औद्योगिक प्रशिक्षण के लिए राष्ट्रीय संस्थान

National Institute for Industrial Training

One Premier Organization with Non Profit Status | Registered Under Govt. of WB
Empanelled Under Planning Commission Govt. of India
Inspired By: National Task Force on IT & SD Government of India

National Institute for Industrial Training- One Premier Organization with Non Profit Status Registered Under Govt.of West Bengal,Empanelled Under Planning Commission Govt.of India , Empanelled Under Central Social Welfare Board Govt. of India , Registered with National Career Services , Registered with National Employment Services.

SUBJECT: PYTHON WITH DATA SCIENCE

Submitted by: PREETI SHARMA

Submitted to: SOUMATANU MAJUMDAR

Name: Preeti Sharma

College: Kurukshetra University(U.I.E.T.)

Course: Biotech Engineering

Year: 2nd

Year of passing: 2023

E-mail ID: preetishr27@gmail.com

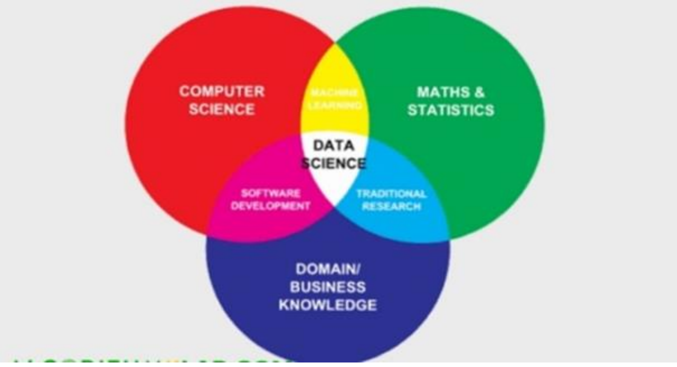
Contents

Acknowledgment

I am privileged and grateful to acknowledge my knowledge to all those who have guided me to put these ideas, well above the level of simplicity and into something concrete. I express my warm gratitude to National Institute for Industrial Training for their constant guidance and supervision as well as for providing necessary information regarding the project. I would like to express my sole thanks of gratitude to (Soumotanu Majumdar) for his support, co-operation and encouragement which helped me in the completion of this project.

My journey of completing the project wouldn’t would remain incomplete if I don’t extend my gratitude to my parents for their love , affection , support and constant guidance.

Introduction



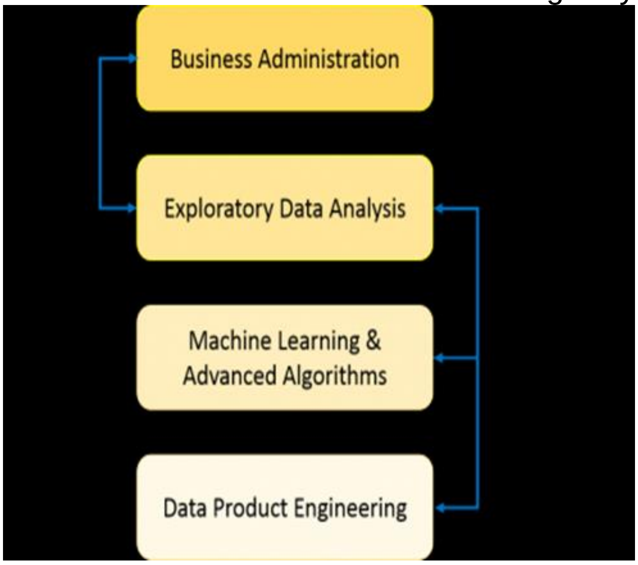
“A combination of information technology , modelling , and business management”- Dr. Thomas Miller of Northwestern University.

In today’s world Data science and Machine Learning are one of the most prominent topics in the arena of technology. Data science and machine learning are two distinguishable topics although they look analogous to each other. Combining , computing , comparing and concluding the insights of datas are known as data science while on the flip side machine learning is the processes by which a result can be obtained from the given data.

What is Data Science ?

Have anyone wondered why our life in this era of iron age has become so comfortable and scientific? It’s only because of computers , smartphones , tablets , laptops and many more electronic devices which have completely digitalized our life and consequently resulted in huge amount data. This data needs to be processed , organized and coordinated. The phenomenon of mastering these process or the study of these data sets are basically categorized under data science.

Data Science is a blend of various tools, algorithms, and machine learning principles with the goal to discover hidden patterns from the raw data. But how is this different from what statisticians have been doing for years?


Business Administration
Exploratory Data Analysis
Machine Learning & Advanced Algorithms
Data Product Engineering

The answer lies in the difference

between explaining and predicting. The principal purpose of Data Science is to find patterns within data. It uses various statistical techniques to analyze and draw insights from the data. From data extraction, wrangling and pre-processing, a Data Scientist must scrutinize the data thoroughly. Then, he has the responsibility of making predictions from the data. The goal of a Data Scientist is to derive conclusions from the data. Through these conclusions, he is able to assist companies in making smarter business decisions. We will divide this blog into various sections to understand the role of a Data Scientist in more detail. Industries need data to help them make careful decisions. Data Science churns raw data into meaningful insights. Therefore, industries need data science. A Data Scientist is a wizard who knows how to create magic using data. A skilled Data Scientist will know how to dig out meaningful information with whatever data he comes across. He helps the company in the right direction. The company requires strong data-driven decisions at which he's an expert. The Data Scientist is an expert in various underlying fields of Statistics and Computer Science. He uses his analytical aptitude to solve business problems. Data Scientist is well versed with problem-solving and is assigned to find patterns in data. His goal is to recognize redundant samples and draw insights from it. Data science requires a variety of tools to extract information from the data. A Data Scientist is responsible for collecting, storing and maintaining the structured and unstructured form of data.

While the role of Data Science focuses on the analysis and management of data, it is dependent on the area that the company is specialized in. This requires the Data Scientist to have domain knowledge of that particular industry.

Advantages

□**A great library ecosystem:** A great choice of libraries is one of the main reasons Python is the most popular programming language used for AI. A library is a module or a group of modules published by different sources like Py Pi which include a pre-written piece of code that allows users to reach some functionality or perform different actions. Python libraries provide base level items so developers don’t have to code them from the very beginning every time.

□**A low entry barrier:** Working in the ML and AI industry means dealing with a bunch of data that you need to process in the most convenient and effective way. The low entry barrier allows more data scientists to quickly pick up Python and start using it for AI development without wasting too much effort on learning the language.

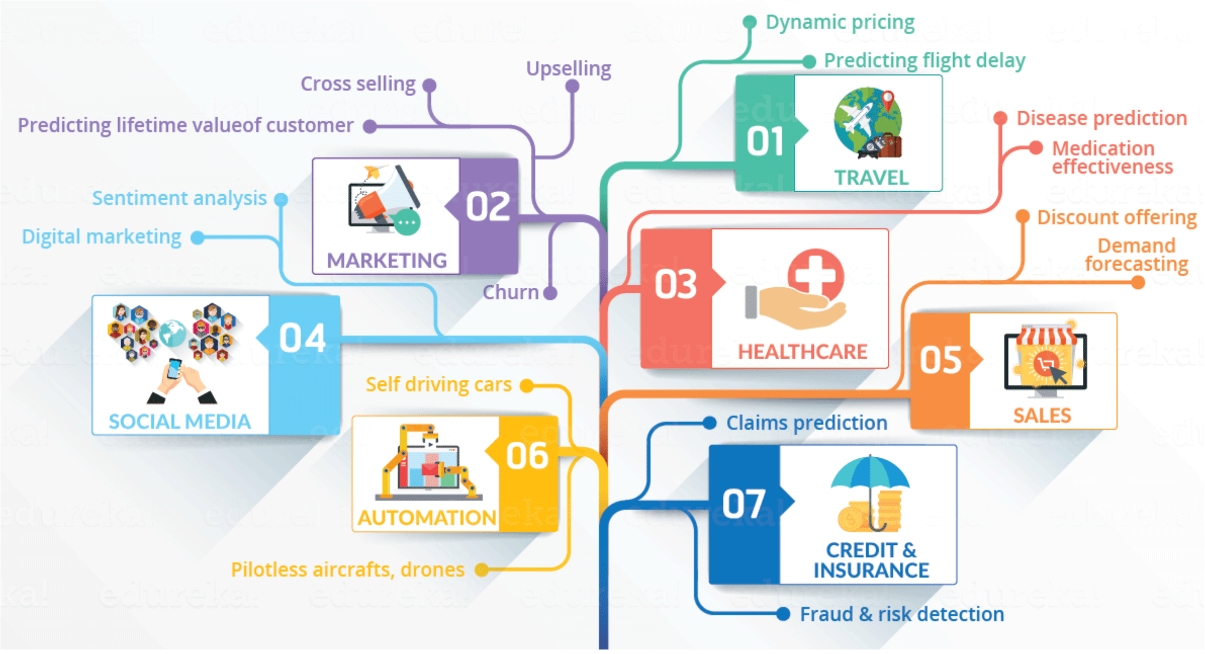
□**Flexibility:** Python for machine learning is a great choice, as this language is very flexible. It offers an option to choose either to use OOPs or scripting. There’s also no need to recompile the source code, developers can implement any changes and quickly see the results. Programmers can combine Python and other languages to reach their goals. The flexibility factor decreases the possibility of errors, as programmers have a chance to take the situation under control and work in a comfortable environment.

