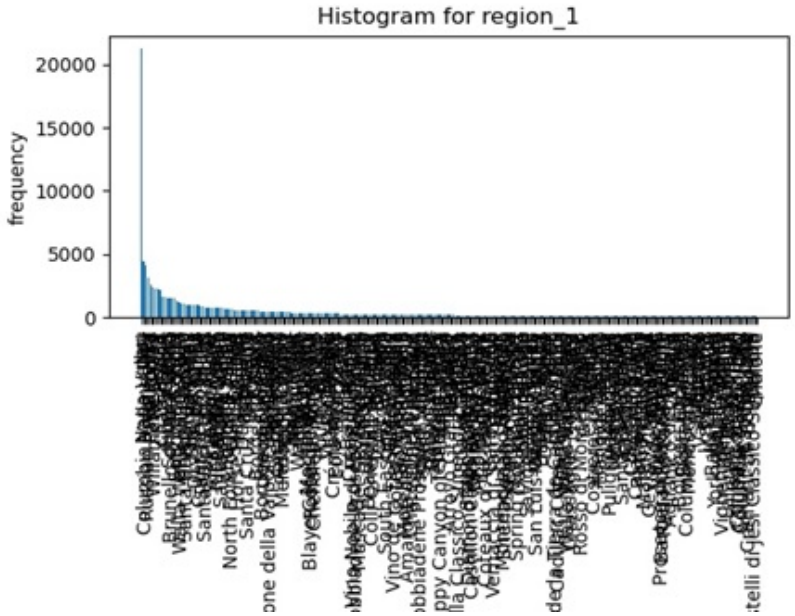

Figure 6: Histogram with values from the "region 1" column

We are not able to read the information from the plot, we increase the minimum value from 70 to 550. In this plot, we can see that the nan values prevail. The prevalent regions are Napa Valley and Columbia Valley (WA) (Figure 7).
Column "region 1" has 21247 nan values.

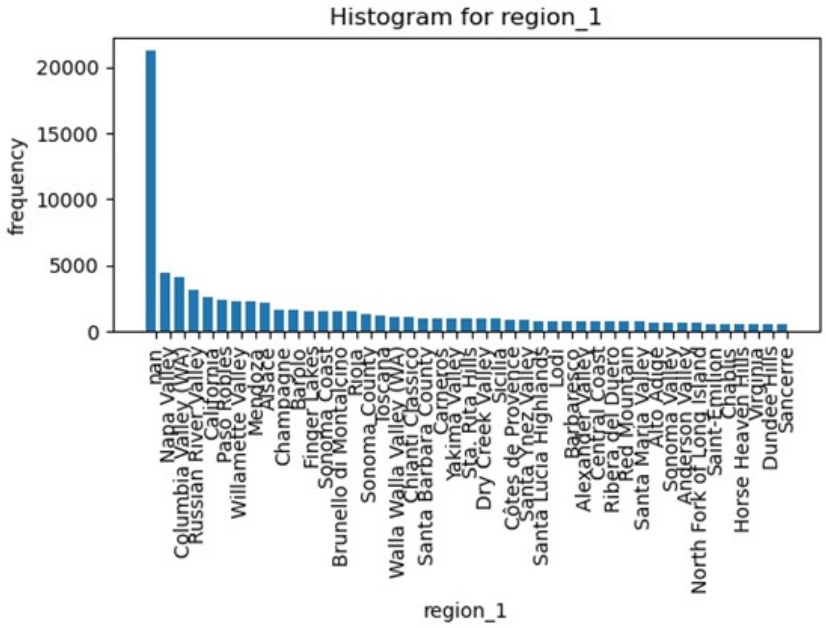

Figure 7: Histogram with values from the "region 1" column

Column "region 2". The most common second regions are Central Coast, Sonoma, Columbia Valley, and Napa (Figure 8).
Column "region 2" has 79460 nan values.

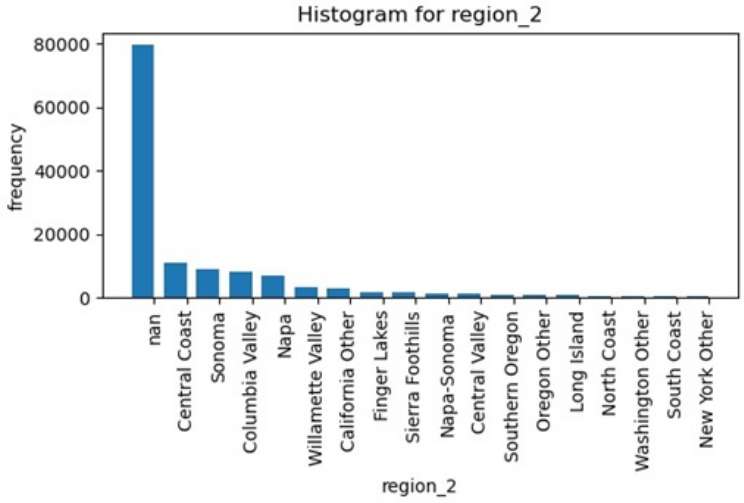

Figure 8: Histogram with values from the "region 2" column

Column "taster name". The most popular taster is the Roger Voss, and the same level is the unknown name. And the second and third popular people are Michael Schachner and Kerin O'Keefe (shown people tastes more than 50 times) (Figure 9). Column "taster name" has 26244 nan values.

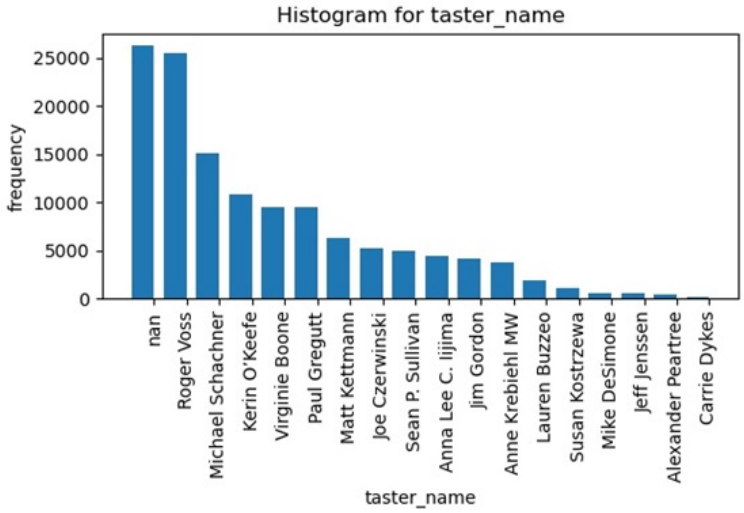

Figure 9: Histogram with values from the "taster name" column

Column "taster twitter handle". This data correlated with the taster name. The most popular Twitter handles are @vossroger for Roger Voss, @wineschach

for Michael Schachner, and @kerinokeefe for Kerin O'Keefe (Figure 10).Column "taster twitter handle" has 31213 nan values.

