CS513: Theory & Practice of Data Cleaning

Final Project Project Phase-II (Summer 2023)

University of Illinois, Urbana-Champaign

Team ID: Team 88

Git Repository: https://github.com/adamwm3/cs513-data-cleaning-final-project-team88

UoI Box: https://uofi.box.com/s/mxz30z18rnkzqd6jxl42hdt1nklcbyhr

Ahmed Elfarra	Ahmedse2@illinois.edu
Nishanth Alladi	Nalladi2@illinois.edu
Adam Michalsky	Adamwm3@illinois.edu

Use Case Overview:

Use Case: Which wines does a taster prefer based on the points awarded and region it was cultivated in?
Role: A skeptical wine enthusiast based out of the United States
Motivation: A wine enthusiast is skeptical about the ratings given to some wine produced at their local winery. They suspect that each taster has a bias for preferred varieties of wine.

Summary:

The winery-kaggle dataset primarily contained defects that consisted of missing values and incorrect data types. Our analysis from *Phase I* and the analysis documented in the *Phase II Jupyter Notebook* identified the columns country, province, region_1, designation, taster_name, and price contained empty values. Empty values were either removed or in some cases repaired in order to support U_1.

Data was removed from the dataset if it was deemed that the record was *"fit for purpose"*. In the context of this project, the data is *"fit for purpose"* only if the data was complete for key data points in U_1. The corresponding records were removed from the dataset if country, province, region_1, designation, or taster_name were empty in the raw dataset. In addition to the record removal we removed the region_2 and taster_twitter_handle columns since they didn't offer any distinguishing information that wasn't already represented in some way (e.g the value of region_1 is encompassing of region_2) or it just wasn't relevant for U_1.

The raw dataset contained 128,908 records in its initial state and after the data pruning exercise, the dataset had 75,036 records remaining. While the pruning removed 57% of the data entries, the reality is that ~55% of the data had issues with location related data which makes it unusable for U_1. Some records were removed during the pruning process for the designation field, however this was done in an effort to offer the most completed record.

Once the pruning was complete, data and data types were repaired. In the winery-kaggle dataset, only the points and prices fields required type casting to a numeric value. The price field also required a repair due to missing values. The missing values were addressed by training a simple model on the data and making predictions on the prices. While this repair is fabricating data, we felt that doing this type of repair was better than simply removing the records since price is not required for U_1.

In summation, the data set was made *"fit for purpose"* by performing data removal, or pruning, of records and columns and by repairing a limited number of records. We did have discussion on other repairs such as determining a region_1 value based on the contents of the designation field but we lacked the ability to validate the inferred values. The inability to validate

the data after more complicated repairs led us to decide to not perform the repairs. This can be a further improvement on the dataset given the proper resources and time.

Data Analysis and Repair:

Data analysis in Phase I was performed using OpenRefine and Jupyter notebook. For Phase II we opted to leverage the jupyter notebook and it is available in the git repository listed below along with raw and clean versions of the dataset. The notebook contains most of the code used for the analysis and the narrative is available in *Detailed Analysis* below. All other supplementary materials are shared in the repository as well.

Github: https://github.com/adamwm3/cs513-data-cleaning-final-project-team88
UoI Box: https://uofi.box.com/s/mxz30z18rnkzqd6jxl42hdt1nklcbyhr

Raw Data Statistics:

Dimension	**Count**
Record	129,971
Column	13

Defect	**Action Performed**	**Metrics**
No value provided for the field country.	Removed the record.	Both defects were addressed in the same step. Record: 129,908 (63) Column: 13 (0)
No value provided for the field province.	Removed the record.	
No value provided for the field region_2.	Removed the column..	Record: 129,908 (0) Column: 12 (1)
No value provided for the field region_1.	Removed the record..	Record: 108,724 (21,184) Column: 12 (0)
No value provided for the field designation.	Removed the record..	Record: 75,036 (33,688) Column: 12 (0)
No value provided for the field	Repaired the record.	Records Updated: 2070

taster_name.		Record: 75,036 (0) Column: 12 (0)
The column taster_twitter_handle is not required for U_1.	Removed the column	Record: 75,036 (0) Column: 11 (1)
No value provided for the field price.	Repaired the record.	Records Updated: 4861 Record: 75,036 (0) Column: 11 (0)
The field, price, is not the correct data type.	Converted to numeric value.	All records affected. Record: 75,036 (0) Column: 11 (0)
The field, points, is not the correct data type.	Converted to numeric value.	All records affected. Record: 75,036 (0) Column: 11 (0)

IC-Violation Report Function:

These functions are included in the jupyter notebook and documented separately as *ic-violation-report.txt*.

```
def report_empty_value_count(df, fieldname):
  try:
    print("FIELD - country: " + str(df[fieldname].isna().sum()))
  except KeyError:
    print("FIELD - " + fieldname + " has been removed.")

def ic_violation_report(df):
  #Banner
  print("==============START IC VIOLATION REPORT==============")
  #Report Dataframe Summary
  print("DATAFRAME SUMMARY: ")
  print("-----------------------------------------------------")
  df.info()
  print("\n")
  print("FIELD EMPTY VALUE REPORT: ")
  print("-----------------------------------------------------")
  #Check NA country violations
```

```
report_empty_value_count(df, 'country')
#Check NA description violations
report_empty_value_count(df, 'description')
#Check NA designation violations
report_empty_value_count(df, 'designation')
#Check NA points
report_empty_value_count(df, 'points')
#Check NA price
report_empty_value_count(df, 'price')
#Check NA province violations
report_empty_value_count(df, 'province')
#Check NA region_1 violations
report_empty_value_count(df, 'region_1')
#Check NA region_2 violations
report_empty_value_count(df, 'region_2')
#Check NA tastername
report_empty_value_count(df, 'taster_name')
#Check NA taster_twitter_handle
report_empty_value_count(df, 'taster_twitter_handle')
#Check NA title
report_empty_value_count(df, 'title')
#Check NA variety
report_empty_value_count(df, 'variety')
#Check NA winery
report_empty_value_count(df, 'winery')

#End Banner
print("==============END IC VIOLATION REPORT==============","\n")
```

Raw Dataset Report:

```
==============START IC VIOLATION REPORT==============
DATAFRAME SUMMARY:
------------------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
Int64Index: 129971 entries, 0 to 129970
Data columns (total 13 columns):
 #   Column                 Non-Null Count   Dtype
---  ------                 --------------   -----
 0   country                129908 non-null  object
 1   description            129971 non-null  object
 2   designation            92506 non-null   object
 3   points                 129971 non-null  int64
 4   price                  120975 non-null  float64
 5   province               129908 non-null  object
 6   region_1               108724 non-null  object
 7   region_2               50511 non-null   object
 8   taster_name            103727 non-null  object
 9   taster_twitter_handle  98758 non-null   object
 10  title                  129971 non-null  object
 11  variety                129970 non-null  object
 12  winery                 129971 non-null  object
dtypes: float64(1), int64(1), object(11)
memory usage: 13.9+ MB

FIELD EMPTY VALUE REPORT:
------------------------------------------------------------
FIELD - country: 63
FIELD - country: 0
FIELD - country: 37465
FIELD - country: 0
FIELD - country: 8996
FIELD - country: 63
FIELD - country: 21247
FIELD - country: 79460
FIELD - country: 26244
FIELD - country: 31213
FIELD - country: 0
FIELD - country: 1
FIELD - country: 0
==============END IC VIOLATION REPORT==============
```

