Technical Project Report

Student Name : Sathvik Bhupal

SUID : L00171188

Title: Training Alcohol Data with Machine Learning Classifiers.

Author: Sathvik Bhupal

Supervisor: Professor Dr. Shagufta Henna

Degree: MSc Data Science

GITHUB Link: https://github.com/sathvikbhupal1/TechnicalProject

Executive Summary

The project or study is based on the Alcohol dataset [1]. The dataset has a wide range of information regarding various Alcoholic and Non Alcoholic drinks with their manufacturing country, ABV(Alcohol by volume), Tasting Notes, Category it belongs to, Rating, Number of ratings received, Tasting notes, Food pairing, Suggested Temperature to serve, Price and Description. The dataset contains 33885 records after initial cleaning. Machine Learning classifiers have been built and implemented on the dataset after splitting the data into train and test proportions respectively. The train test split is done to train the data perfectly and then checking it on the test data to calculate the accuracy of the model. The dataset can be used to study the correlation between the alcohol percentage and its price. The machine learning model is made to train and understand the relation mainly between ABV and its pricing. The training further helps the model to predict prices based in the ABV of the alcohol.

Introduction

The Alcohol dataset [1] has three parts of CSV files such as beer data, spirits data and wine data which is used to perform an analysis and train to show its correlation and impact on the alcohol prices with the help of various information or features provided in the dataset. Machine learning classifiers are used to train the data and to predict output. Random Forest Classifier, Support vector machines, linear regression classifiers are used to perform the training for the data in this project.

The data undergoes various cleaning operations such as removal of null and NA values, removal of outliers, feature filtering and then classifiers are implemented further.

The following **Hypothesis** is checked/evaluated using the Alcohol dataset-

H0 = There is a good correlation between ABV and Price so ABV helps in contributing to the prediction of price.

H1 = There is no correlation between ABV and Price, henceforth ABV cannot contribute in predicting the price of the Alcohol.

Description of the data

The dataset is a root of this technical project and it is always the right data which builds up the project with an accurate predictions. The alcohol dataset [1] used in the project is a precise lake of information about the Alcohol brands, Alcohol manufacturers, Manufacturer's Country, Rating, Price and much more. The alcohol dataset [1] has three different csv files containing beer, wine and spirits respectively with all the fields of information mentioned above. The dataset has got many missing fields in rows and columns, the data is divided in three csv files, the data contains few special characters which has to be taken care of, the dataset has completely unfilled columns, are few of the challenges to transform the dataset. The dataset comprises of the following columns which describes the corresponding row values to it.

- Name : Name of the alcohol.
- Country : Country where the alcohol is produced/manufactured.
- Brand : Brand name of the Alcoholic Drink.
- Categories : Type of alcohol it belongs to.
- Tasting Notes : Taste of the alcohol.
- Base Ingredient : Main ingredient used while producing the alcohol.
- Years Aged : Total number of years that alcohol is stored since its manufacture.
- Rating : Rating of the alcohol.
- Rate Count : Number of people rated for that particular alcohol.
- Price : Selling price of the alcohol.
- Volume : Quantity of the alcohol to which the selling price is mentioned for.
- Description : Description of the alcohol.

Dataset URL - https://www.kaggle.com/datasets/limtis/wikiliqdataset

Methodology

1. **Importing and cleaning the Dataset**

   The three CSV files are imported to the environment with files.upload() function where we are given the option to upload files from the computer and are converted as df(data frame) 1. Then df.columns and df.dropna() function are used to decribe the columns and also to drop unnecessary columns from the data frame 2. Dropna() 2 function with axis mentioned in parameters is being used to drop the NA values of the data frame to avoid improper training of data on the classifier.