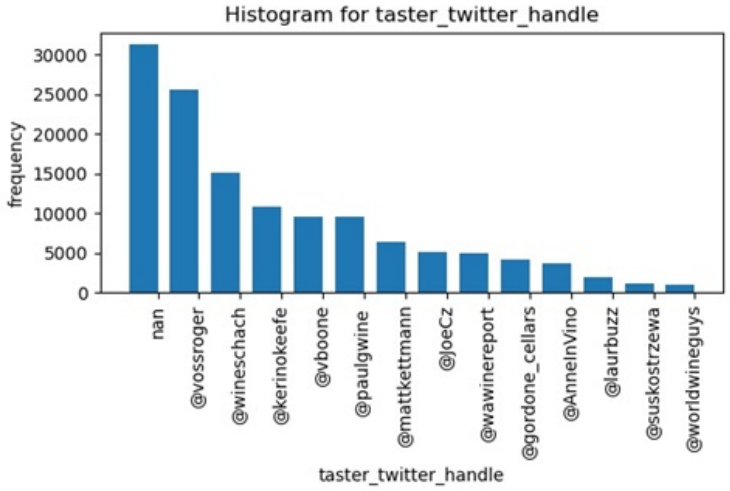

Figure 10: Histogram with values from the "histogram twitter handle" column

Column "variety". More than 60 gives us not readable data (Figure 11).

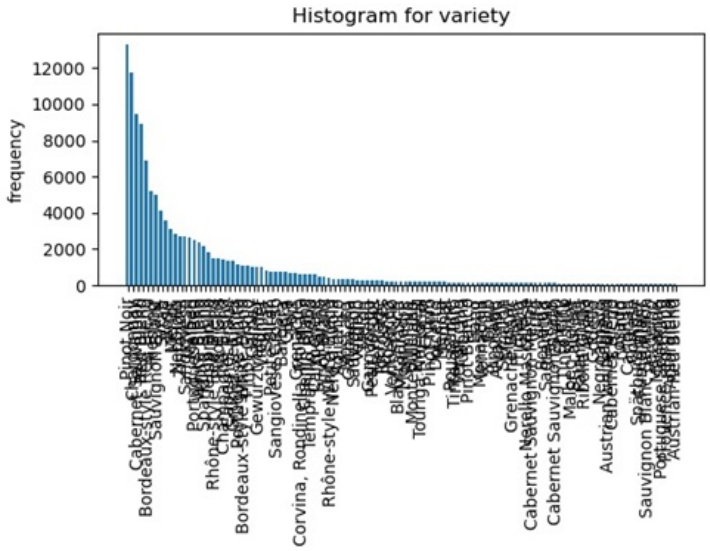

Figure 11: Histogram with values from the "histogram variety" column

The most popular varieties are Pinot Noir, Chardonnay, Cabernet Sauvignon (Figure 12). Column "variety" has 1 nan values.

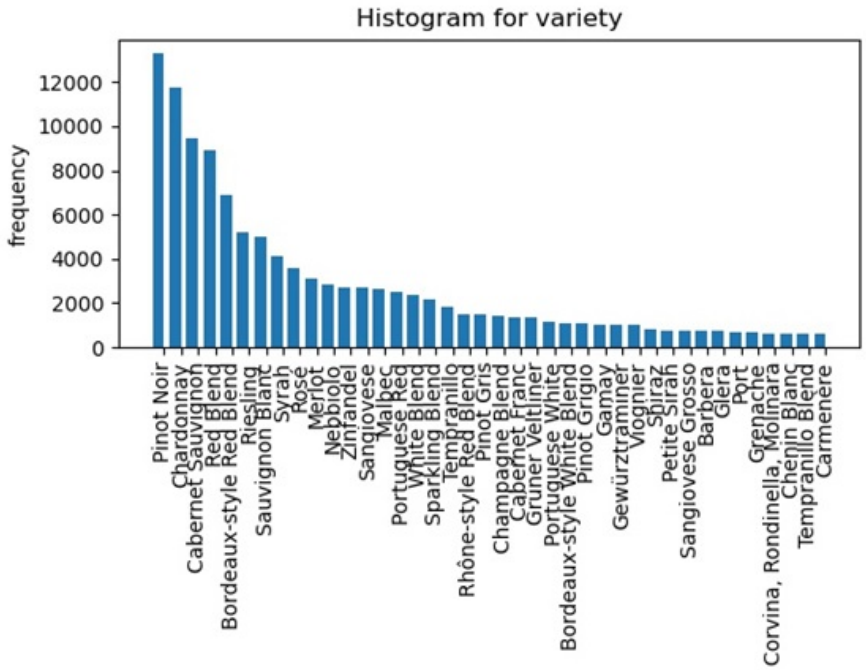

Figure 12: Histogram with values from the "histogram variety" column

Column "winery". The frequent wineries are Wines and Winemakers, Testarossa, and DFJ Vinhos (Figure 13).
Column "winery" has 0 nan values.

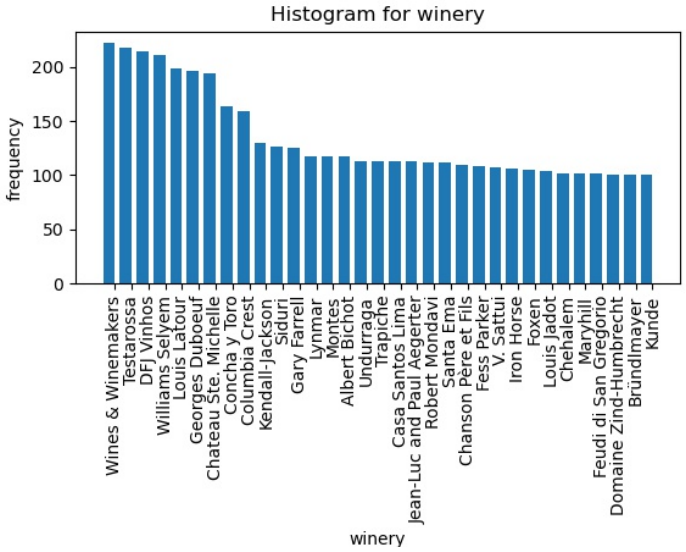

Figure 13: Histogram with values from the "histogram winery" column

Column "year". Most of the wines were produced in the 21st century (2000s) (Figure 14). Column 13 with name "year" has 4269 nan values.