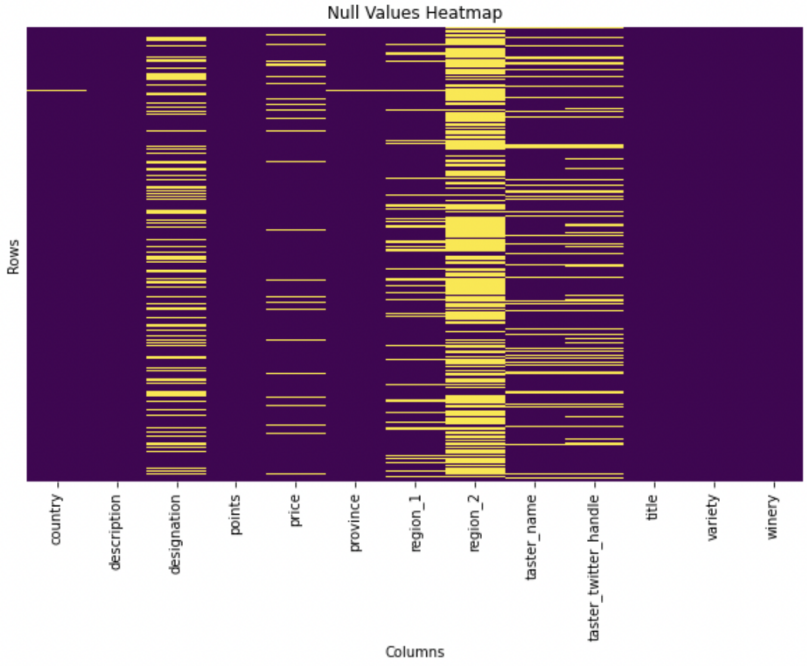
Processed Dataset Report:

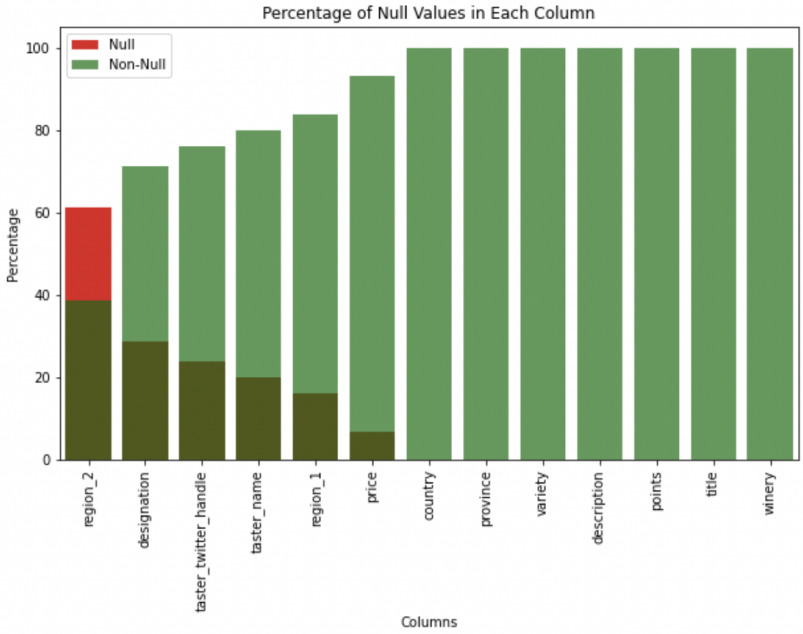
```
==============START IC VIOLATION REPORT==============
DATAFRAME SUMMARY:
------------------------------------------------------------
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 75036 entries, 0 to 75035
Data columns (total 11 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   country       75036 non-null  object
 1   description   75036 non-null  object
 2   designation   75036 non-null  object
 3   points        75036 non-null  int64
 4   price         75036 non-null  float64
 5   province      75036 non-null  object
 6   region_1      75036 non-null  object
 7   taster_name   75036 non-null  object
 8   title         75036 non-null  object
 9   variety       75036 non-null  object
 10  winery        75036 non-null  object
dtypes: float64(1), int64(1), object(9)
memory usage: 6.3+ MB

FIELD EMPTY VALUE REPORT:
------------------------------------------------------------
FIELD - country: 0
FIELD - country: 0
FIELD - country: 0
FIELD - country: 0
FIELD - country: 0
FIELD - country: 0
FIELD - country: 0
FIELD - region_2 has been removed.
FIELD - country: 0
FIELD - taster_twitter_handle has been removed.
FIELD - country: 0
FIELD - country: 0
FIELD - country: 0
==============END IC VIOLATION REPORT==============
```

Detailed Analysis:

To begin with, it is essential to define what constitutes missing values. Missing data arises when a particular value was not measured, cannot be measured, or has been lost due to errors. Missing values result in an incomplete dataset. It is not only crucial to identify the presence of missing values within a dataset but also imperative to determine the specific columns and the extent to which they contain missing values.

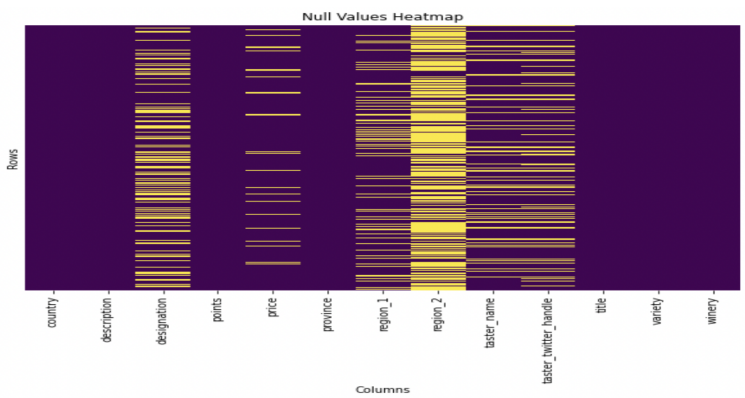


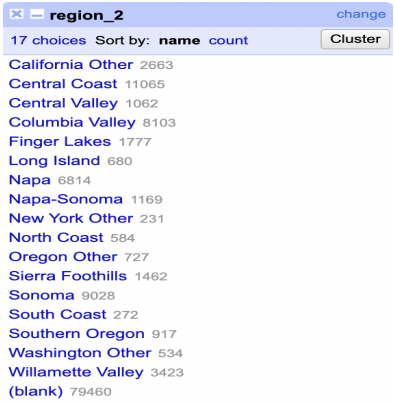


When faced with Null Values, several standard approaches can be employed:

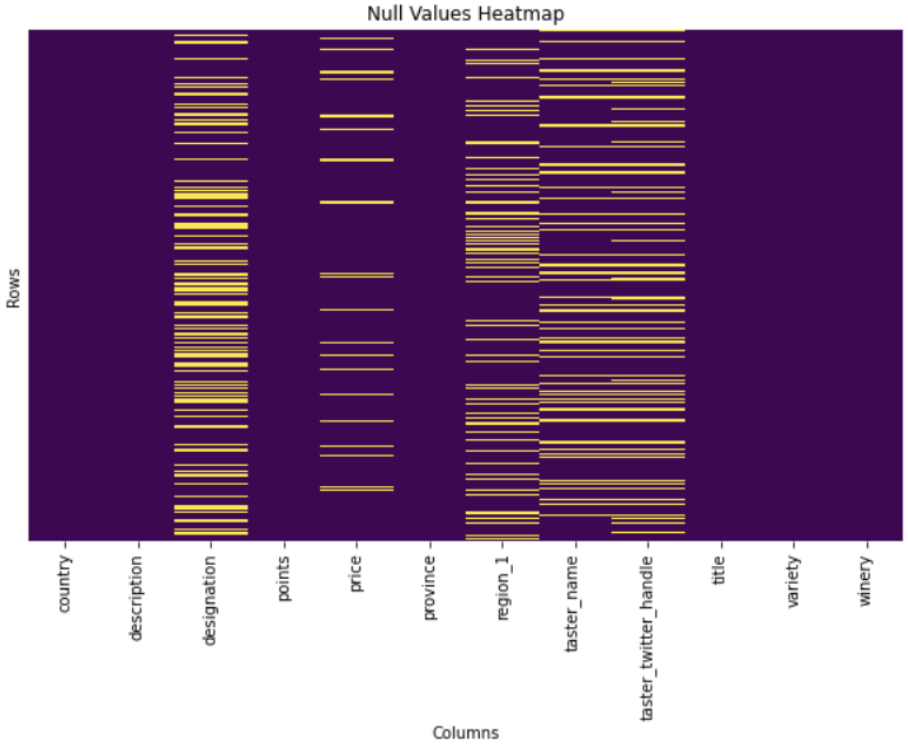
1. Discard rows with a significant number of missing values. However, this method should only be used if the missing values within the particular row are extensive.
1. Remove columns with a considerable number of missing values if the column's relevance is minimal, and the number of missing values is substantial.
2. Substitute missing values with the column average. For categorical features, the most common value can be used, while for numerical features, the median or mean of the column can serve as the average.
3. Utilize predictive models to replace missing values with predicted ones, employing Machine Learning Algorithms.
4. Replace missing values with zeros or 'Unknown' and treat them as regular values, if applicable.

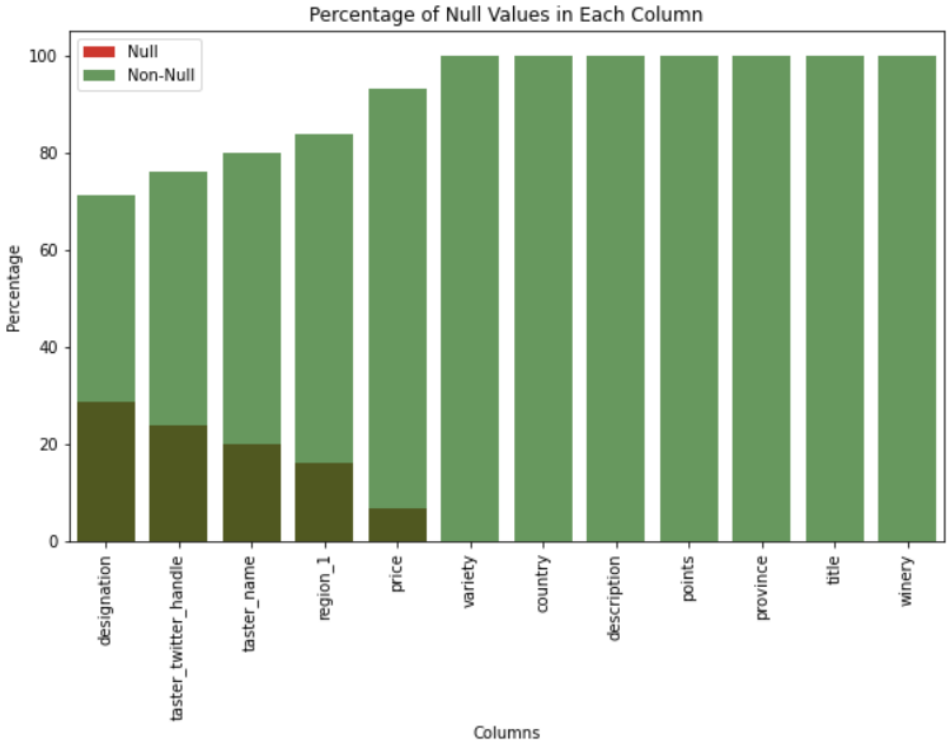
As evident from the presented plot and the OpenRefine history, it is noticeable that the columns 'country' and 'province' have only a small number of missing values. So we drop the blank rows of the dataset since it wont have too much of an effect on the rest of the dataset.



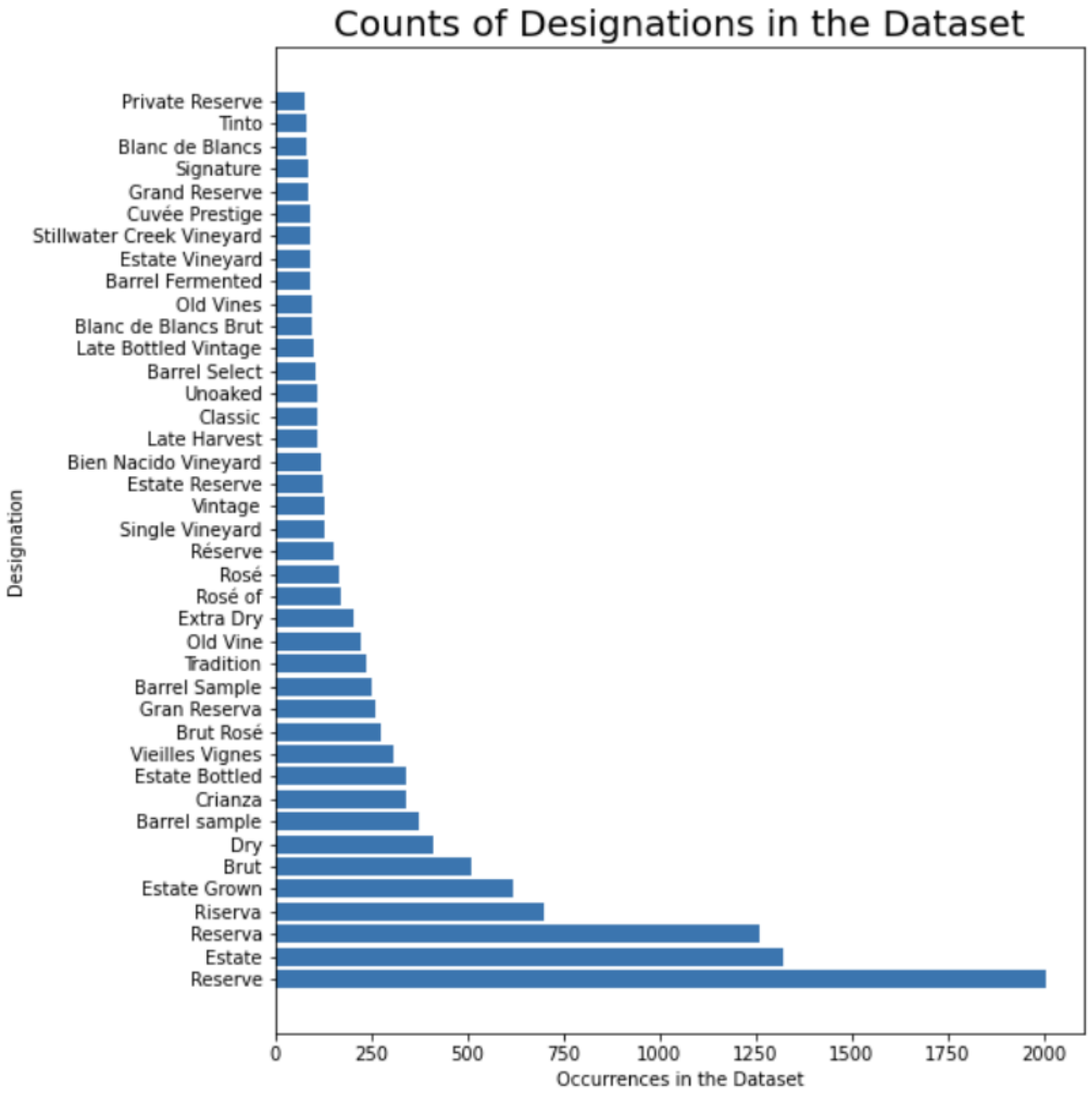


Due to the substantial number of Null Values in the 'region_2' column, accounting for 60% of the values, and its lack of significant information, we can safely eliminate the column from the dataset.