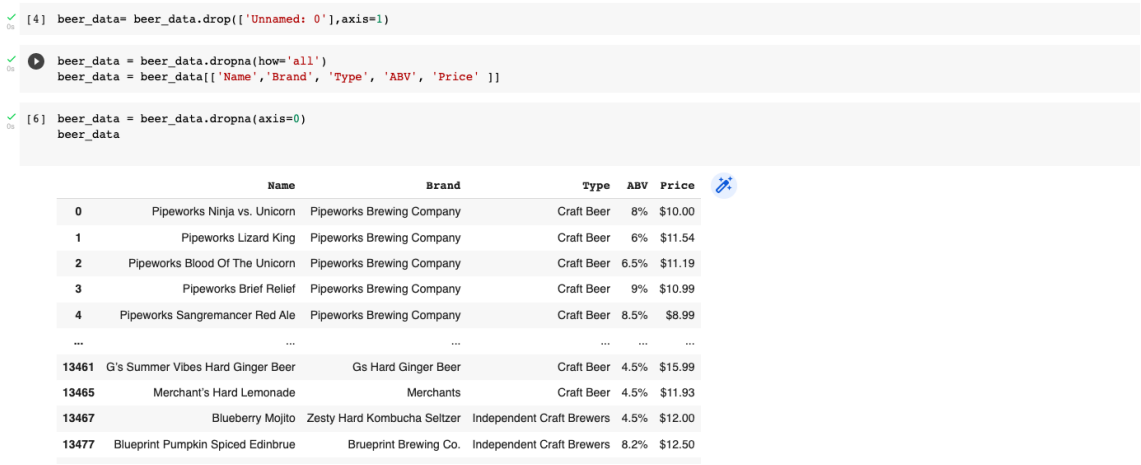
```
import pandas as pd
import io
from google.colab import files


uploaded = files.upload()
beer_data = pd.read_csv(io.BytesIO(uploaded['beer_data.csv']), header = 0)
```

```
Choose files  beer_data.csv
• beer_data.csv(text/csv) - 5468090 bytes, last modified: 15/05/2022 - 100% done
Saving beer_data.csv to beer_data (2).csv
```

```
[3] type(beer_data)
    beer_data.columns
```

```
Index(['Unnamed: 0', 'Name', 'Country', 'Brand', 'Categories', 'Type',
       'Tasting Notes', 'ABV', 'IBU', 'Calories Per Serving (12 OZ/0.35L)',
       'Carbs Per Serving (12 OZ/0.35L)', 'Food Pairing',
       'Suggested Serving Temperature', 'Rating', 'Rate Count', 'Price',
       'Volume', 'Description'],
      dtype='object')
```

Figure 1: Loading the Data.



```
[4] beer_data= beer_data.drop(['Unnamed: 0'],axis=1)

    beer_data = beer_data.dropna(how='all')
    beer_data = beer_data[['Name','Brand', 'Type', 'ABV', 'Price' ]]

[6] beer_data = beer_data.dropna(axis=0)
    beer_data
```

	Name	Brand	Type	ABV	Price
0	Pipeworks Ninja vs. Unicorn	Pipeworks Brewing Company	Craft Beer	8%	$10.00
1	Pipeworks Lizard King	Pipeworks Brewing Company	Craft Beer	6%	$11.54
2	Pipeworks Blood Of The Unicorn	Pipeworks Brewing Company	Craft Beer	6.5%	$11.19
3	Pipeworks Brief Relief	Pipeworks Brewing Company	Craft Beer	9%	$10.99
4	Pipeworks Sangremancer Red Ale	Pipeworks Brewing Company	Craft Beer	8.5%	$8.99
...	...	...	...	...	...
13461	G's Summer Vibes Hard Ginger Beer	Gs Hard Ginger Beer	Craft Beer	4.5%	$15.99
13465	Merchant's Hard Lemonade	Merchants	Craft Beer	4.5%	$11.93
13467	Blueberry Mojito	Zesty Hard Kombucha Seltzer	Independent Craft Brewers	4.5%	$12.00
13477	Blueprint Pumpkin Spiced Edinbrue	Brueprint Brewing Co.	Independent Craft Brewers	8.2%	$12.50

Figure 2: Cleaning the Data.