□**Platform independence:** Python is not only comfortable to use and easy to learn but also very versatile. What we mean is that Python for machine learning development can run on any platform including Windows, MacOS, Linux, UNIX, and twenty-one others. To transfer the process from one platform to another, developers need to implement several small-scale changes and modify some lines of code to create an executable form of code for the chosen platform. Developers can use packages like Py Installer to prepare their code for running on different platforms. Again, this saves time and money for tests on various platforms and makes the overall process more simple and convenient.

□**Readability**: Python is very easy to read so every Python developer can understand the code of their peers and change, copy or share it. There’s no confusion, errors or conflicting paradigms, and this leads to a more efficient exchange of algorithms, ideas, and tools between AI and ML professionals.

- **Good visualization options:** We’ve already mentioned that Python offers a variety of libraries, and some of them are great visualization tools. However, for AI developers, it’s important to highlight that inartificial intelligence, deep learning, and machine learning, it’s vital to be able to represent data in a human-readable format. Libraries like Matplotlib allow data scientists to build charts, histograms, and plots for better data comprehension, effective presentation, and visualization. Different application programming interfaces also simplify the visualization process and make it easier to create clear reports.
- **Community support:** It’s always very helpful when there’s strong community support built around the programming language. Python is an open-source language which means that there’s a bunch of resources open for programmers starting from beginners and ending with pros. A lot of Python documentation is available online as well as in Python communities and forums, where programmers and machine learning developers discuss errors, solve problems, and help each other out. Python programming language is absolutely free as is the variety of useful libraries and tools.
- **Growing Popularity:** As a result of the advantages discussed above, Python is becoming more and more popular among data scientists. This means it’s easier to search for developers and replace team players if required. Also, the cost of their work may be not as high as when using a less popular programming language.

Future Scope



Let’s dig deeper and see how Data Science is being used in various domains.

- How about if you could understand the precise requirements of your customers from the existing data like the customer’s past browsing history, purchase history, age and income. No doubt you had all this data earlier too, but now with the vast amount and variety of data, you can train models more effectively and recommend the product to your customers with more precision. Wouldn’t it be amazing as it will bring more business to your organization?

Let’s take a different scenario to understand the role of Data Science in decision making. How about if your car had the intelligence to drive you home? The self-driving cars collect live data from sensors, including radars, cameras, and lasers to create a map of its surroundings. Based

- on this data, it takes decisions like when to speed up, when to speed down, when to overtake, where to take a turn – making use of advanced machine learning algorithms.
- Let’s see how Data Science can be used in predictive analytics. Let’s take weather forecasting as an example. Data from ships, aircraft, radars, satellites can be collected and analyzed to build models. These models will not only forecast the weather but also help in predicting the occurrence of any natural calamities. It will help you to take appropriate measures beforehand and save many precious lives.

System Requirements

□**Operating system:** Windows 7 or newer, 64-bitmacOS 10.13+, or Linux, including Ubuntu, RedHat, CentOS 6+, and others.

□**System architecture:** Windows- 64-bit x86, 32-bitx86; MacOS- 64-bit x86; Linux- 64-bit x86, 64-bitPower8/Power9.

□**Disk Space:** Minimum 5 GB disk space to download and install anaconda distribution.

□**RAM**: 2 GB RAM recommended.

□**Graphics:** For neural networks in Machine Learning, better graphics cards will yield faster results with some high-end graphics cards created especially for Machine Learning purposes. However, Google Colab or some similar website can be used to perform same tasks using cloud computing, without needing a graphics card.

Objectives

The goal of this project is to write a python program to implement a few Machine Learning algorithms, namely, Linear Regression, Logistic Regression, Decision Tree and Random Forest, with a graphical user interface, using Tkinter, such that users without much knowledge of programming or Machine Learning can use these methods with ease. The objectives are as follows:

□The Tkinter application should allow the user to select .csv(comma separated values) files to use as the data set.

□The user should be able to view the file as a table in order to properly select the variables.

□The users should be able to select the independent variables as well as the dependent variable.

□The program should allow the user to enter the test size and random state for the train test split operation.

□The program should allow the user to obtain the coefficients and scatter plot in case of Linear Regression.

□The user should be able to view the confusion matrix and classification report for Logistic, Regression, Decision Tree and Random Forest.

□The user should be able to view the mean absolute error, mean squared error and root mean squared error in all the cases.

□The program is to be written in such a way that other Machine Learning algorithms can be added without much difficulty.

SOURCE CODE

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore') from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
import tkinter as tk
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV, cross_val_score
from imblearn.over_sampling import SMOTE
!pip install imblearn
df = pd.read_csv("winequality-red.csv")
df.columns
plt.figure(figsize=(10, 6))
        sns.countplot(df["quality"], palette="muted")
        df["quality"].value_counts()
print("Rows, columns: " + str(df.shape))
df.head()
df.isnull().sum()
df.info()
df.describe()
fig, ax = plt.subplots(ncols=6, nrows=2,figsize=(20,20))
index = 0
ax = ax.flatten()
for col, value in df.items():
if col != 'type':
```

```
sns.boxplot(y=col, data=df, ax=ax[index]) index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)


fig, ax = plt.subplots(ncols=6, nrows=2, figsize=(20,10))
index = 0
ax = ax.flatten()
for col, value in df.items():
if col != 'type':
sns.distplot(value, ax=ax[index]) index += 1
plt.tight_layout(pad=0.5, w_pad=0.7, h_pad=5.0)
#Here we see that fixed acidity does not give any specification to classify the quality
 fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'fixed acidity', data = df)
#Here we see that its quite a downing trend in the volatile acidity as we go higher the
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'volatile acidity', data = df)
#Composition of citric acid go higher as we go higher in the quality of the wine
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'citric acid', data = df)
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'residual sugar', data = df)
#Composition of chloride also go down as we go higher in the quality of the wine
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'chlorides', data = df)
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'free sulfur dioxide', data = df)
 fig = plt.figure(figsize = (10,6))


sns.barplot(x = 'quality', y = 'total sulfur dioxide', data = df)
#Sulphates level goes higher with the quality of wine
fig = plt.figure(figsize = (10,6))
```