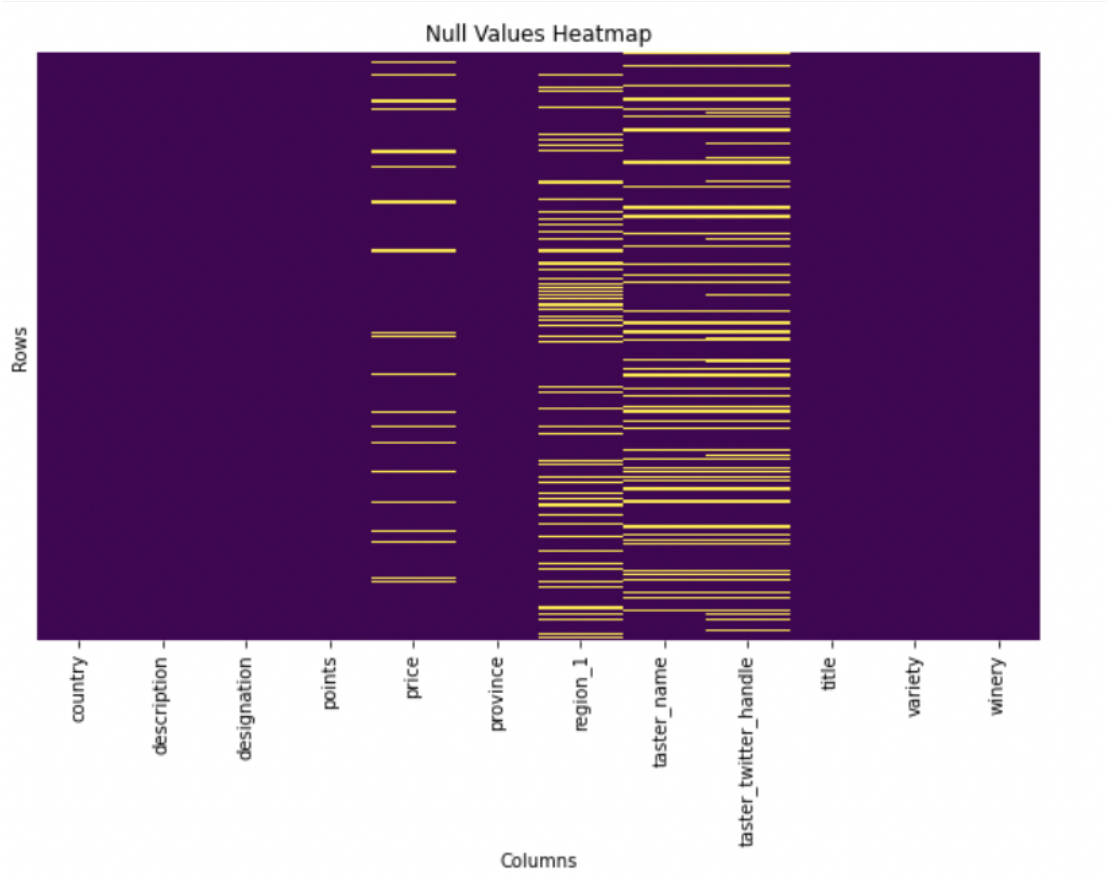


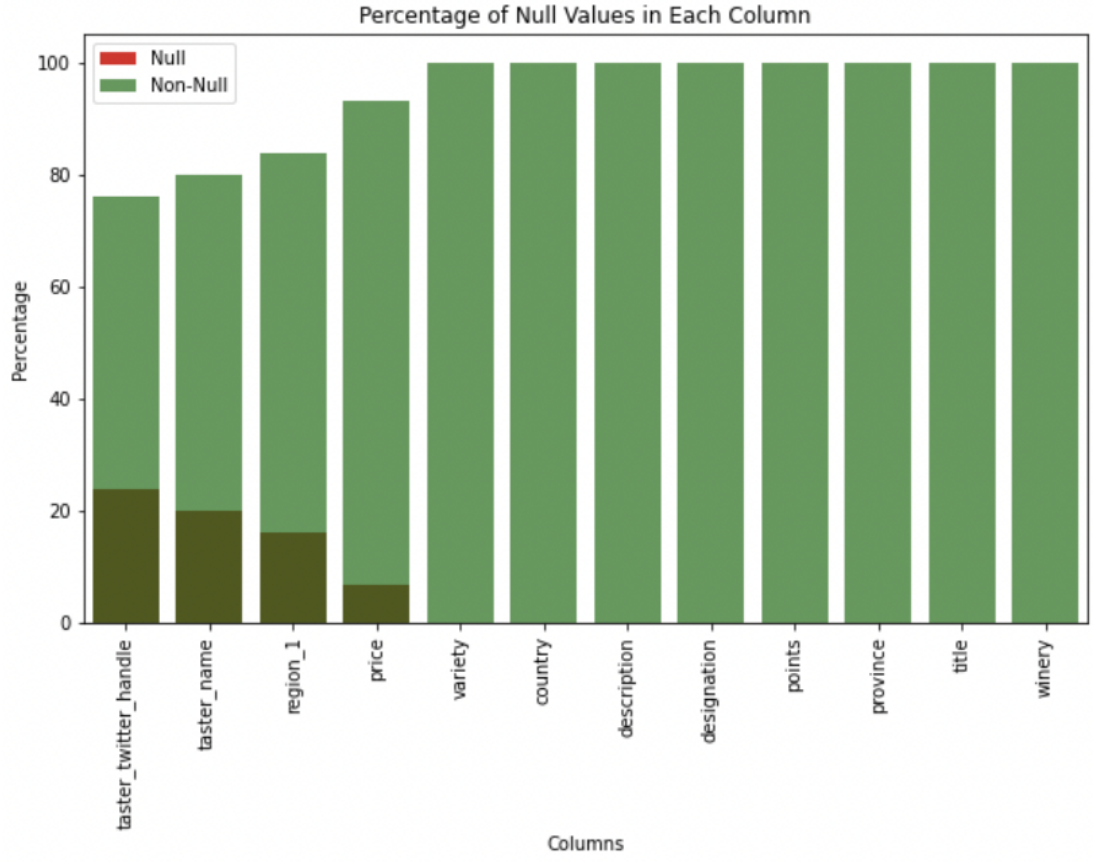


The 'designation' column exhibits the second highest number of Missing Values, with over 30% of its values being absent.