Then the other two CSV files are loaded in the same manner and are converted into data frames with cleaning done. The dataset looks problematic when looked at its features because the Type feature in first df is matching with the catergory feature in second and third df. So to avoid that faulty nature of the data df.columns function is used to redefine the df with the proper column names and here we are chaning category columns in second and third dataset as Type as shown in the figure below 3.

```
[12] spirits_data.columns = ['Name','Brand', 'Type', 'ABV', 'Price' ]
     spirits_data
```

	Name	Brand	Type	ABV	Price
0	Dekuyper Triple Sec Liqueur	DeKuyper Liqueur	Citrus, Triple Sec Liqueur, Liqueur	24%	$10.99
1	DeKuyper Peachtree Schnapps Liqueur	DeKuyper Liqueur	Liqueur	20%	$11.69
2	DeKuyper Sour Apple Pucker Schnapps Liqueur	DeKuyper Liqueur	Liqueur	15%	$11.99
3	DeKuyper Blue Curacao Liqueur	DeKuyper Liqueur	Liqueur	24%	$11.99
4	DeKuyper Buttershots Schnapps Liqueur	DeKuyper Liqueur	Liqueur	15%	$12.99
...	...	...	...	...	...
12862	Killepitsch	Killepitsch	Liqueur	35%	$23.99
12863	Or G French Liqueur	Or G	Liqueur	34%	$8.95
12865	Roiano Liquore	Roiano	Liqueur, Nuts, Amaretto Liqueur	40%	$16.99
12866	Very Special Chocolates Classic Assortment Liq...	Very Special Chocolates	Chocolate, Sweet Liqueur, Liqueur	5%	$20.62
12868	Don Felix Anejo Tequila	Don Felix	Anejo Tequila, Tequila	40%	$56.34

10471 rows × 5 columns

Figure 3: Renaming the Columns.

The concat function as pd.concat is used to merge all the three data frames as a single big data frame to perform analysis and training. The collective or combined data frame consists of 33885 records after cleaning with the five columns or features chosen by us 4.

```
[19] combined_df = pd.concat([beer_data, spirits_data, wine_data])
     combined_df
```

	Name	Brand	Type	ABV	Price
0	Pipeworks Ninja vs. Unicorn	Pipeworks Brewing Company	Craft Beer	8%	$10.00
1	Pipeworks Lizard King	Pipeworks Brewing Company	Craft Beer	6%	$11.54
2	Pipeworks Blood Of The Unicorn	Pipeworks Brewing Company	Craft Beer	6.5%	$11.19
3	Pipeworks Brief Relief	Pipeworks Brewing Company	Craft Beer	9%	$10.99
4	Pipeworks Sangremancer Red Ale	Pipeworks Brewing Company	Craft Beer	8.5%	$8.99
...	...	...	...	...	...
26091	Anne Brigitte, Pays d'Oc 2018, Rosé Wine	Anne Brigitte	Pink Wine, Ros Wine	13%	$0.00
26092	Tribute To Grace Rose of Grenache 2016	A Tribute To Grace	Pink Wine, Ros Wine	13.1%	$0.00
26095	Angel Affair Rosé	Angel Affair	Pink Wine, Ros Wine	12.5%	$11.99
26096	Accademia dei Racemi Burlesque Rose	Accademia dei Racemi	Pink Wine, Ros Wine	13%	$12.99
26098	Centorri Moscato Di Pavia	Centorri	Moscato, White Wine	6.5%	$12.99

33885 rows × 5 columns

Figure 4: Combined DataFrame.

2. **Visualisation**

The Data has been visualised with the help of matplotlib, plotly and seaborn with different types of plots/graphs. Plotly has been used to plot a scatter plot between ABV and Price feature to understand its correlation 6. Matplotlib is used to show the bar graph with the help of Type and ABV plotted against Price on the Y-axis 7. Seaborn is used to generate heatmap 8 to understand the correlation in the data to avoid biased nature in the data.

The Data has been cleaned further with proper classification. Funcrions like .str and .replace have been used to replace the different category names to their proper names to increase the efficiency of the model training and output 9.

Special Characters in the Data has been removed with the help of the same .str.replace() function and here in the data the percentage symbol from ABV

```
[20] %matplotlib inline
     import matplotlib.pyplot as plt
     import seaborn as sn

     import plotly.express as px

     df = combined_df
     fig = px.scatter(df, x="ABV", y="Price",
                      width=800, height=400)

     fig.update_layout(
         margin=dict(l=20, r=20, t=20, b=20),
         paper_bgcolor="LightSteelBlue",
     )
     fig.update_xaxes(categoryorder='category ascending')
     fig.update_yaxes(categoryorder='category ascending')
     fig.show()
```

Figure 5: Libraries Imported and Coding for Scatter Plot.