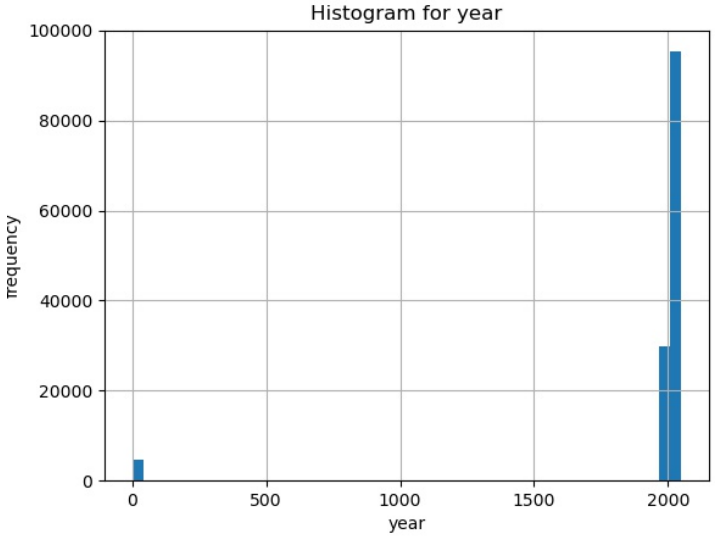

Figure 14: Histogram with values from the "histogram year" column

Column "designation"
The frequent designations are Reserve, Estate, and Reserva (Figure 15). Column "designation" has 37465 nan values.

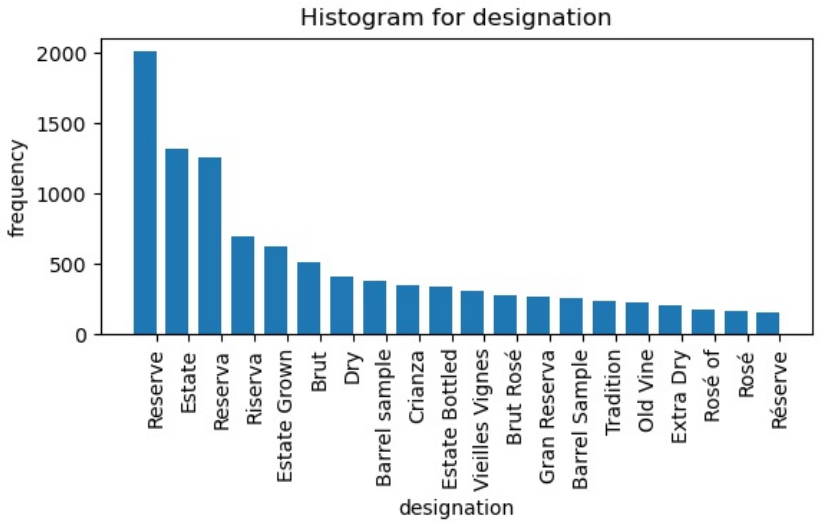

Figure 15: Histogram with values from the "histogram designation" column

The columns "description" and "title" consist of sentences, that's why no histograms are made for them.
Column "description" has 0 nan values.
Column "title" has 0 nan values.

d. We have computed statistics for columns containing numeric values. These include the minimum, maximum, and mean values, standard deviation, and median (Table 1).

	Minimal value	Maximal value	Average value	Standard deviation	Median value	Number of NAN values
points	80	100	88.447	3.04	88	0
price	4.0	3300	35.363	41.022	25	8996
year	1000	8000	2011.148	70.192	2011	4609

Table 1: Statistics for columns containing numeric values.

e. Before converting dataset into vectors, preprocessing of the dataset is done. Firstly, we removed duplicates in dataset. The dataset has NAN values in 9 columns out of 14.

After removing duplicates, there are fewer NAN values left:
Column 0 with name "country" has 59 nan values;
Column 2 with name "designation" has 34545 nan values;
Column 4 with name "price" has 8395 nan values;
Column 5 with name "province" has 59 nan values;
Column 6 with name "region 1" has 19560 nan values;
Column 7 with name "region 2" has 73219 nan values;
Column 8 with name "taster name" has 24917 nan values;
Column 9 with name "taster twitter handle" has 29446 nan values;
Column 11 with name "variety" has 1 nan values;
Column 13 with name "year" has 4269 nan values.

For the numeric values columns "price" and "year" NAN values were filled with the median. We utilized Imputation transformer for completing missing values by using the SimpleImputer class from sklearn.impute package. We also additionally pre-examined the options for using the "mean" and "most frequently" methods and based on the results obtained, we came to the conclusion that mathematically the most acceptable method for filling in the missing values in the dataset would be the method "median". The reason is outliers in the data set can affect the mean, leading it to not accurately represent all the scores. In this case, median is a better measure since outliers do not affect it.
The "region 2" column has been removed because it has 73219 NAN values.
All other textual NAN values are filled with the same the SimpleImputer class by using "constant value" filling NAN with "missing value" numbered 1, 2, 3 according to the column number. Since the OneHot algorithm is self-written, before feeding text columns, we replaced Nan values with missing values with a row index so that Nan values are not taken into account in the N most frequent values of this column.
Further, before the vectorization procedure, we converted the text to lowercase, then removed punctuation marks, after which we deleted Stop words as well. Removed the "title" column because it consists of values from different columns (winery year designation (region 1)).

Transformation of data from a dataset to a vector of real numbers has been implemented. We used two methods to transform the data: One-Hot encoding

and Doc2Vec.
NAN strings that are filled with missing value are converted to zero vector. Tables 2 and 3 show how many unaccounted for total values and how many of these all unaccounted values belong to the former Nan values in percent.
One-hot encoding makes our data more expressive. By using numeric values, it is easier for us to determine the likelihood of our values. One-hot encoding is used for the columns below:

	Number of unique values before data processing	Number of rows	Number of frequent unique values counted in a column	Unaccounted data from dataset (%)	Unaccounted NAN from dataset (%)
country	44	119988	40	0.051	0.049
points	21	119988			
price	8785	119988	150	0.48	
province	426	119988	200	0.825	0.049
region 1	1230	119988	500	19.631	16.302
taster name	20	119988	18	20.771	20.766
taster twitter handle	16	119988	15	24.541	24.541
year	4366	119988	30	0.084	
variety	708	119988	150	3.198	0.0008

Table 2: Table of counted and unaccounted values by columns in the method one hot encoding

To transform column sentences to vectors, we used Doc2vec:

	Number of unique values before data processing	Number of rows	Chosen dimensionality of Doc2Vec vector
description	119955	119988	700
designation	37980	119988	300
winery	16757	119988	600

Table 3: Table of counted and unaccounted values by columns in the method Doc2Vec

f. One-hot encoding is used for the "variety" column. One-Hot encoding on new data will give a zero vector of a predetermined dimension.

To improve prediction on our model, we can use Bagging. Bagging will allow us to reduce the variance of the trained classifier, by decreasing the value, it prevents overfitting. Bagging efficiency is achieved due to the fact that the basic algorithms trained on different subsamples are quite different, and their errors are mutually compensated, and also due to the fact that outlier objects may not fall into some training subsamples.