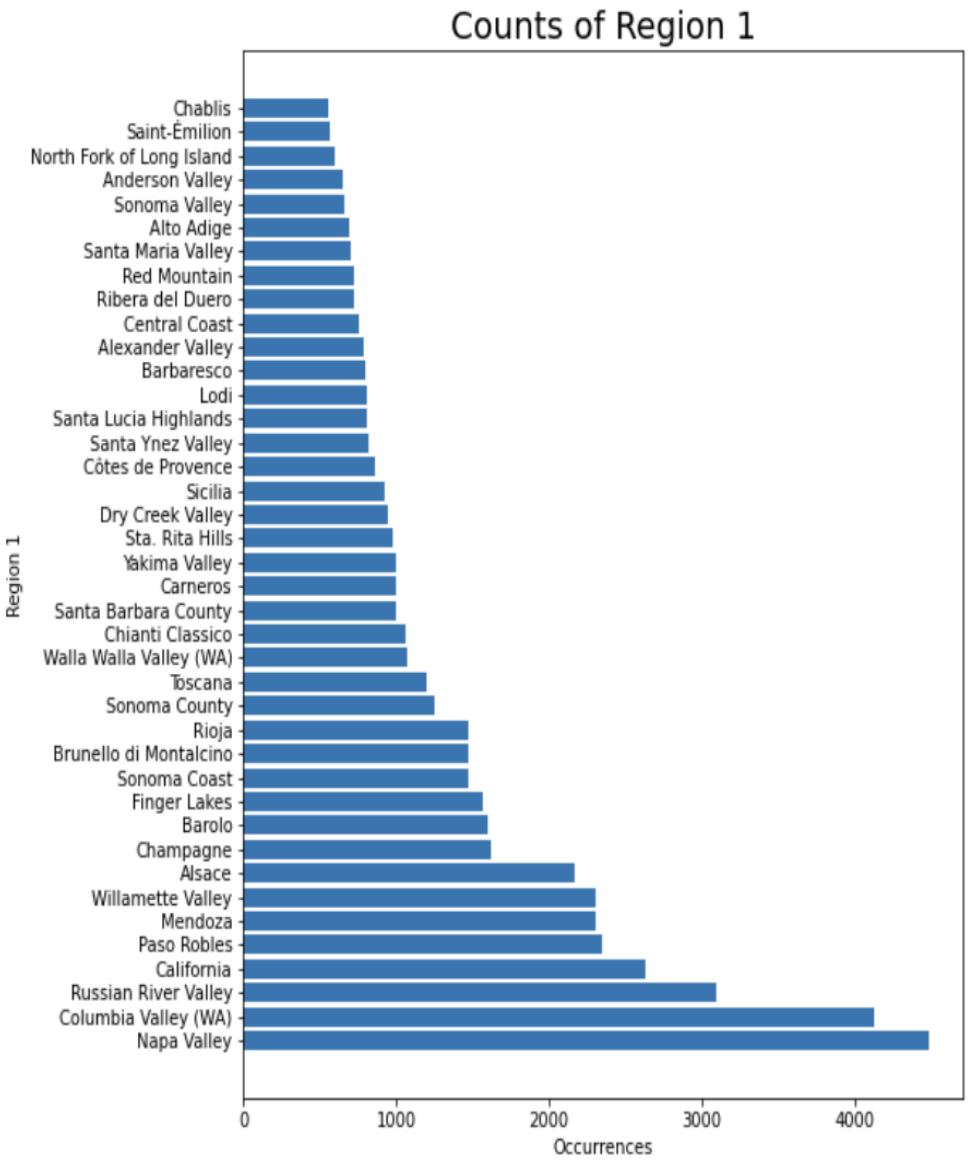


Frequently occurring values in this feature could hold valuable insights regarding its relationships with other features. Instead of dropping the column, we should address the missing values by selecting an appropriate replacement. However, the most prevalent label, 'Reserve,' appears in just over 1% of the rows, making it unsuitable for filling the gaps. An alternative approach is to handle missing values by replacing them with 'Unknown' and treating it on par with other known values

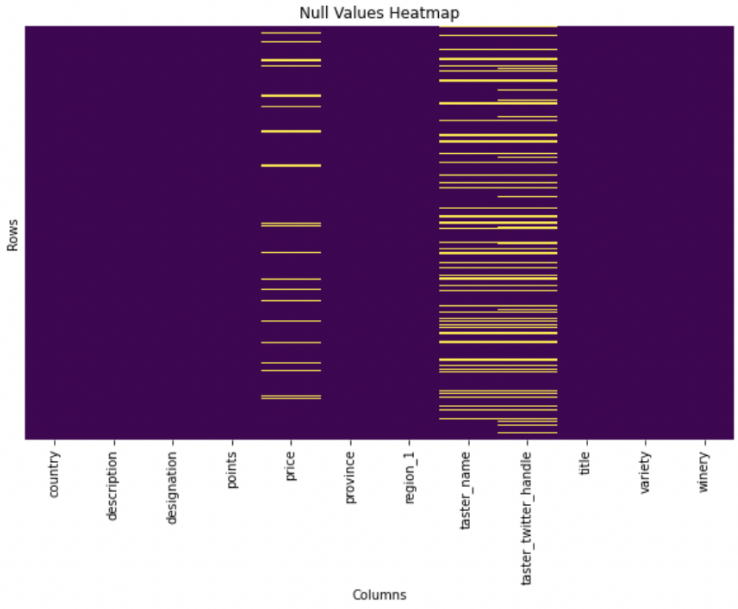

Null Values Heatmap
Rows
country
description
designation
points
price
province
region_1
taster_name
taster_twitter_handle
title
variety
winery
Columns

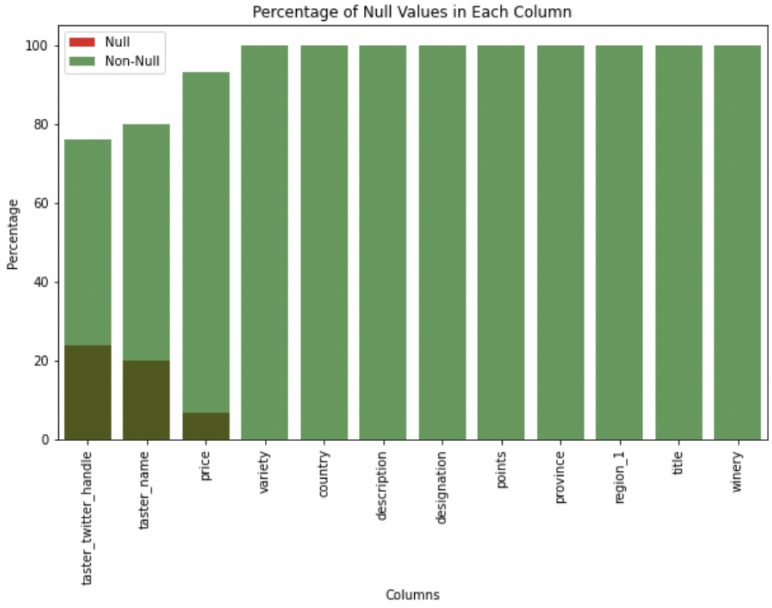

Percentage of Null Values in Each Column
Null
Non-Null
Percentage
0
20
40
60
80
100
taster_twitter_handle
taster_name
region_1
price
variety
country
description
designation
points
province
title
winery
Columns

Now examining the region_1 column, we can determine that it has a substantial amount of missing value, with high amounts of occurrences of values we can apply the same method as we did with the designation.