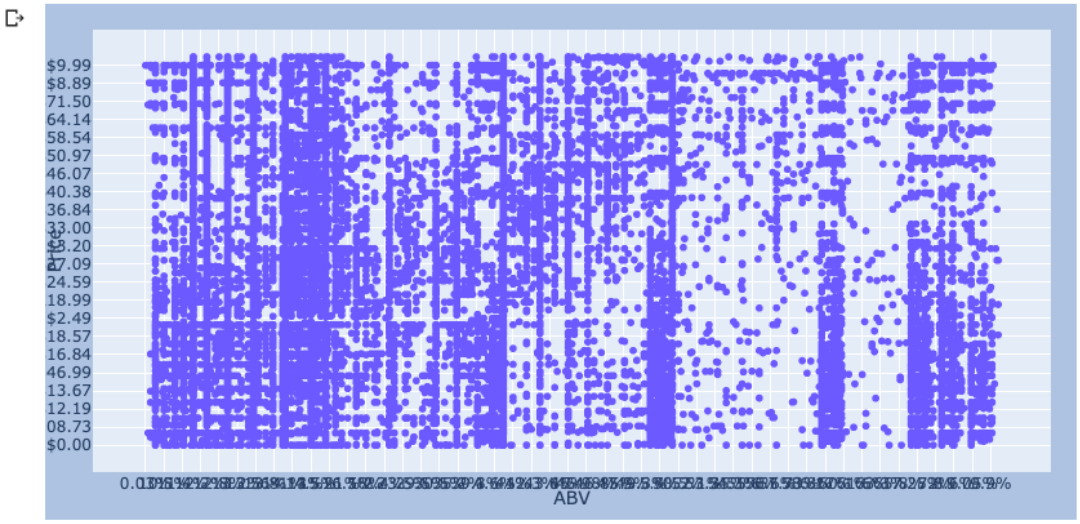


Figure 6: Scatter Plot Graph by Plotly.

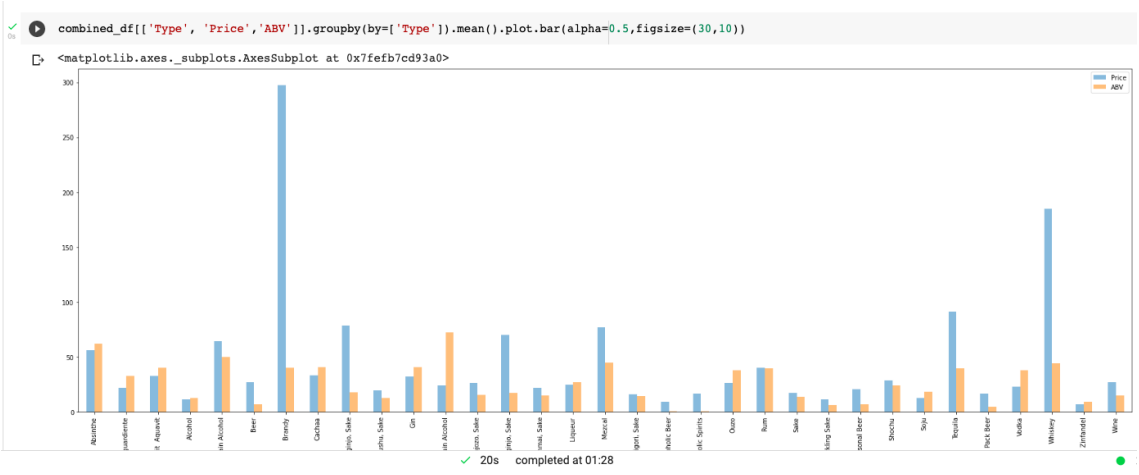
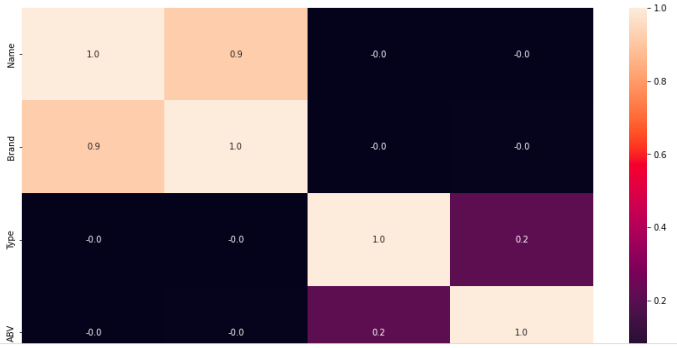


Figure 7: Bar Graph by Matplotlib.

and the dollar symbol from Price has to be removed to plot and train the data perfectly 10.