```
sns.barplot(x = 'quality', y = 'sulphates', data = df)
#Alcohol level also goes higher as te quality of wine increases
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'alcohol', data = df)
#Making binary classificaion for the response variable.
#Dividing wine as good and bad by giving the limit for the quality
bins = (2, 6.5, 8)
group_names = ['bad', 'good']
df['quality'] = pd.cut(df['quality'], bins = bins, labels = group_names)
#Now lets assign a labels to our quality variable
label_quality = LabelEncoder()
#Bad becomes 0 and good becomes 1
df['quality'] = label_quality.fit_transform(df['quality'])
df['quality'].value_counts()
sns.countplot(df['quality'])
sns.scatterplot(x='fixed acidity', y='density', data=df)
plt.figure(figsize=(12,8),dpi=200)
sns.scatterplot(x='fixed acidity',y='density', data=df,hue='quality', palette='viridis')
sns.scatterplot(x='volatile acidity', y='density', data=df)
plt.figure(figsize=(12,8),dpi=200)
sns.scatterplot(x='volatile acidity',y='density', data=df,hue='quality', palette='viridis’)
sns.scatterplot(x='citric acid', y='density', data=df)
plt.figure(figsize=(12,8),dpi=200)
sns.scatterplot(x='citric acid',y='density', data=df,hue='quality', palette='viridis')
sns.scatterplot(x='alcohol', y='density', data=df)
plt.figure(figsize=(12,8),dpi=200)
sns.scatterplot(x='alcohol',y='density', data=df,hue='quality', palette='viridis')
sns.scatterplot(x='residual sugar', y='density', data=df)
plt.figure(figsize=(12,8),dpi=200)
```

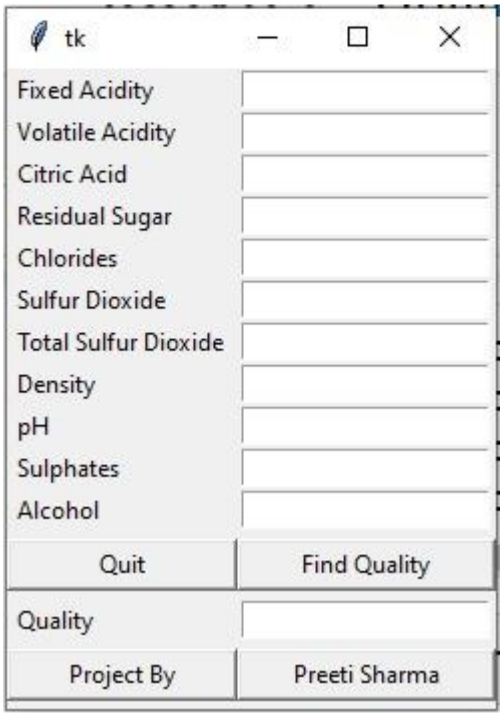
```
sns.scatterplot(x='residual sugar',y='density', data=df,hue='quality', palette='viridis’)
corr = df.corr()
#Let's look at the correlation among the variables using Correlation chart
colormap = plt.cm.viridis
plt.figure(figsize=(12,12))
plt.title('Correlation of Features', y=1.05, size=15)
sns.heatmap(df.astype(float).corr(),linewidths=0.1,vmax=1.0, square=True,
linecolor='white', annot=True)
Data Transformation We want to transfer the score(num) to low-medium-high quality
level(categorical) by:
3,4 -> low
5,6 -> medium
7,8,9 -> high
quality = df["quality"].values
category = []
for num in quality:
if num < 5:
category.append("Low")
elif num > 6:
category.append("High")
else:
category.append("Medium")
 [(i, category.count(i)) for i in set(category)]
#Now seperate the dataset as response variable and feature variabes
X = df.drop('quality', axis = 1)
y = df['quality']
#Train and Test splitting of data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
#Applying Standard scaling to get optimized result
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
```

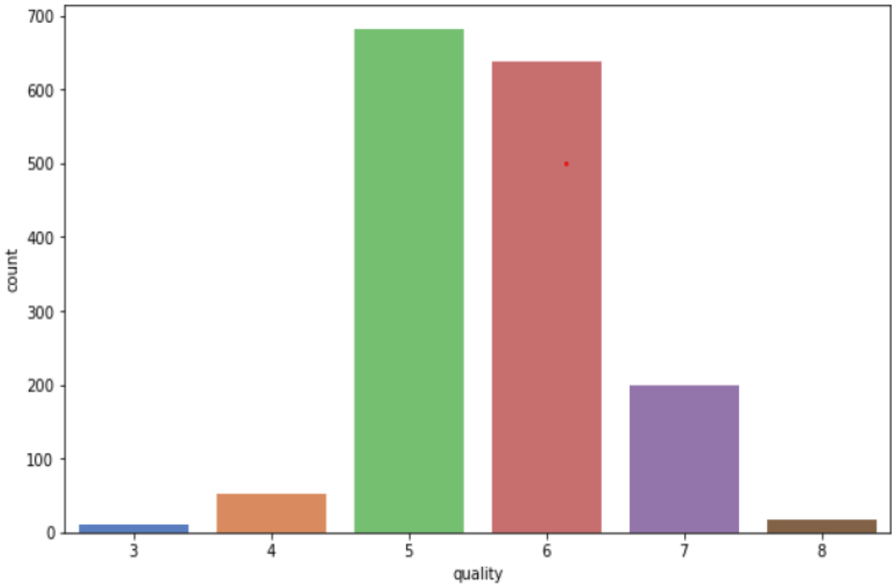
```
X_test = sc.fit_transform(X_test)
rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train, y_train)
pred_rfc = rfc.predict(X_test)
#Let's see how our model performed
print(classification_report(y_test, pred_rfc))
#Confusion matrix for the random forest classification
print(confusion_matrix(y_test, pred_rfc))
sgd = SGDClassifier(penalty=None)
sgd.fit(X_train, y_train)
pred_sgd = sgd.predict(X_test)
svc = SVC()
svc.fit(X_train, y_train)
pred_svc = svc.predict(X_test)
print(classification_report(y_test, pred_svc))
# @hidden_cell
# relabel back : 0 means good, 1 for low, 2 for medium for better visualization
y_test_re = list(y_test)
for i in range(len(y_test_re)):
if y_test_re[i] == 0:
y_test_re[i] = "good"
if y_test_re[i] == 1:
y_test_re[i] = "low"
if y_test_re[i] == 2:
y_test_re[i] = "medium"
pred_sgd_re = list(pred_sgd)
for i in range(len(pred_sgd_re)):
if pred_sgd_re[i] == 0:
pred_sgd_re[i] = "good"
if pred_sgd_re[i] == 1:
pred_sgd_re[i] = "low"
if pred_sgd_re[i] == 2:
pred_sgd_re[i] = "medium"
y_actu = pd.Series(y_test_re, name='Actual')
y_pred = pd.Series(pred_sgd_re, name='Predicted')
svm_confusion = pd.crosstab(y_actu, y_pred)
```

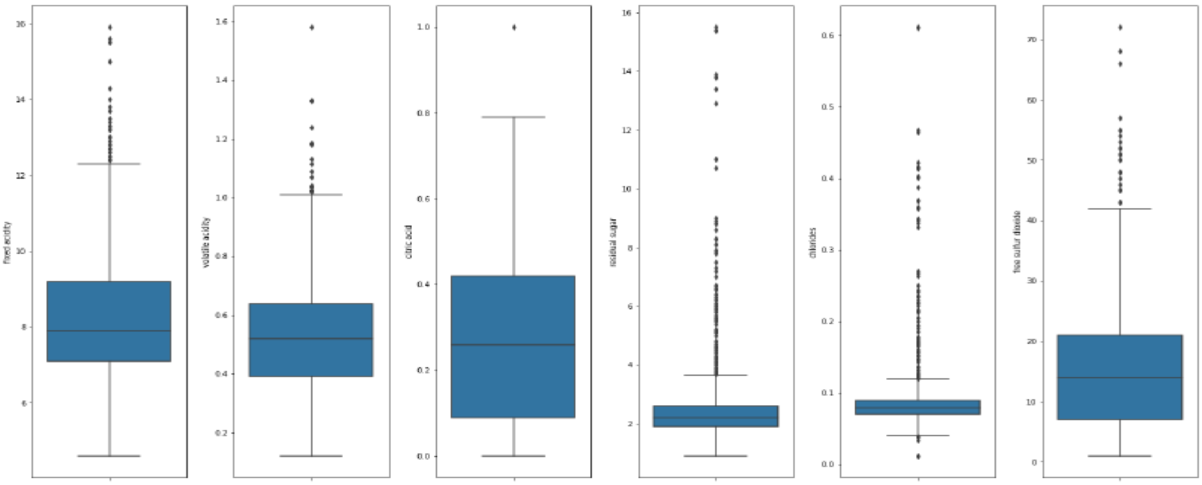
```
svm_confusion
Grid Search CV
#Finding best parameters for our SVC model
param = {
'C': [0.1,0.8,0.9,1,1.1,1.2,1.3,1.4],
'kernel':['linear', 'rbf'],
'gamma' :[0.1,0.8,0.9,1,1.1,1.2,1.3,1.4]
}