2 Task 2

b. Exploring the relationship between columns in our dataset. b.2. In Figures 16, 17 a scatter plot is drawn for each column, showing the predicted value of the column, p (v) on the x-axis, and the corresponding value of the column of points on the y-axis.

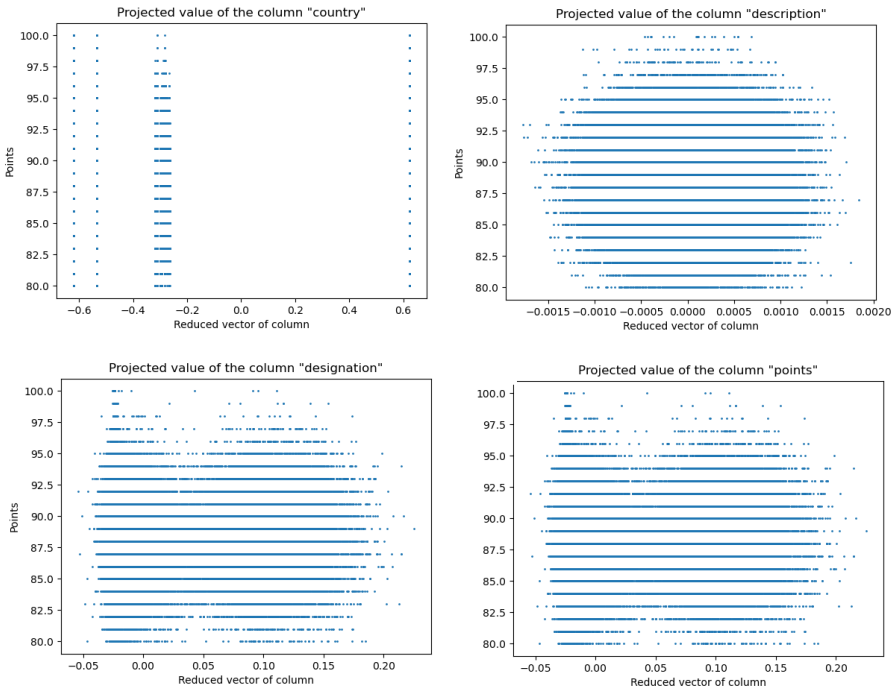

Figure 16: The predicted value of the column

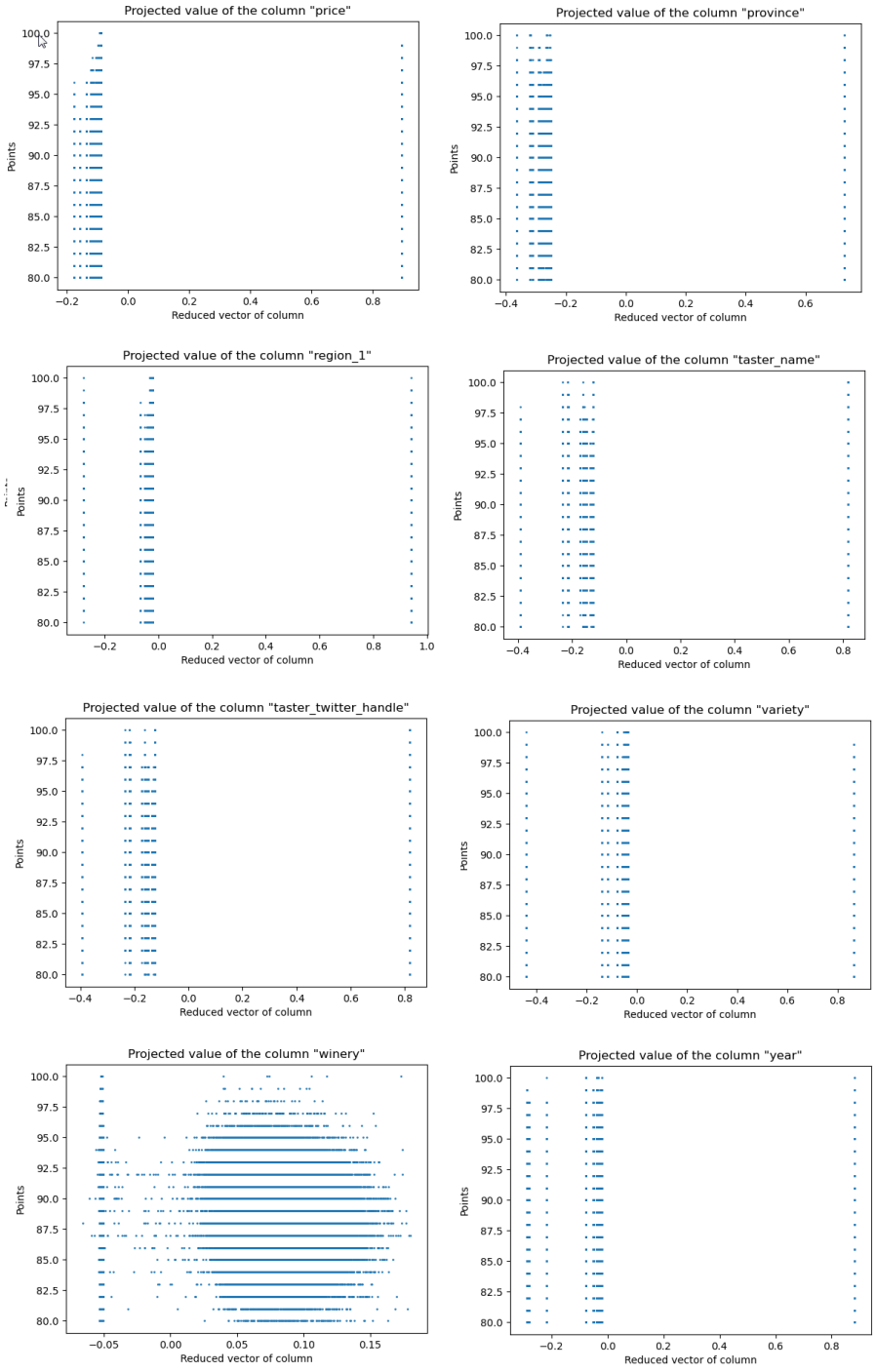

Figure 17: The predicted value of the column

b.3. Figures 18, 19 show the projected data, predicted data after a ridge regression for each column and regression line in each case.