The Data is converted into numerical values using one hot encoding method with the help of LabelEncoder() function 11. The data is first converted in the form of array and then made to pass through LabelEncoder Function 11.

```
corr_matrix = combined_df.corr()

import seaborn as sn
import matplotlib.pyplot as plt

plt.figure(figsize=(15,8))
sn.heatmap(corr_matrix, annot=True, fmt=".1f")
plt.show()
```



✓ 20s completed at 01:28

Figure 8: Heat Map by Seaborn.

```
#combined_df['Type'] = test_df.Type.str.replace(r'(^.*Beer.*$)', 'Beer')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Craft.*$)', 'Beer')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Vodka.*$)', 'Vodka')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Rum.*$)', 'Rum')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Brandy.*$)', 'Brandy')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Tequila.*$)', 'Tequila')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Liqueur.*$)', 'Liqueur')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Ready.*$)', 'Alcohol')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Whiskey.*$)', 'Whiskey')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Gin.*$)', 'Gin')
combined_df['Type'] = combined_df.Type.str.replace(r'(^.*Wine.*$)', 'Wine')
combined_df
```

```
The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:4: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:5: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:6: FutureWarning:

The default value of regex will change from True to False in a future version.

<ipython-input-23-222fe82c912a>:7: FutureWarning:

The default value of regex will change from True to False in a future version.
```

✓ 20s completed at 01:28

Figure 9: Categorising the Data with .str.replace function.

The converted array which consists of numerical values is again brought back to the data frame form with the help of pd.dataframe() function.

The Dataset is then converted from object data type to float using .astype() 12 function as machine learning algorithms cannot perform analysis on object data types. The 0 values from ABV and Price are then replaced with NA using .replace function() as 0 values can hamper the algorithm training.

Dropna function is used to remove the new NA values. The outliers from the data are removed 13 for the better performance of algorithm and output as the outliers are responsible for the high variance in the data.

3. **Applying Machine Learning Algorithm/Classifier**

   - The following Machine Learning classifiers/algorithms are used to achieve desired outcomes -

```
combined_df['Price'] = combined_df['Price'].str.replace('$', '')
combined_df['Price'] = combined_df['Price'].astype(float)
combined_df['ABV'] = combined_df['ABV'].str.replace('%', '')
combined_df['ABV'] = combined_df['ABV'].astype(float)
combined_df
```

```
<ipython-input-24-66dfd5ebleaf>:1: FutureWarning:

The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as lit
```

	Name	Brand	Type	ABV	Price
0	Pipeworks Ninja vs. Unicorn	Pipeworks Brewing Company	Beer	8.0	10.00
1	Pipeworks Lizard King	Pipeworks Brewing Company	Beer	6.0	11.54
2	Pipeworks Blood Of The Unicorn	Pipeworks Brewing Company	Beer	6.5	11.19
3	Pipeworks Brief Relief	Pipeworks Brewing Company	Beer	9.0	10.99
4	Pipeworks Sangremancer Red Ale	Pipeworks Brewing Company	Beer	8.5	8.99
...	...	...	...	...	...
26091	Anne Brigitte, Pays d'Oc 2018, Rosé Wine	Anne Brigitte	Wine	13.0	0.00
26092	Tribute To Grace Rose of Grenache 2016	A Tribute To Grace	Wine	13.1	0.00
26095	Angel Affair Rosé	Angel Affair	Wine	12.5	11.99
26096	Accademia dei Racemi Burlesque Rose	Accademia dei Racemi	Wine	13.0	12.99