grid_svc = GridSearchCV(svc, param_grid=param, scoring='accuracy', cv=10)
grid_svc.fit(X_train, y_train)
#Best parameters for our svc model
grid_svc.best_params_
#Let's run our SVC again with the best parameters.
svc2 = SVC(C = 1.2, gamma = 0.9, kernel= 'rbf')
svc2.fit(X_train, y_train)
pred_svc2 = svc2.predict(X_test)
print(classification_report(y_test, pred_svc2))
grid_svc.best_score_
df.columns[:-1]
#Now lets try to do some evaluation for random forest model using cross validation.
rfc_eval = cross_val_score(estimator = rfc, X = X_train, y = y_train, cv = 10)
rfc_eval.mean()
from sklearn.ensemble import AdaBoostClassifier
model3 = AdaBoostClassifier(random_state=1)
model3.fit(X_train, y_train)
y_pred3 = model3.predict(X_test)
print(classification_report(y_test, y_pred3))
from sklearn.ensemble import GradientBoostingClassifier
model4 = GradientBoostingClassifier(random_state=1)
model4.fit(X_train, y_train)
y_pred4 = model4.predict(X_test)
print(classification_report(y_test, y_pred4))
# Filtering df for only good quality
df_temp = df[df['quality']==1]
```

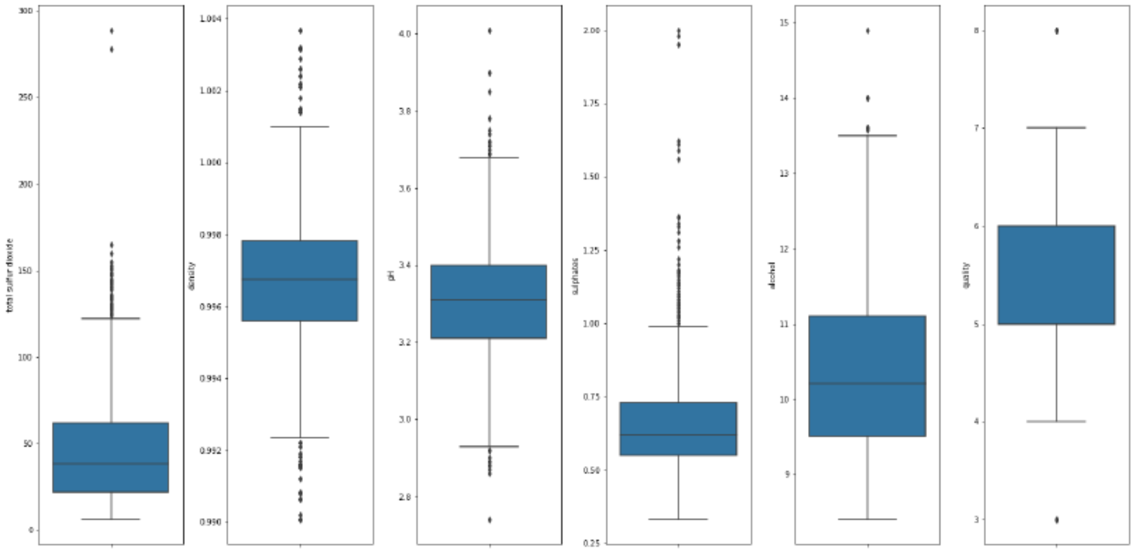
```
df_temp.describe()
# Filtering df for only bad quality
df_temp2 = df[df['quality']==0]
df_temp2.describe()
def showQuality():
new = np.array([[float(e1.get()),float(e2.get()),float(e3.get()),float(e4.get()),float(e5.get())
Ans = RF_clf.predict(new)
fin=str(Ans)[1:-1]
#IT WILL remove[ ]
quality.insert(0, fin)
#Train and evaluate the Random Forest Classifier with Cross Validation
# Instantiate the Random Forest Classifier
RF_clf = RandomForestClassifier(random_state=0)
# Compute k-fold cross validation on training dataset and see mean accuracy score
cv_scores = cross_val_score(RF_clf,X_train, y_train, cv=10, scoring='accuracy')
#Perform predictions
RF_clf.fit(X_train, y_train)
pred_RF = RF_clf.predict(X_test)
master = tk.Tk()
tk.Label(master, text="Fixed Acidity", anchor="nw", width=15).grid(row=0)
tk.Label(master, text="Volatile Acidity", anchor="nw", width=15).grid(row=1)