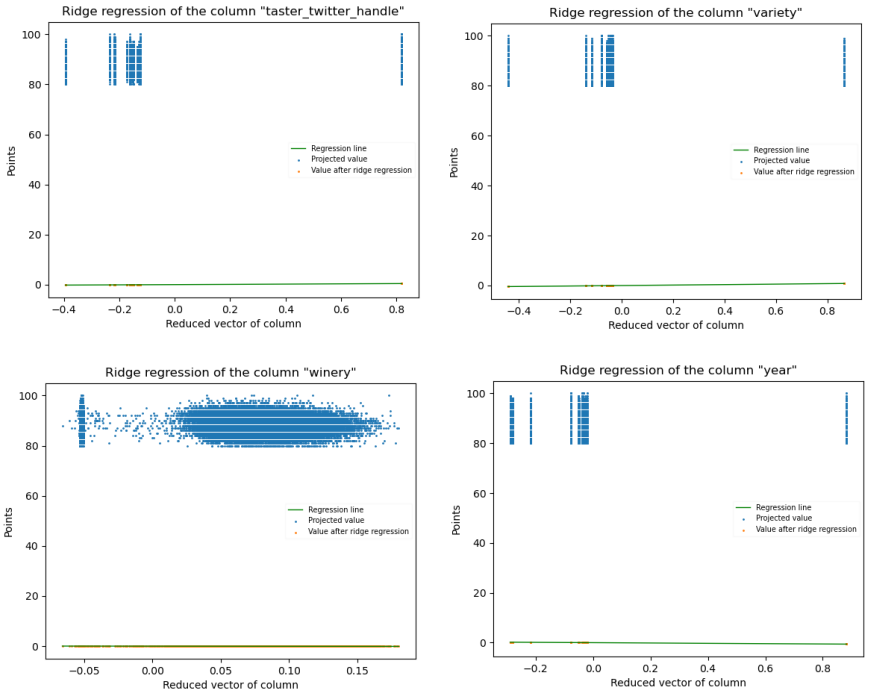

Figure 18: The predicted value of the column with a linear function in two dimensions

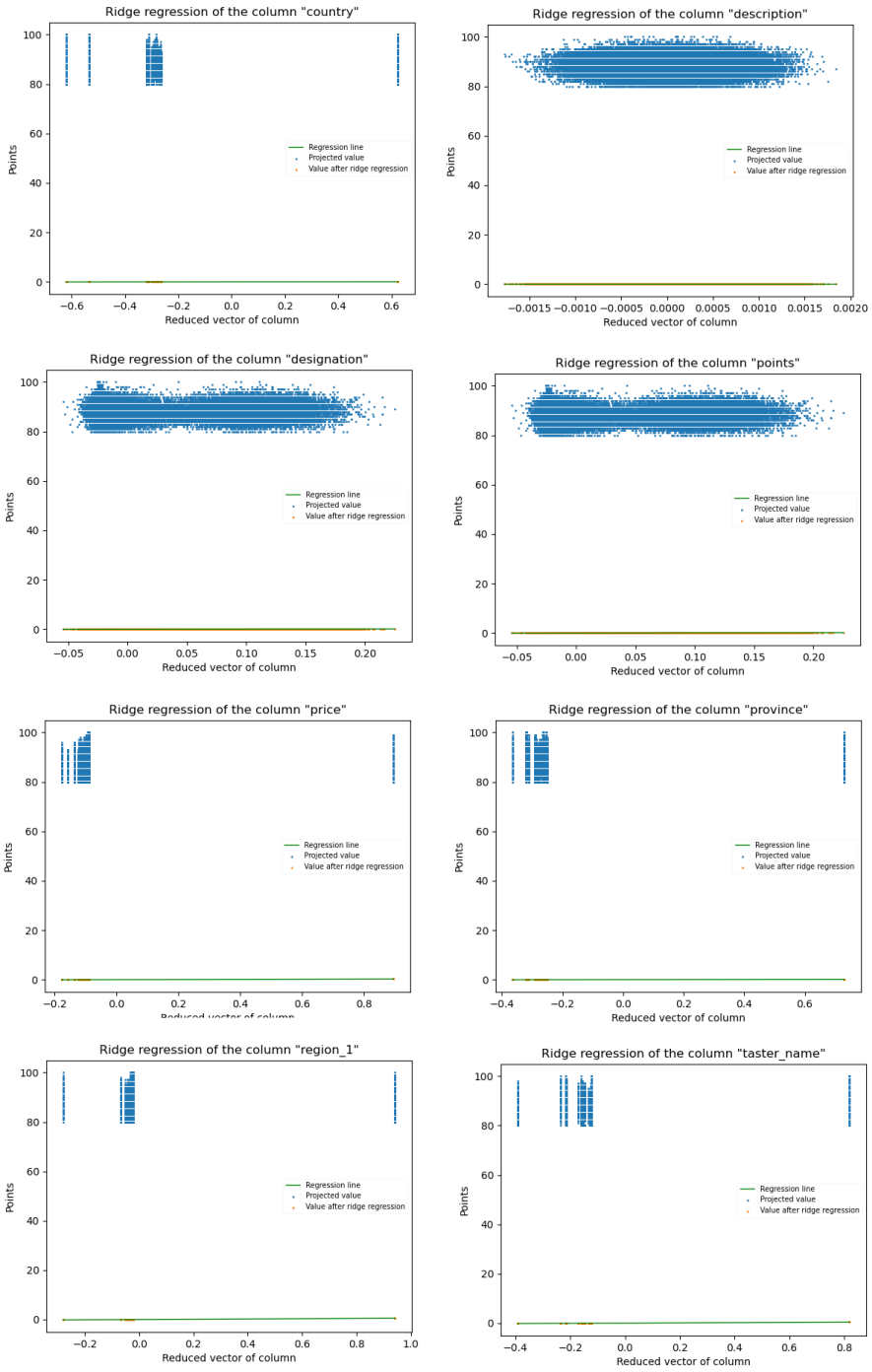

Figure 19: The predicted value of the column with a linear function in two dimensions

c. Figures 20, 21, 22, 23 show the result of the created RidgeRegressionBias class that uses the unchanged class from a) as the black box for the ridge regression with bias. For each column, two versions of the plot are created. The second version 2 shows the slope of the regression with bias.