Yakima Valley 997
Yamhill County 2
Yarra Valley 111
Yecla 24
Yolo County 47
York Mountain 9
Yorkville Highlands 79
Yountville 92
Zonda Valley 3
(blank) 21247

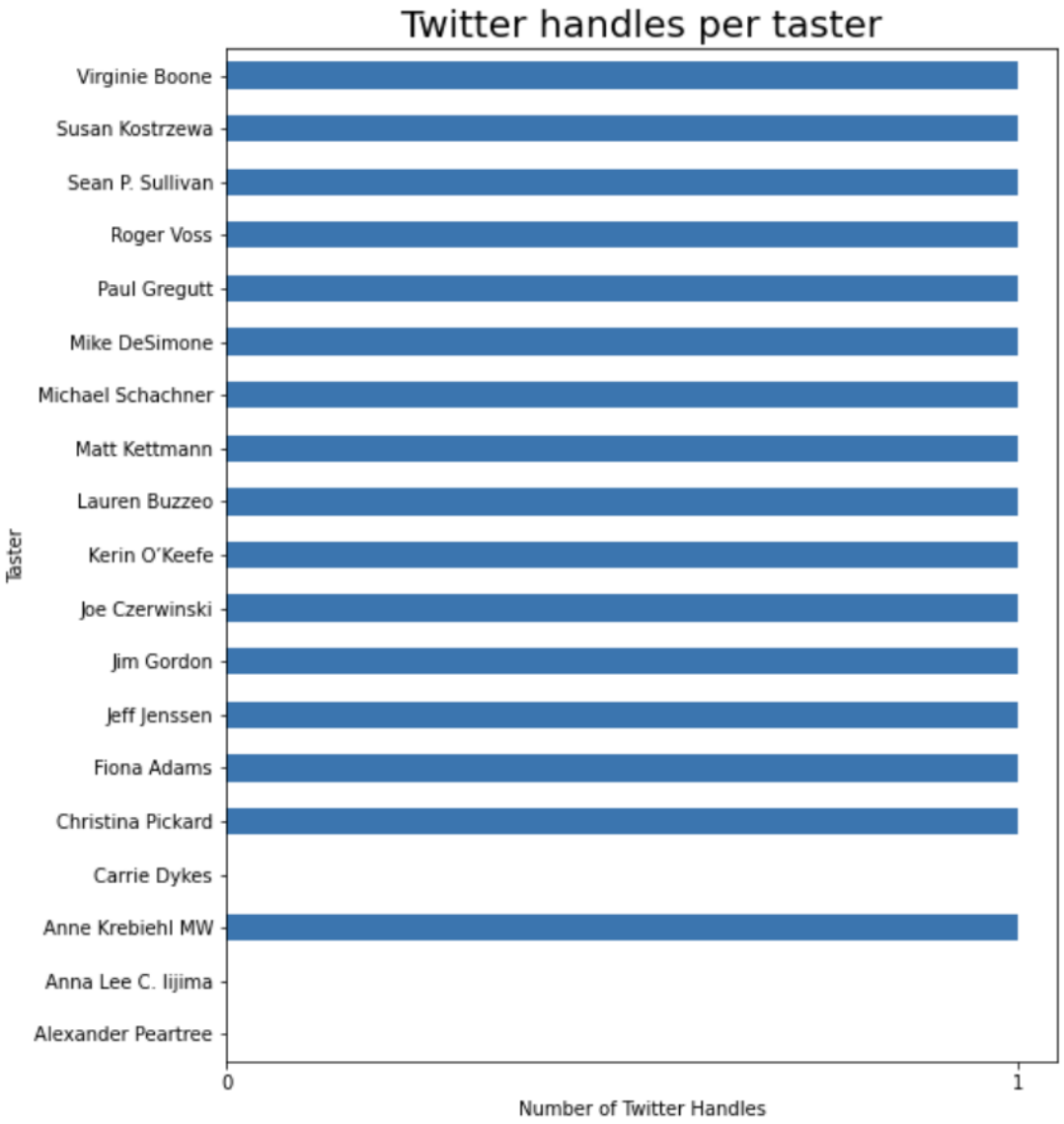




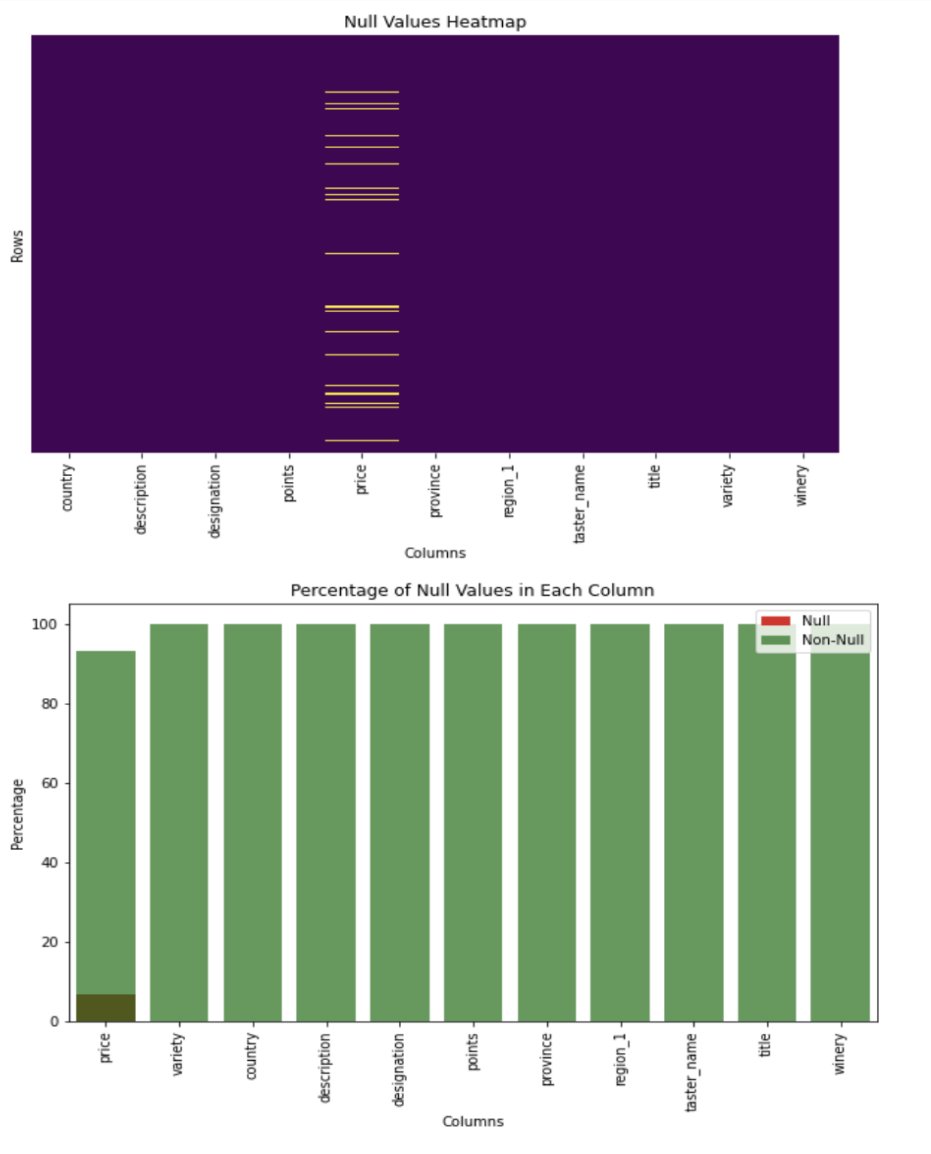
To address the 'taster_name' and 'taster_twitter_handle' columns, we need to gain insights into their relationship. Both columns pertain to reviewers, and it's possible that some reviewers are exclusively identified by their name, while others are solely recognized by their Twitter handle. To consolidate this information, we aim to merge these two columns into a unified identifier for each taster. To achieve this, we would prioritize the 'taster_name' as the primary identifier to distinguish tasters from one another. Consequently, we would fill any missing values in the 'taster_name' column with the corresponding Twitter handle from the same row. Additionally, we should consider the possibility of multiple reviewers sharing the same Twitter handle or a single reviewer using multiple Twitter handles when reviewing wines. These complexities will require careful handling during the merging process.