tk.Label(master, text="Citric Acid", anchor="nw", width=15).grid(row=2)
tk.Label(master, text="Residual Sugar", anchor="nw", width=15).grid(row=3)
tk.Label(master, text="Chlorides", anchor="nw", width=15).grid(row=4)
tk.Label(master, text="Sulfur Dioxide", anchor="nw", width=15).grid(row=5)
tk.Label(master, text="Total Sulfur Dioxide", anchor="nw", width=15).grid(row=6)
tk.Label(master, text="Density", anchor="nw", width=15).grid(row=7)
tk.Label(master, text="pH", anchor="nw", width=15).grid(row=8)
tk.Label(master, text="Sulphates", anchor="nw", width=15).grid(row=9)
tk.Label(master, text="Alcohol", anchor="nw", width=15).grid(row=10)
tk.Label(master, text = "Quality", anchor="nw", width=15).grid(row=13)
e1 = tk.Entry(master)
e2 = tk.Entry(master)
e3 = tk.Entry(master)
e4 = tk.Entry(master)
e5 = tk.Entry(master)
e6 = tk.Entry(master)
e7 = tk.Entry(master)
e8 = tk.Entry(master)
e9 = tk.Entry(master)
e10 = tk.Entry(master)
e11 = tk.Entry(master)
quality = tk.Entry(master)
e1.grid(row=0, column=1)
e2.grid(row=1, column=1)
e3.grid(row=2, column=1)
e4.grid(row=3, column=1)
e5.grid(row=4, column=1)
e6.grid(row=5, column=1)
e7.grid(row=6, column=1)
e8.grid(row=7, column=1)
e9.grid(row=8, column=1)
e10.grid(row=9, column=1)
e11.grid(row=10, column=1)
quality.grid(row=13, column=1)
```

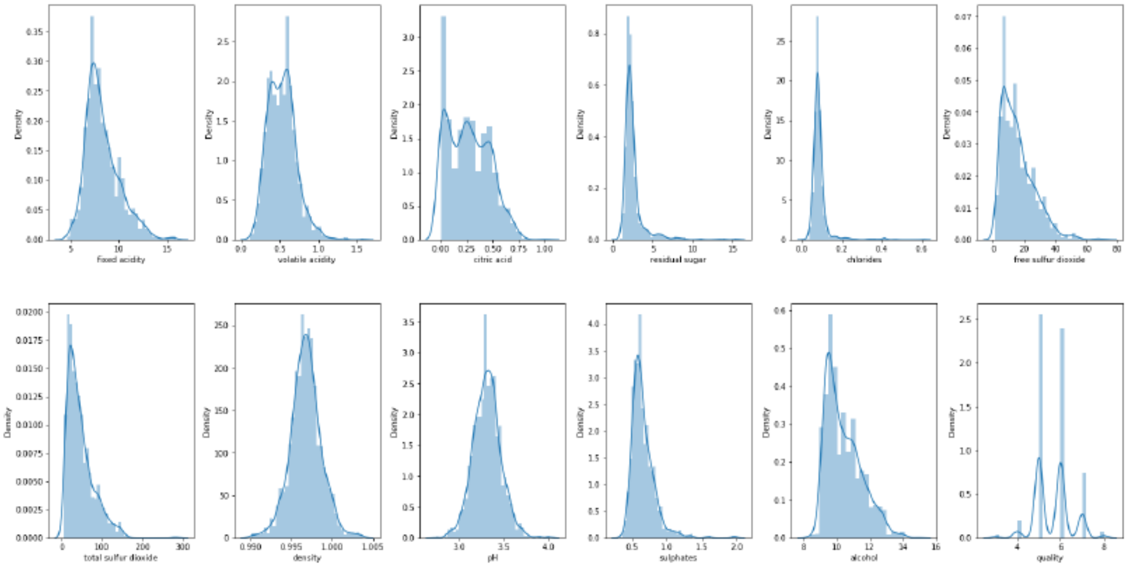
```
tk.Button(master, text='Quit', command=master.destroy,width=15).grid(row=11, column=0,
tk.Button(master, text='Find Quality', command=showQuality,width=17).grid(row=11, colum
tk.Button(master, text='Project By',width=15).grid(row=14, column=0, pady=4)
tk.Button(master, text='Preeti Sharma',width=17).grid(row=14, column=1, pady=4)
master.mainloop()
```

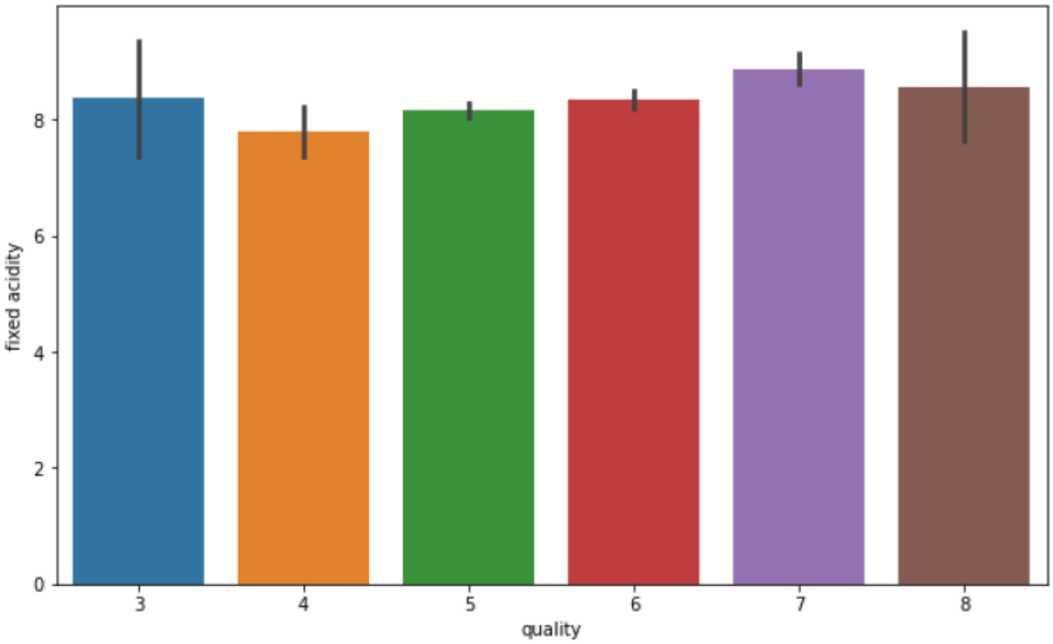


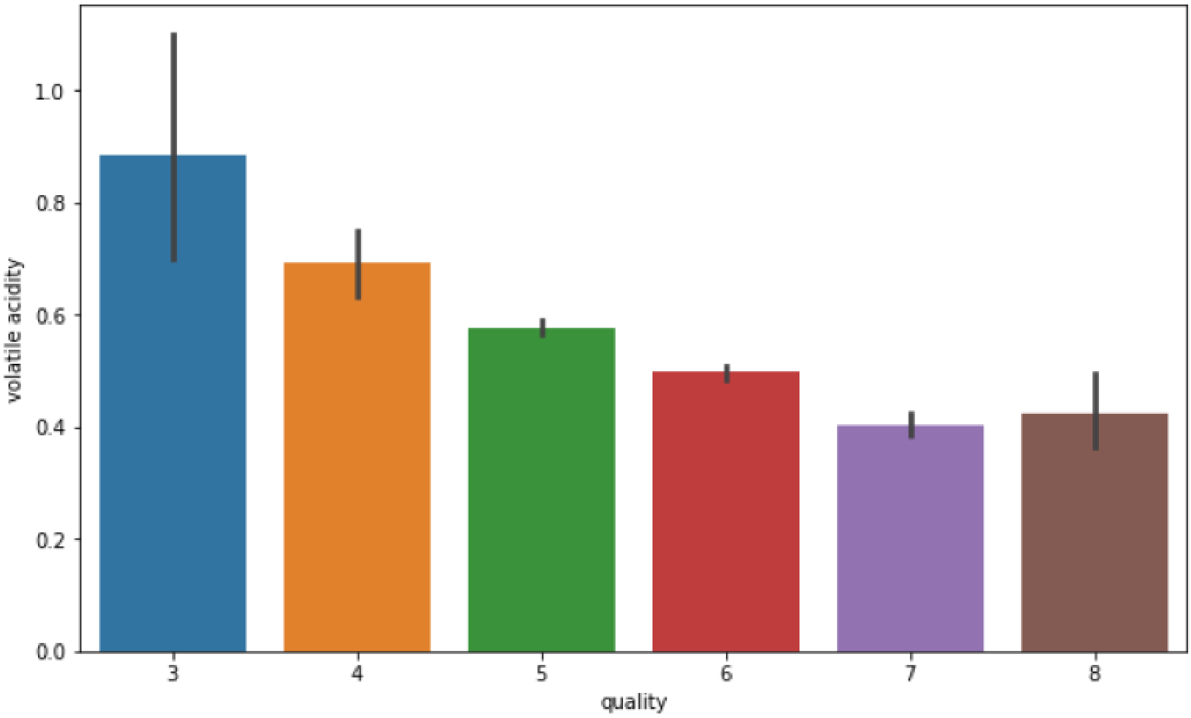


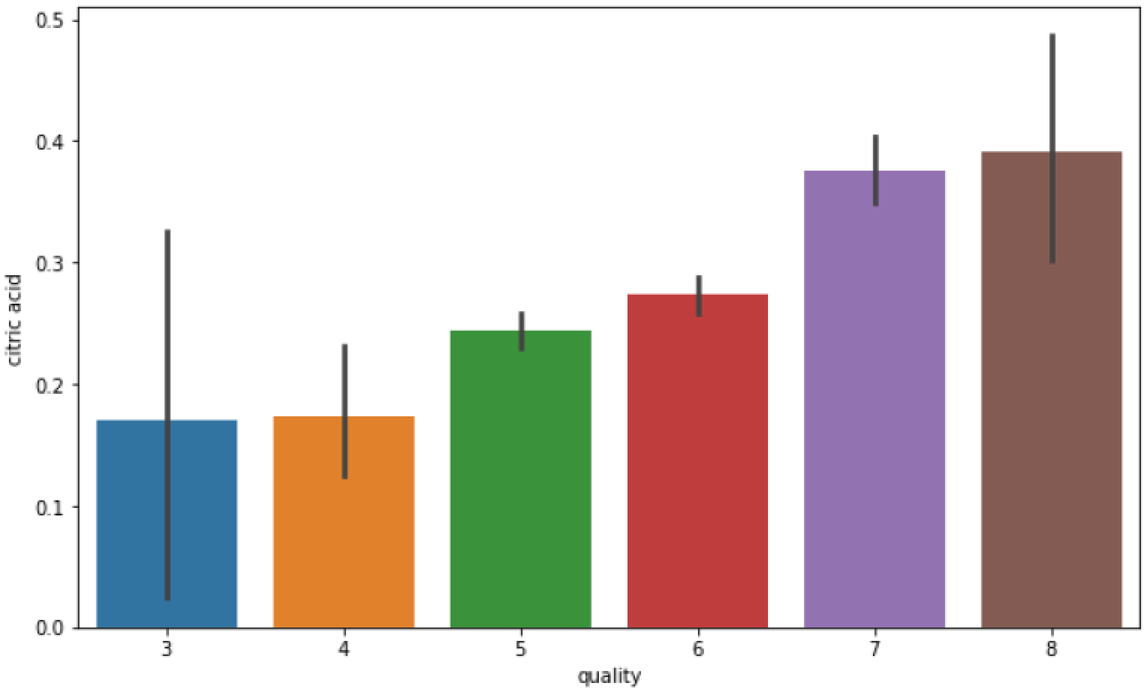

fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide

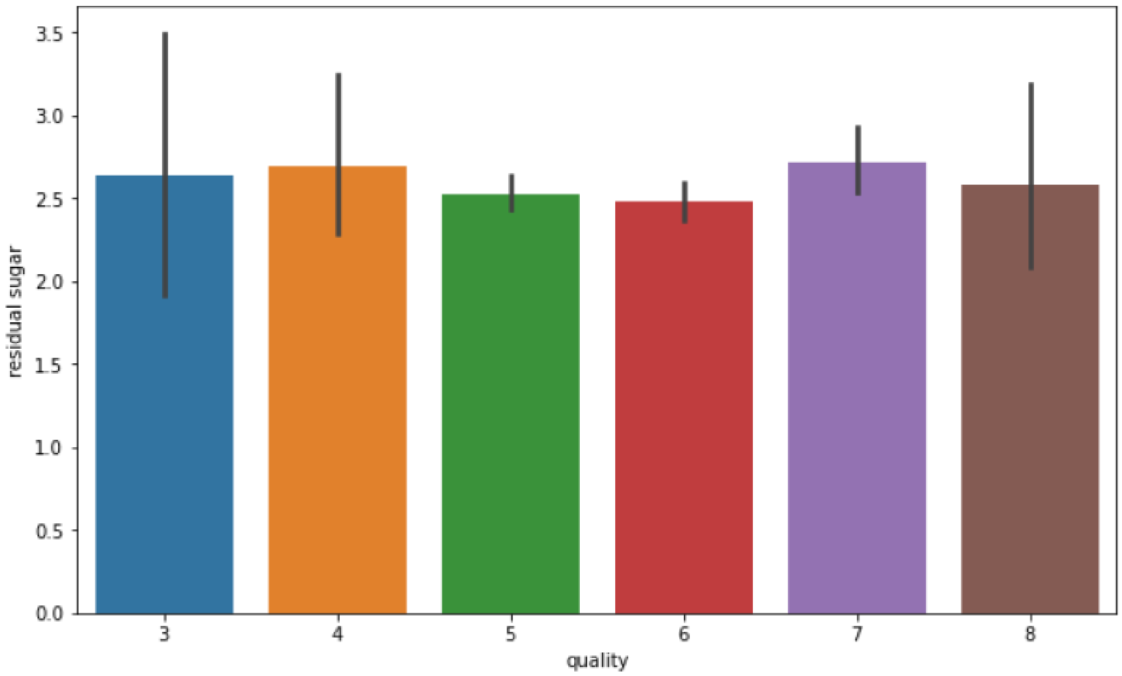

total sulfur dioxide
density
pH
sulphates
alcohol
quality

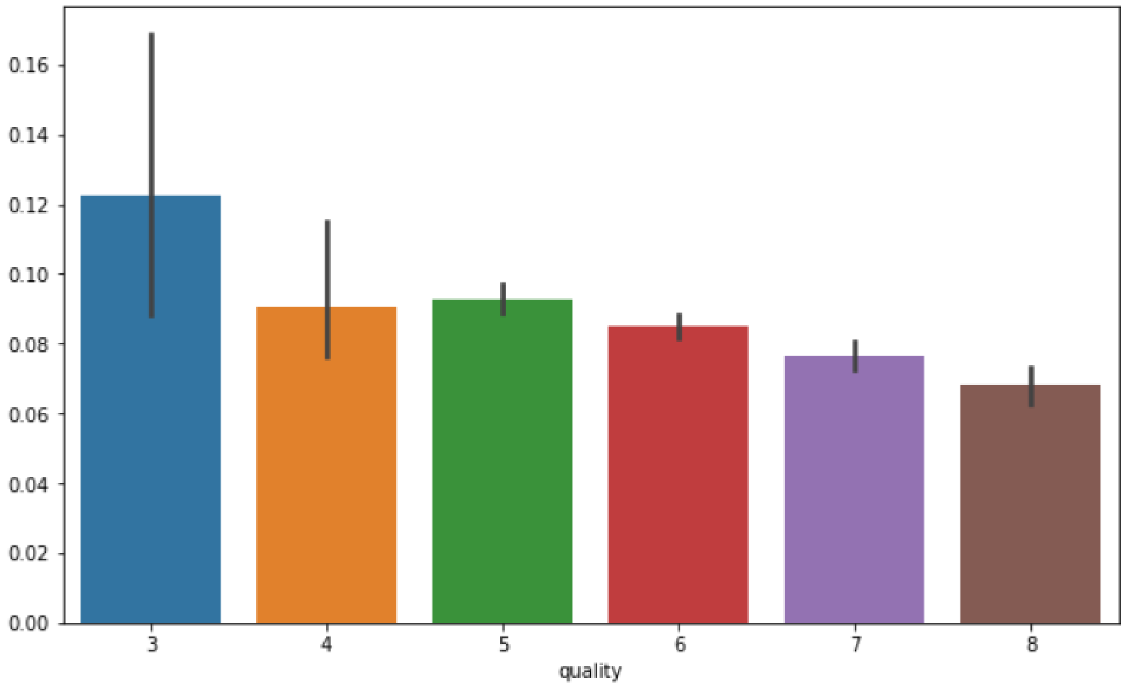

Density