✓ 20s completed at 01:28

Figure 10: Removing the special characters with .str.replace() function.

```
combined_df = combined_df.iloc[:,:].values
combined_df
```

```
array([['Pipeworks Ninja vs. Unicorn', 'Pipeworks Brewing Company',
        'Beer', 8.0, 10.0],
       ['Pipeworks Lizard King', 'Pipeworks Brewing Company', 'Beer',
        6.0, 11.54],
       ['Pipeworks Blood Of The Unicorn', 'Pipeworks Brewing Company',
        'Beer', 6.5, 11.19],
       ...,
       ['Angel Affair Rosé', 'Angel Affair', 'Wine', 12.5, 11.99],
       ['Accademia dei Racemi Burlesque Rose', 'Accademia dei Racemi',
        'Wine', 13.0, 12.99],
       ['Centorri Moscato Di Pavia', 'Centorri', 'Wine', 6.5, 12.99]],
      dtype=object)
```

```
[ ] from sklearn.preprocessing import LabelEncoder
```

```
[ ] transform = LabelEncoder()
    combined_df[:,0]= transform.fit_transform(combined_df[:,0])
    combined_df[:,1]= transform.fit_transform(combined_df[:,1])
    combined_df[:,2]= transform.fit_transform(combined_df[:,2])

    combined_df
```

```
array([[24057, 7898, 5, 8.0, 10.0],
       [24047, 7898, 5, 6.0, 11.54],
       [23996, 7898, 5, 6.5, 11.19],
```

Figure 11: Converting row values from string to int with LabelEncoder() .

```
combined_df['ABV'] = combined_df['ABV'].astype(float)
combined_df['Name'] = combined_df['Name'].astype(float)
combined_df['Brand'] = combined_df['Brand'].astype(float)
combined_df['Type'] = combined_df['Type'].astype(float)
combined_df['Price'] = combined_df['Price'].astype(float)
```

```
[ ] price_col = combined_df['Price']
    price_col.replace(to_replace = 0, value =pd.NA, inplace=True)
```

```
[ ] abv_col =combined_df['ABV']
    abv_col.replace(to_replace = 0, value = pd.NA, inplace=True)
```

```
[ ] #combined_df = combined_df.dropna(how='all')
    combined_df
```

	Name	Brand	Type	ABV	Price
0	24057.0	7898.0	5.0	8.0	10.0
1	24047.0	7898.0	5.0	6.0	11.54
2	23996.0	7898.0	5.0	6.5	11.19
3	24002.0	7898.0	5.0	9.0	10.99
4	24070.0	7898.0	5.0	8.5	8.99
...	...	...	...	...	...
33880	1547.0	428.0	32.0	13.0	<NA>

Figure 12: Converting data type object to float.

```
[ ]
    q_low = combined_df["Price"].quantile(0.01)
    q_hi  = combined_df["Price"].quantile(0.99)

    combined_df = combined_df[(combined_df["Price"] < q_hi) & (combined_df["Price"] > q_low)]
```

```
[ ]
    q_low = combined_df["ABV"].quantile(0.01)
    q_hi  = combined_df["ABV"].quantile(0.99)

    combined_df = combined_df[(combined_df["ABV"] < q_hi) & (combined_df["ABV"] > q_low)]
```

Figure 13: Removal of Outliers from the class label and ABV.

The Data is now ready for the Machine Learning classifier and before applying the classifier on it, PCA()(Principle Component Analysis) function 14 is used for dimensonality reduction if there are any features which are heavily correlated. Fit tranform function is used to tranform the Y labels with the help of a LabelEncoder 14 to improve the accuracy on linear regression and random forest models as few classifiers like random forest work well with classification problems.

```
[ ] X = combined_df.drop(columns=['Price'])
    Y = combined_df['Price']
```

```
[ ] from sklearn.decomposition import PCA

    pca = PCA(n_components=3)
    X2 = pca.fit_transform(X)
```

```
    from sklearn import preprocessing
    from sklearn import utils

    #convert y values to categorical values
    lab = preprocessing.LabelEncoder()
    y_transformed = lab.fit_transform(Y)
```

Figure 14: Performing PCA and fit.tranform() on Dataset.

(a) **Linear Regression Model**
The linear regression model is a supervised machine learning model and one of the simplest from machine learning techniques. This regression model simulates the relationship between the independent and dependent variables [2]. This quality of model makes it a good fit for our technical project as we can simulate the relationship between two or more independent variable such as ABV, category etc to the target variable(price). The linear regression model performs the task of predicting the target value based on the simulated relation with its independent value. Thus, it justifies the reason to be a good fit to the technical project.
The Data is split in the form of X and Y where Y is the class label and X is the remaining part of data. Train and test split library is imported to train data in train and test format and metrics. accuracy is imported from accuracy to check the accuracy of the data 15.