Here we determine that **rows containing a name but no twitter handle: 4969** and **rows containing a twitter handle but no taster name: 0.**

Given the likelihood of tasters reviewing wines with multiple Twitter handles and potentially altering their reviewing style with different handles (e.g., due to Twitter account changes), it becomes essential to identify any instances of tasters using multiple Twitter handles. To accomplish this, we should conduct a thorough scan of the data to detect any tasters who have employed multiple Twitter handles for reviewing wines. This examination will aid us in understanding the variations in reviewing patterns associated with different Twitter accounts and ensure the appropriate handling of these cases during data analysis and consolidation processes.



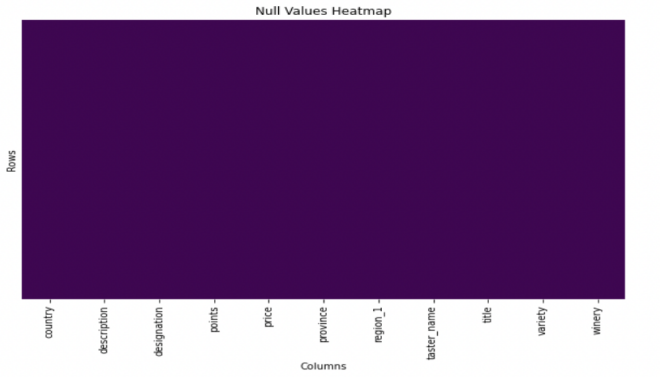
As depicted earlier, every taster employs either a single Twitter Handle or none at all. Consequently, we can readily eliminate the 'taster_twitter_handle' column since it doesn't offer any additional information for our models. Additionally, we have the option to substitute any missing values in the 'taster_name' column with "Unknown."

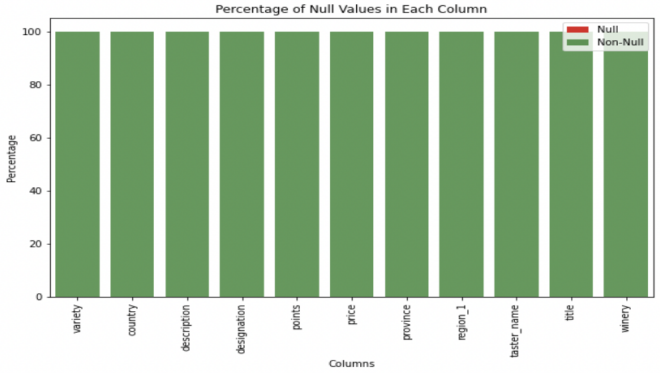


The final column containing Null values is 'price'. We can easily handle this issue by replacing the missing values with the mean of the column. This approach ensures that the column means remain unchanged, which could be crucial for subsequent analysis steps. However, it's important to note that using this method, just like replacing missing values with the column's median, lacks precision and doesn't maintain the relationships between the respective feature and other features.

To address the missing values in the 'price' column, we'll use a simple example of predicting them using KNN (K-Nearest Neighbors) imputation. In this method, we replace the missing 'price' values with the median of the 'price' values from the K nearest neighbors. While this technique may not accurately reveal the real prices in all cases, it retains some residuals and preserves the relationships between the features used in the prediction process. For simplicity, we'll only use the features 'points', 'country', and 'taster_name' to predict the missing values.

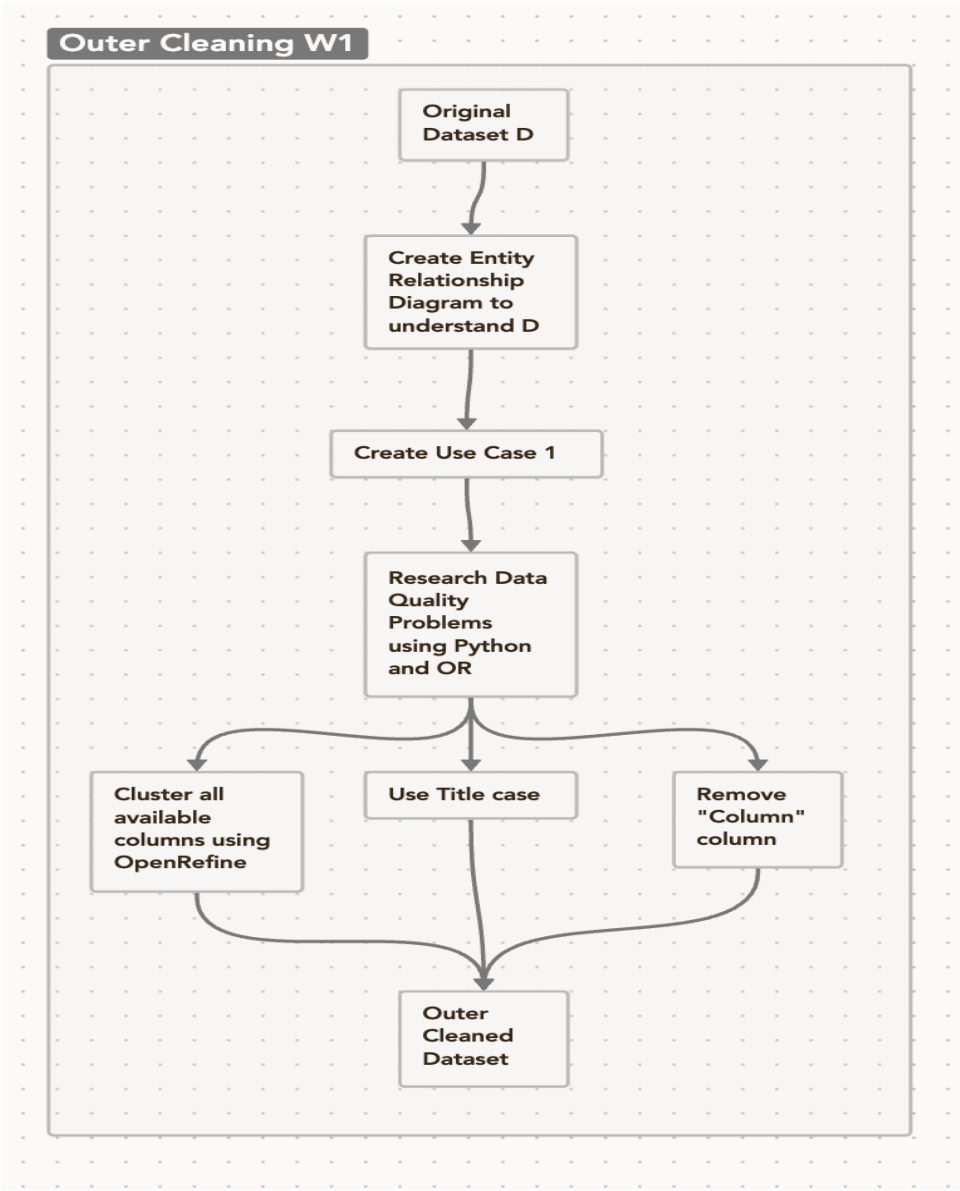
Since our goal is to predict prices, which are numerical values, we'll use a regression model, specifically the KNeighboursRegressor. One significant advantage of employing a KNN Regressor instead of other regression models is that, by definition, it doesn't predict values that fall outside the range of the already collected data observations (i.e., the training data). Please note that the sklearn KNeighboursRegressor predicts the mean of the y-values of the K nearest neighbors. While in some cases, predicting the median might be more suitable, it's unfortunate that predicting the mean is hardcoded in sklearns KNeighboursRegressor. For the purpose of this tutorial, we will rely on sklearn for simplicity. To start, we need to convert the categorical columns 'country' and 'taster_name' into indicator variables. Then, we'll store the training data and the data on which we want to predict the prices separately.