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol
quality

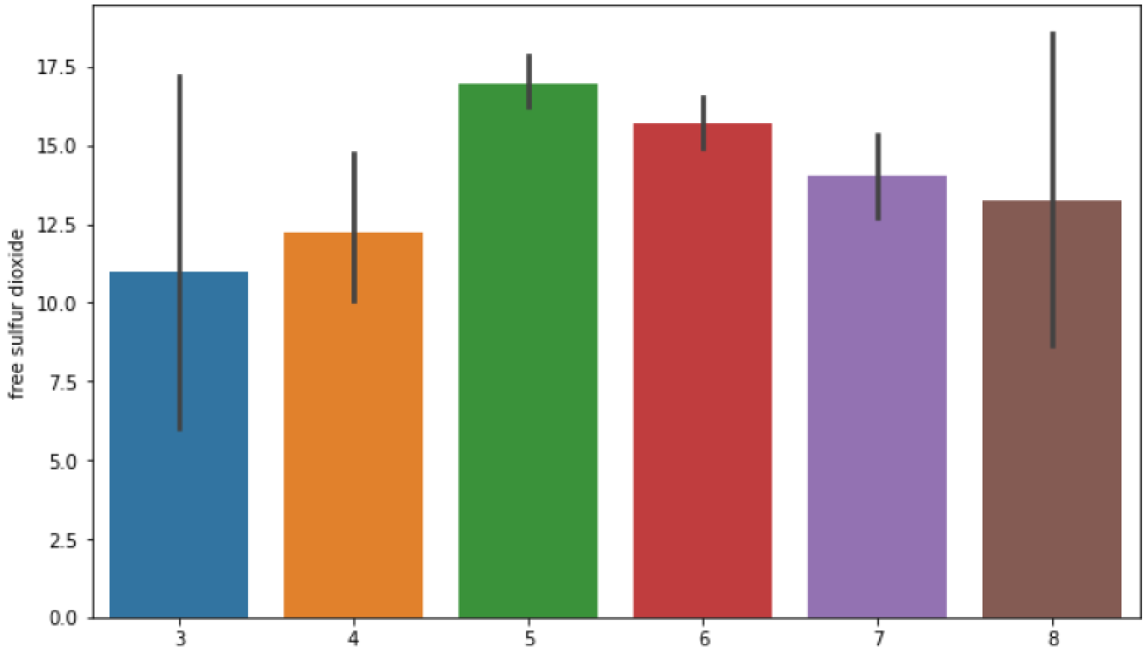

fixed acidity
0
2
4
6
8
3
4
5
6
7
8
quality

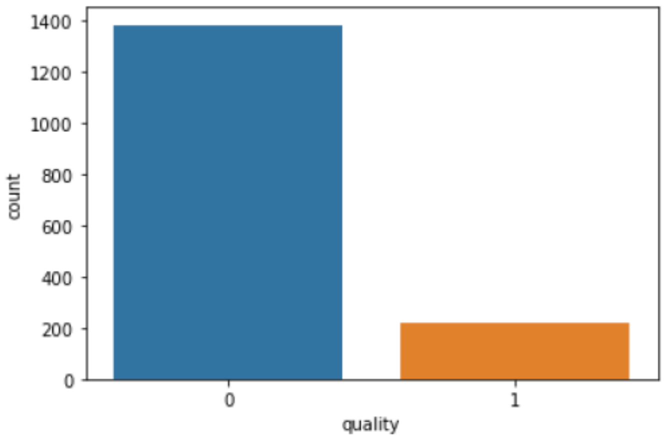

volatile acidity
0.0
0.2
0.4
0.6
0.8
1.0
3
4
5
6
7
8
quality

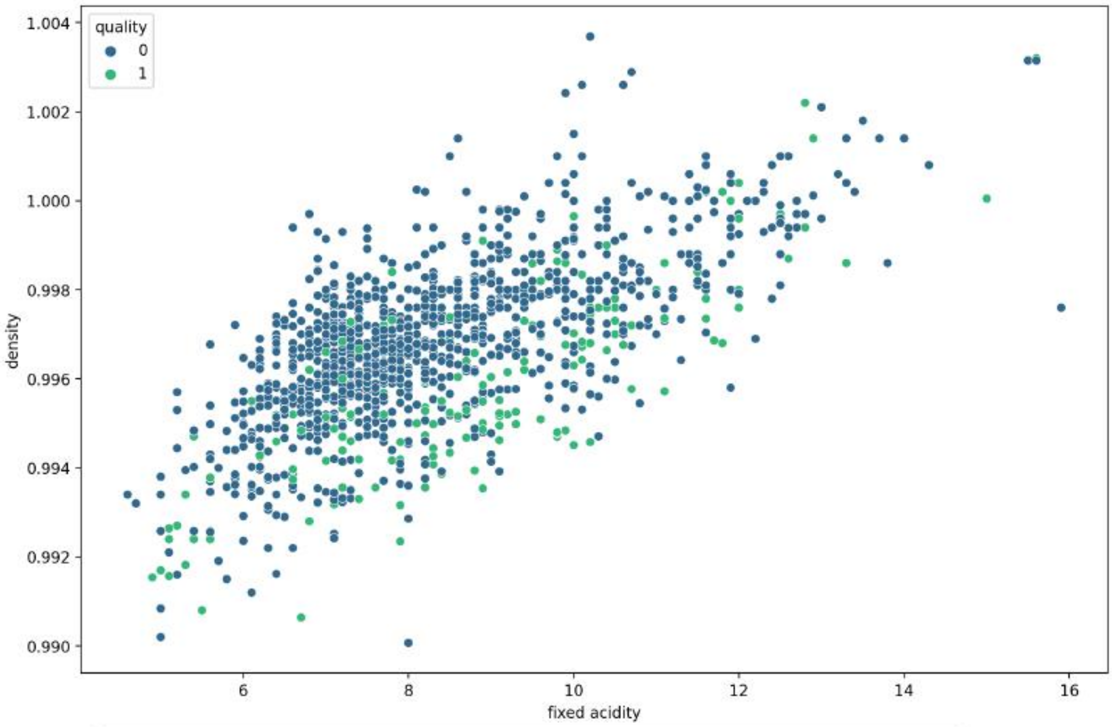

citric acid
0.0
0.1
0.2
0.3
0.4
0.5
3
4
5
6
7
8
quality

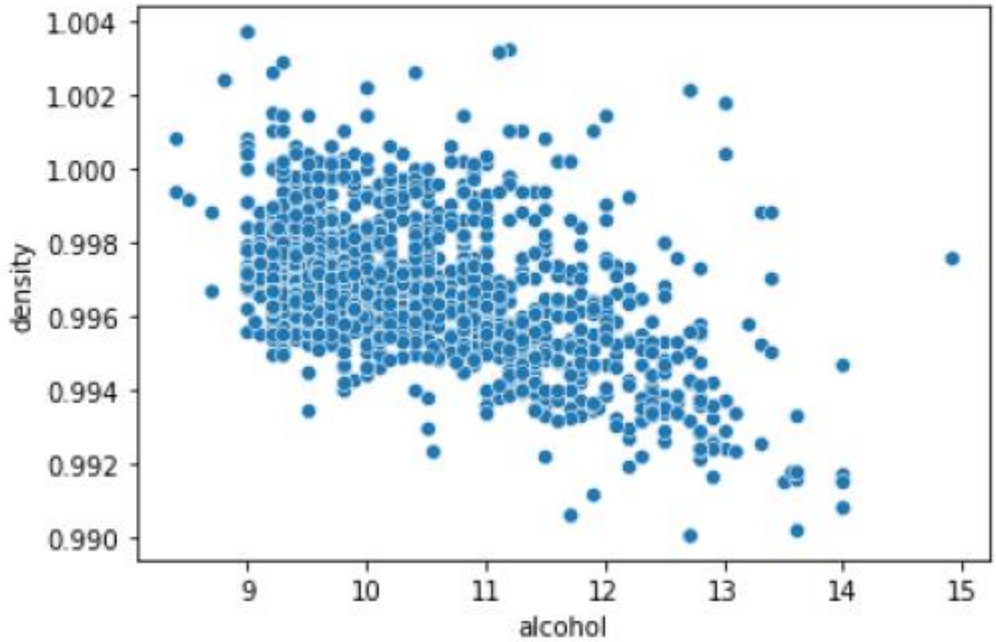

residual sugar
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
3
4
5
6
7
8
quality

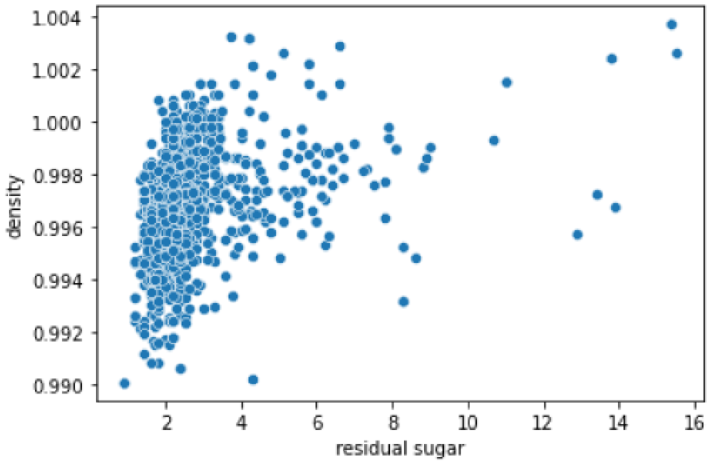

0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
3
4
5
6
7
8
quality

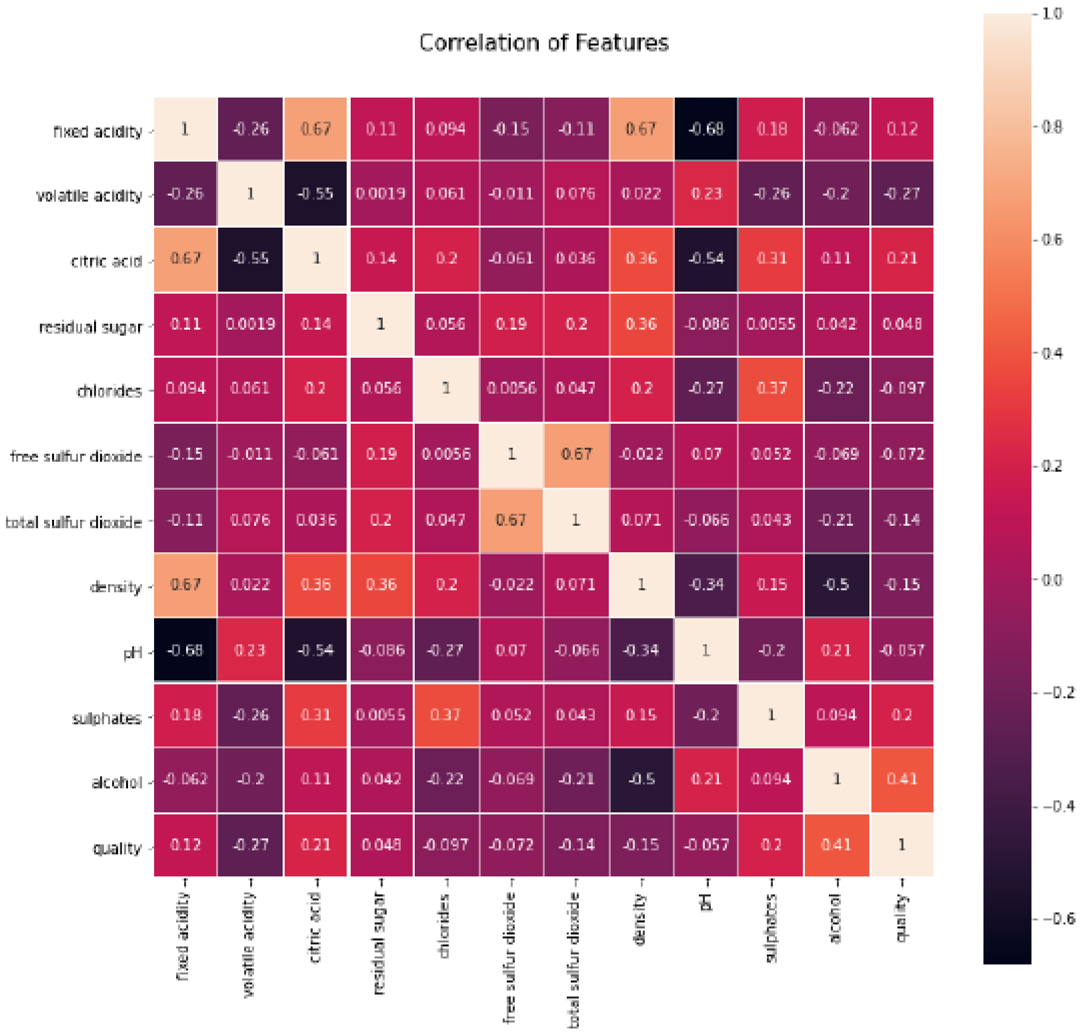

free sulfur dioxide
17.5
15.0
12.5
10.0
7.5
5.0
2.5
0.0
3
4
5
6
7
8
quality


count
1400
1200
1000
800
600
400
200
0
0
1
quality


quality
● 0
● 1
density
1.004
1.002
1.000
0.998
0.996
0.994
0.992
0.990
6
8
10
12
14
16
fixed acidity


density
1.004
1.002
1.000
0.998
0.996
0.994
0.992
0.990
9
10
11
12
13
14
15
alcohol


density
1.004
1.002
1.000
0.998
0.996
0.994
0.992
0.990
2
4
6
8
10
12
14
16
residual sugar



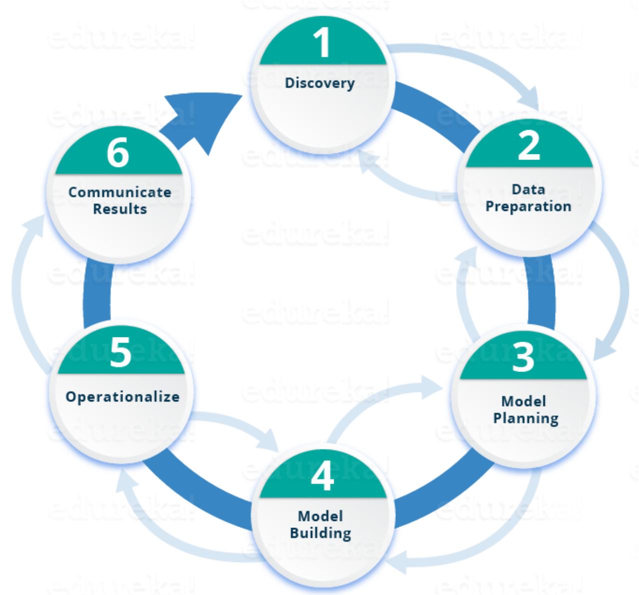
Correlation of Features

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
fixed acidity	1	-0.26	0.67	0.11	0.094	-0.15	-0.11	0.67	-0.68	0.18	-0.062	0.12
volatile acidity	-0.26	1	-0.55	0.0019	0.061	-0.011	0.076	0.022	0.23	-0.26	-0.2	-0.27
citric acid	0.67	-0.55	1	0.14	0.2	-0.061	0.036	0.36	-0.54	0.31	0.11	0.21
residual sugar	0.11	0.0019	0.14	1	0.056	0.19	0.2	0.36	-0.086	0.0055	0.042	0.048
chlorides	0.094	0.061	0.2	0.056	1	0.0056	0.047	0.2	-0.27	0.37	-0.22	-0.097
free sulfur dioxide	-0.15	-0.011	-0.061	0.19	0.0056	1	0.67	-0.022	0.07	0.052	-0.069	-0.072
total sulfur dioxide	-0.11	0.076	0.036	0.2	0.047	0.67	1	0.071	-0.066	0.043	-0.21	-0.14
density	0.67	0.022	0.36	0.36	0.2	-0.022	0.071	1	-0.34	0.15	-0.5	-0.15
pH	-0.68	0.23	-0.54	-0.086	-0.27	0.07	-0.066	-0.34	1	-0.2	0.21	-0.057
sulphates	0.18	-0.26	0.31	0.0055	0.37	0.052	0.043	0.15	-0.2	1	0.094	0.2
alcohol	-0.062	-0.2	0.11	0.042	-0.22	-0.069	-0.21	-0.5	0.21	0.094	1	0.41