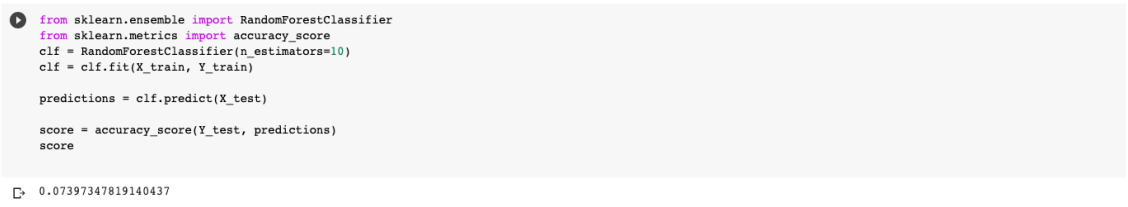
(b) **Random Forest Model** Random forest model is a frequently used machine learning algorithm. It is easy to use and flexible to adapt, which attracted many projects. It is made up of multiple collection of decision trees which contributes to the accuracy of predictions. The

```
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import accuracy_score, mean_absolute_error, mean_squared_error
import matplotlib
import matplotlib.pyplot as plt
import numpy as np


X_train, X_test, Y_train, Y_test = train_test_split(X,y_transformed, train_size = 0.8)


clf = LinearRegression()
clf.fit(X_train, Y_train)

predictions = clf.predict(X_test)

print("Score:", clf.score(X_test, Y_test))
```

```
Score: 0.2698086964582588
```

Figure 15: Linear Regression classifier with Accuracy.

random forest classifier works well for both regression and classification problems [4].Thus, it makes this model an option to be kept in mind while applying machine learning algorithm to the project.
The Random Forest classifier is built and the split data is then passed into the built classifier to train and test the data where we set the train percentage as 80 percent by passing it in the parameter. The nestimators parameter from the Random Forest classifier has been set as 10 to reduce the computing costs and the classifier is run to train and test the data. Accuracy is predicted after the training of data 16.



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
clf = RandomForestClassifier(n_estimators=10)
clf = clf.fit(X_train, Y_train)

predictions = clf.predict(X_test)

score = accuracy_score(Y_test, predictions)
score
```

```
0.07397347819140437
```

Figure 16: Random Forest classifier with Accuracy.

Conclusion

The Alcohol Dataset has many insights which were captured by us throughout this project by Cleaning, Visualising and applying Machine Learning Classifiers to it through Python. The insights give us a brief meaning of the correlation of its features and prediction of the prices through that correlation. The vast amount of data also helps in throwing light on the Alcohol manufactures of different countries or their contribution of Alcohol to the world. The Alcohol Dataset with analysis and classifiers applied and performed on them says it clearly that the predictions are not accurate and the correlation is not strong to train the model and predict through classifiers. Hence, **we reject H0 and accept H1**

Scope of Work

I would use all the features for the cleaning, visualisation and prediction which is very different with the one we have done, as we have chosen a handpicked features for the prediction. I would totally change my approach from the Data preperation to cleaning. I will have a statistical approach to the data rather than expertise approach and I would use different Machine learning classifiers other than the ones used in this study. I would create a def function to find out unique values from the dataset and keep ones with low classification for feature selection. I would create a def function to filter out columns which has null values more than 80 percent. I would use gradient boost classifier and also Lasso or Ridge classifier to normalise the data.

References

[1] Limtis(2022). *Wikiliq - Alcohol Dataset Kaggle Website.* Accessed on Jan, 2023.
URL - https://www.kaggle.com/datasets/limtis/wikiliqdataset

[2] Project pro. *Predective modelling techniques.* Accessed on Jan, 2023.
URL - https://www.projectpro.io/article/predictive-modelling-techniques/598

[3] Analytics Vidhya. *Gradient boosting Algorithm: A complete guide for beginners.* Accessed on Jan, 2023.
URL - https://www.analyticsvidhya.com/blog/2021/09/gradient-boosting-algorithm-a-complete-guide-for-beginners

[4] IBM. *What is random forest?.* Accessed on Jan, 2023.
URL - https://www.ibm.com/topics/random-forest