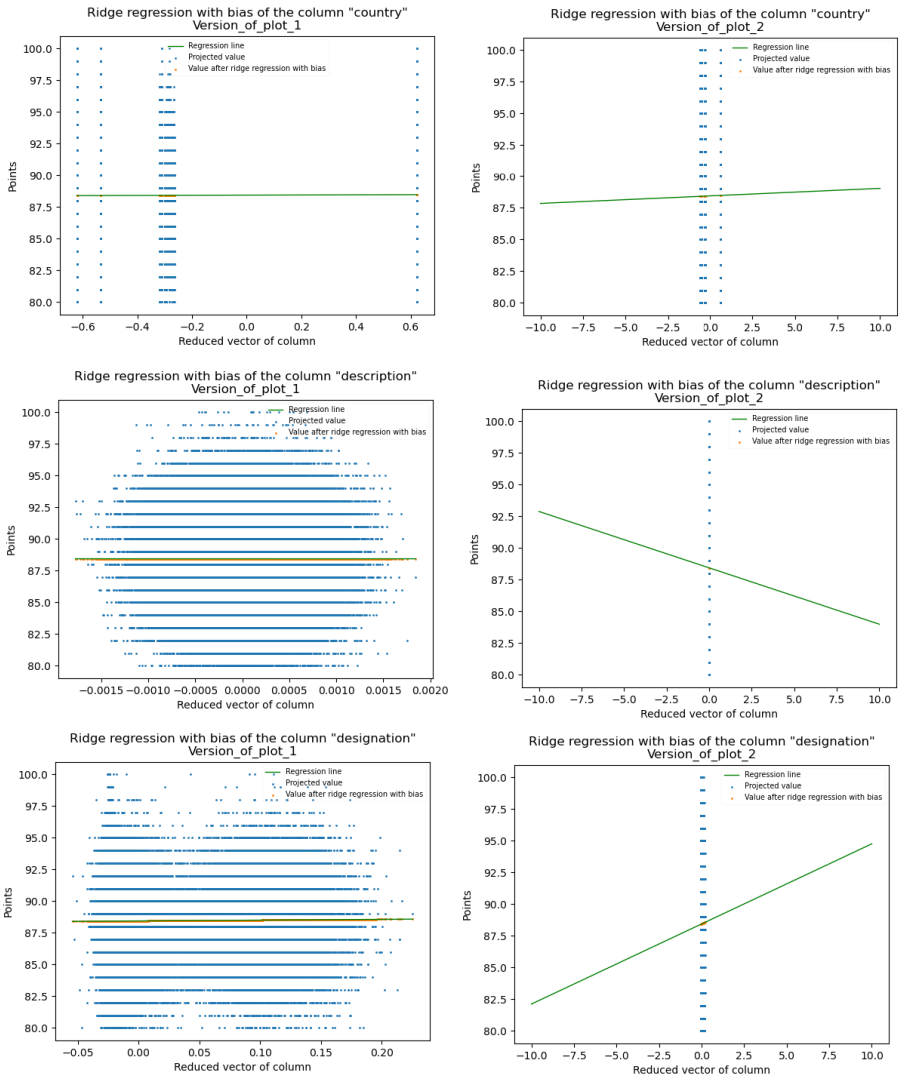

Figure 20: The predicted value of the column with bias

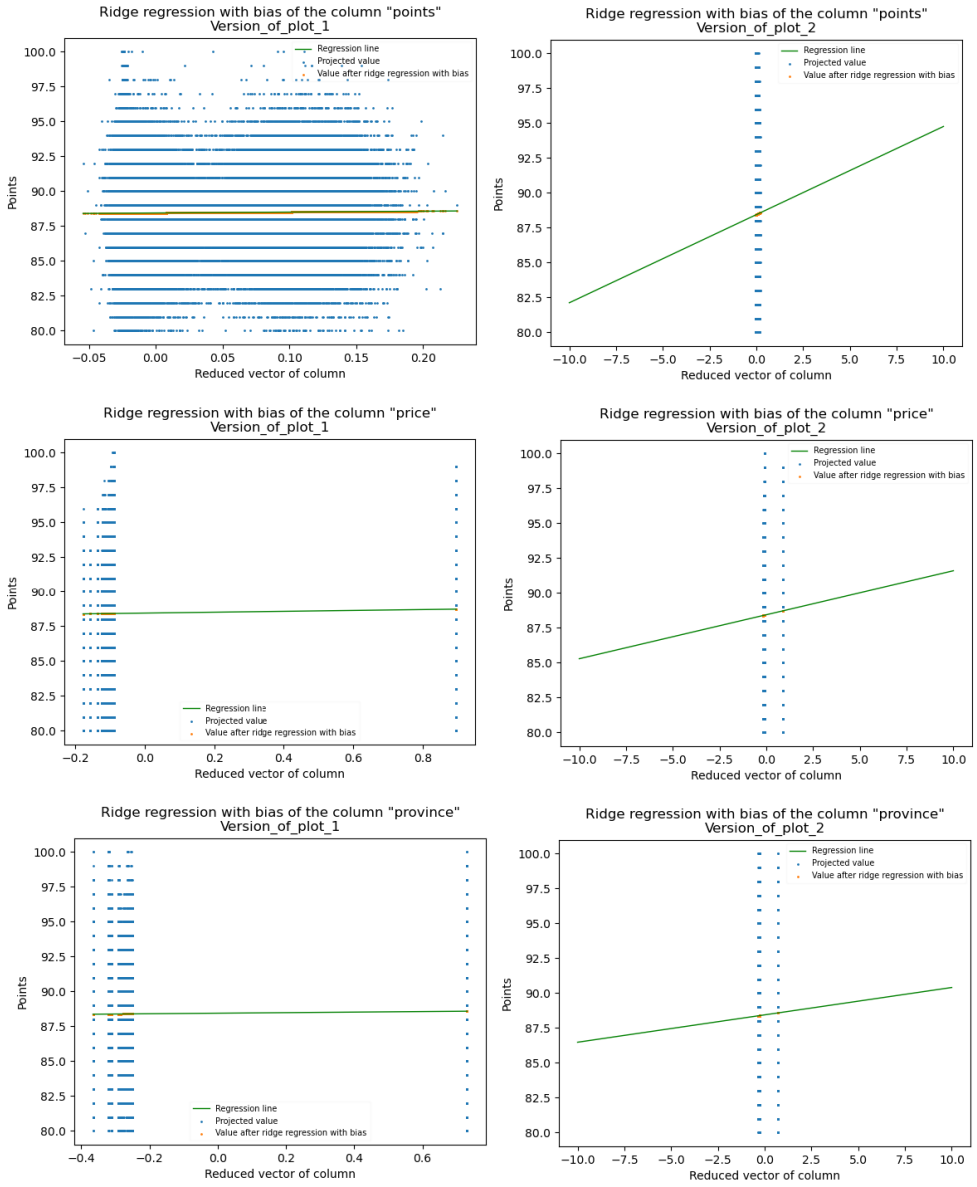

Figure 21: The predicted value of the column with bias

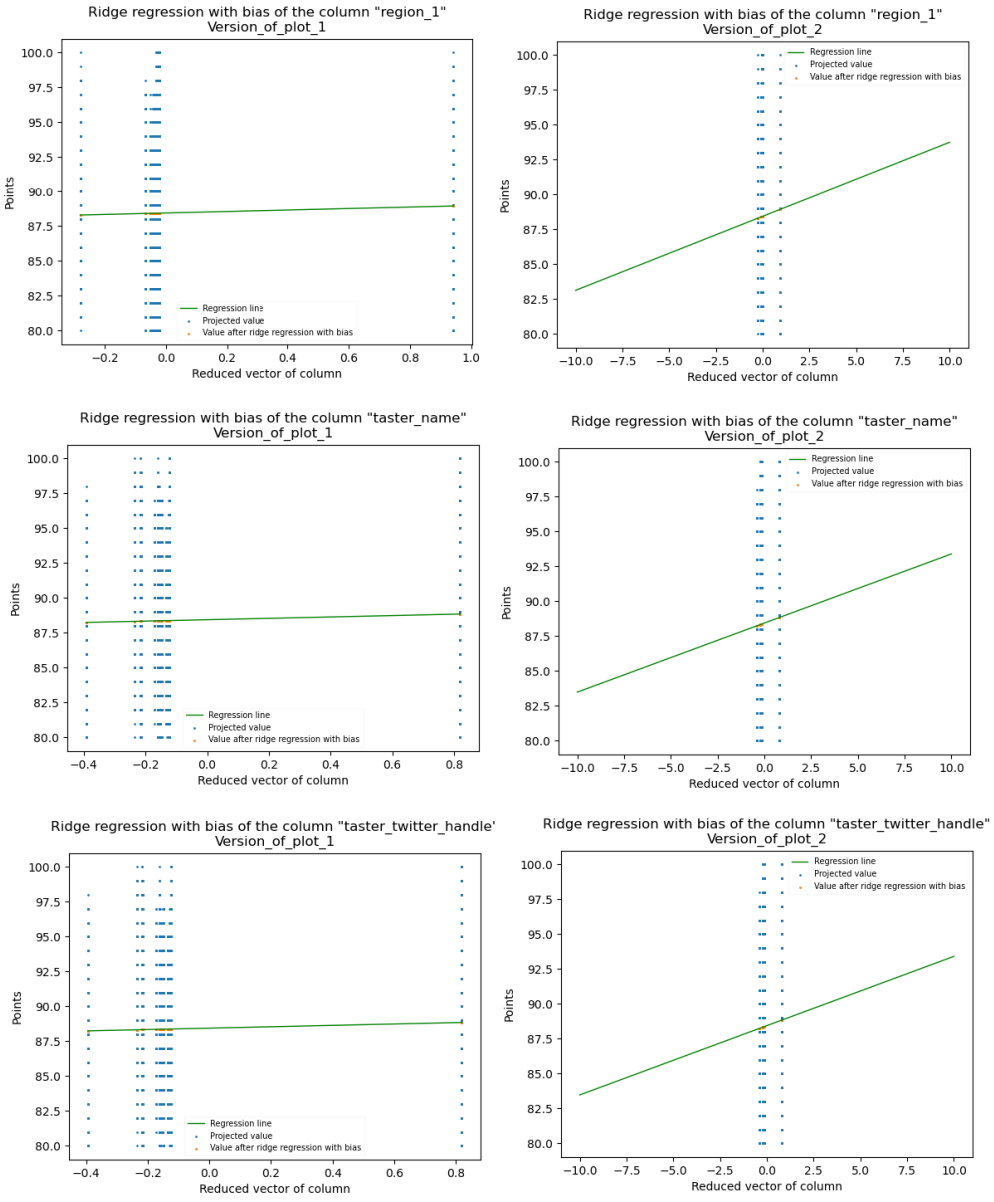

Figure 22: The predicted value of the column with bias

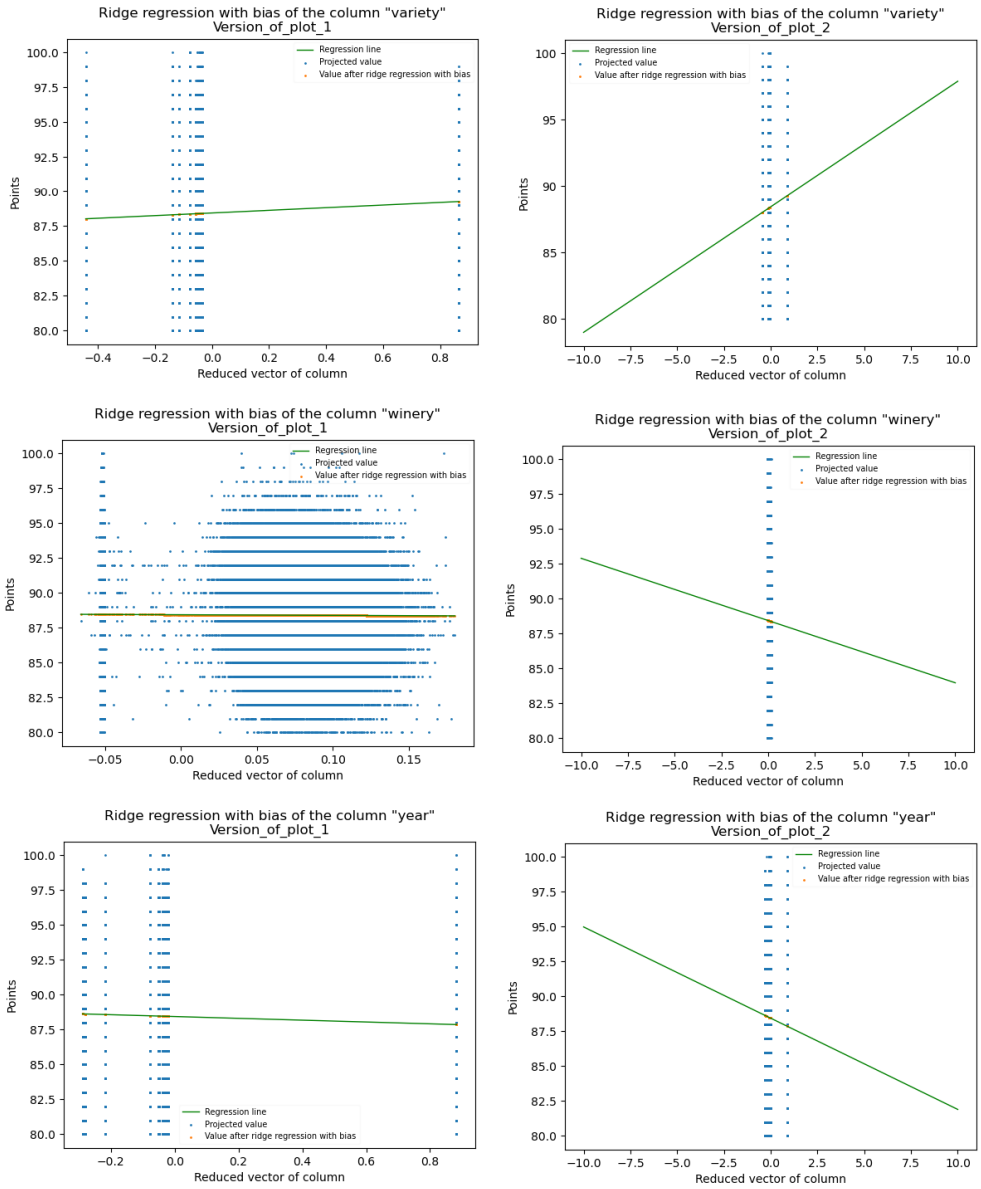

Figure 23: The predicted value of the column with bias

d. After applying improved ridge regression to the non-projected version of the entire wine review data. We used 5-fold cross-validation and the mean square error (MSE) as a measure of performance. MSE of our ridge regression is equal 4.883306313933569.