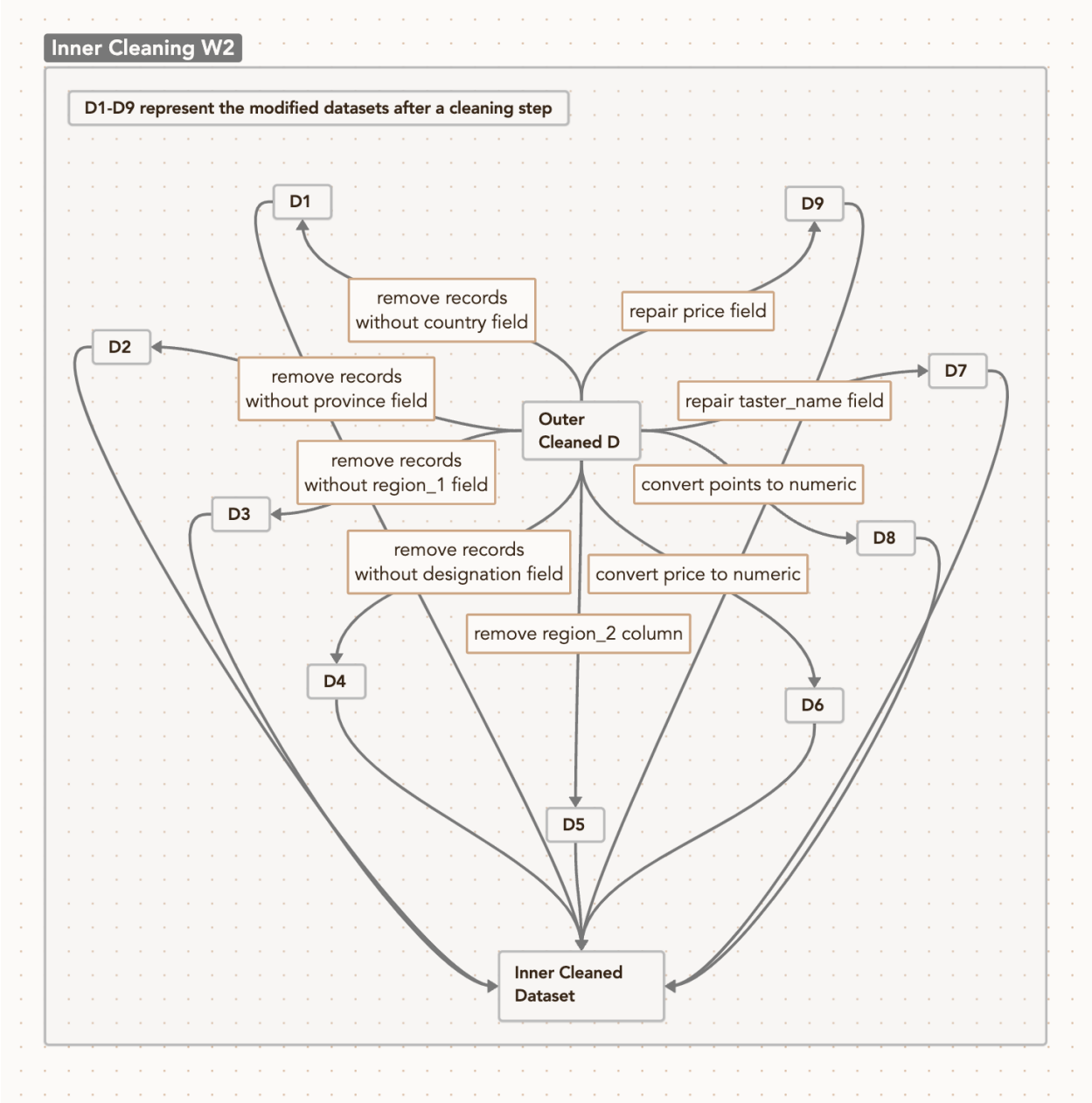




	country	description	designation	points	price	province	region_1	taster_name	title	variety	winery
0	France	The aromas are of new wood and toast as much a...	Unknown	89	30.0	Bordeaux	Graves	Roger Voss	Château Roquetaillade la Grange 2009 Graves	Bordeaux-style Red Blend	Château Roquetaillade la Grange
1	US	Waxy plum is given ample richness from a taste...	Unknown	89	54.0	California	Napa Valley	Virginie Boone	Duckhorn 2014 Merlot (Napa Valley)	Merlot	Duckhorn
2	US	There's an exotic saltiness and briny ocean el...	Jewell Vineyard Dutton Ranch	91	51.0	California	Russian River Valley	Virginie Boone	The Calling 2014 Jewell Vineyard Dutton Ranch ...	Chardonnay	The Calling
3	US	Comprised of Cabernet Sauvignon, Cabernet Fran...	E&E Shaw Vineyard	88	37.0	Washington	Red Mountain	Paul Gregutt	Tapteil Vineyard 2010 E&E Shaw Vineyard Red (R...	Bordeaux-style Red Blend	Tapteil Vineyard
4	US	As always, Seghesio's Cortina is a rich, compl...	Cortina	93	38.0	California	Dry Creek Valley	Unknown	Seghesio 2010 Cortina Zinfandel (Dry Creek Val...	Zinfandel	Seghesio

Visual Workflow:





Lessons Learned:

1. The data being fit for purpose is the primary goal for data cleaning. We found the mantra *"fit for purpose"* helpful when deciding if a repair or record removal was required. An example of this was whether we should have repaired region_1 with designation information. The outcome was that we removed the data because our group could not guarantee the derived information from designation could be *"fit for purpose"*.

2. Validation and verification is important when deriving information from existing columns in the data. There were several cases where we were deciding between data repairs and removals. Since we could not validate the information we derived critical to the U_1, we decided to remove the data. Having tools and resources to perform validation and verification can help increase the size of a dataset depending on the use case.
3. In general, we learned how to leverage tools like OpenRefine for data profiling which helped in *Phase I*. We attempted to leverage YesWorkflow, however we were unable to produce a diagram using the tool. In future work, reviewing how to use YesWorkflow at the beginning of a project would better position a group to use tool like YesWorkflow appropriately.

Contributions:

In general we split the workload where an individual could complete a body of work or answer a single question. We felt that each team member made equal contributions whether it was through meeting collaboration, defect discovery, data cleaning, or documentation.

Contributor	Tasks
Ahmed Elfarra	● Data Profiling ● Analysis & Repair ● Defect Discovery ● Documentation ● Meeting Collaboration
Nishanth Alladi	● Data Profiling ● Analysis Reviewer ● Documentation ● Defect Discovery ● Meeting Collaboration
Adam Michalsky	● Data Profiling ● Analysis Reviewer ● Meeting Facilitator/Collaboration ● Analysis Reviewer ● Documentation