e. If the corresponding feature takes values in a rather big interval, its effect on the final prediction will be disproportionately large.
Therefore, to solve this problem, we can use data normalization or standardization. For this task, the following methods were used with the following results:
1. PowerTransformer: yeo-johnson method [1] [2]: MSE = 4.878919595502275;
2. QuantileTransformer [3]: MSE = 4.879514279821764;
3. MinMaxScaler [6]: MSE = 4.881273866888382;
4. Z-score normalization [4]: MSE = 4.885070366822445;
5. L2 normalization[5]: MSE = 4.900820332214457;
6. L1 normalization[5]: MSE = 5.204246552438361.

f. Forward-Stepwise Selection:
k=1 MSE=6.135010618783397 formed from columns: price;
k=2 MSE=5.624512618648486 formed from columns: price region_1;
k=3 MSE=5.266522423329922 formed from columns: price region_1 taster_name;
k=4 MSE=5.118508326434159 formed from columns: price region_1 taster_name winery;
k=5 MSE=5.029445491306445 formed from columns: price region_1 taster_name winery variety.

g. 1, 2) Figure data_2g and regression lines without using the fi function (Figure 24). Ridge Regression without changing data_2g MSE: 313492.8493221295

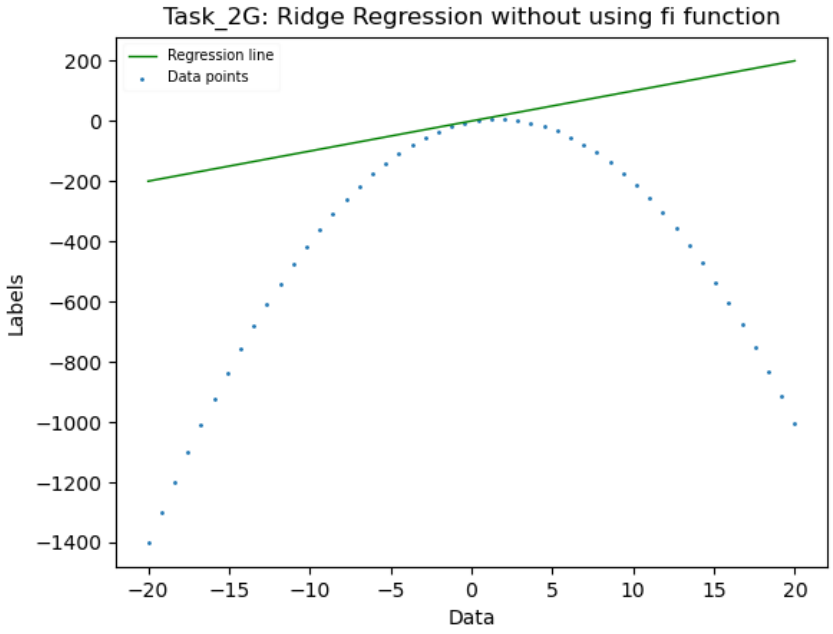

Figure 24: data_2g and regression lines without using the fi function

g. 3, 4) We chose function:

$$\phi\_data = \phi(data\_2g) = a * (x^2) + b * x + c$$

We used brute force to calculate the parameters of the function (Figure 25). So we got:
a = -1.5
b = 5
c = -1
Our function has become such a format:

$$\phi\_data = \phi(data\_2g) = -1.5 * (x^2) + 5 * x - 1$$

Where X is our data_2g data and fi_data is our data_2g data after fi transformation.

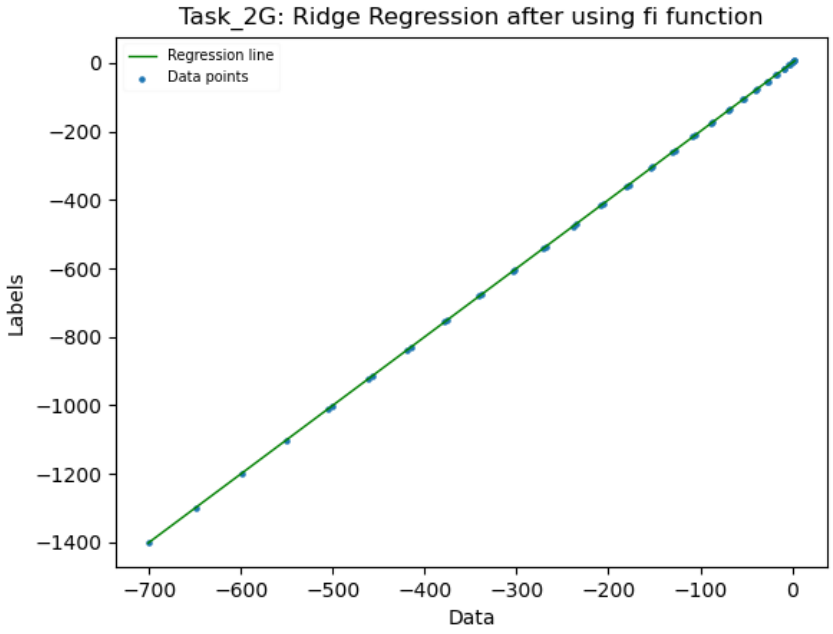

Figure 25: data_2g and regression lines after using the fi function

References

[1] Sklearn.preprocessing.PowerTransformer https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html#sklearn.preprocessing.PowerTransformer 2007-2021. [Online; accessed 5-January-2022]

[2] I.K. Yeo and R.A. Johnson *A new family of power transformations to improve normality or symmetry* Biometrika, 87(4), pp.954-959, 2000.

[3] Sklearn.preprocessing.QuantileTransformer https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.QuantileTransformer.html#sklearn.preprocessing.QuantileTransformer 2007-2021. [Online; accessed 5-January-2022]

[4] scipy.stats.zscore https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.zscore.html 2008-2021. [Online; accessed 5-January-2022]

[5] Scipy.stats.zscore https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.normalize.html#sklearn.preprocessing.normalize 2007-2021. [Online; accessed 5-January-2022]

[6] sklearn.preprocessing.MinMaxScaler https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html#sklearn.preprocessing.MinMaxScaler 2007-2021. [Online; accessed 5-January-2022]