quality	0.12	-0.27	0.21	0.048	-0.097	-0.072	-0.14	-0.15	-0.057	0.2	0.41	1

https://drive.google.com/file/d/1cDUxQXLCGLoTDxGWumK67doRO--fEfbN/view?usp=sharing

Conclusion

Main phases of data science life cycle:-



Data science is emerging as a field that is revolutionizing science and industries alike. Work across nearly all domains is becoming more data driven, affecting both the jobs that are available and the skills that are required. As more data and ways of analyzing them become available, more aspects of the economy, society, and daily life will become dependent on data. It is imperative that educators, administrators, and students begin today to consider how to best prepare for and keep pace with this data-driven era of tomorrow.

Undergraduate teaching, in particular, offers a critical link in offering more data science exposure to students and expanding the supply of data science talent.

Customer data is key to making their lives better. Healthcare industries use the data available to them to assist their customers in their everyday life. Data Scientists in these type of industries have the purpose of analyzing the personal data, health history and create products that tackle the problems faced by customers.

From the above instances of data-centric companies, it is clear that each company uses data differently. The use of data varies as per company requirements. Therefore, the purpose of Data Scientists depends on the interests of the company. the purpose of Data Science, we conclude that Data Scientists are the backbone of data-intensive companies. The purpose of Data Scientists is to extract, preprocess and analyze data. Through this, companies can make better decisions. Various companies have their own requirements and use data accordingly. In the end, the goal of Data Scientist to make businesses grow better. With the decisions and insights provided, the companies can adopt appropriate strategies and customize themselves for enhanced customer experience

Bibliography

- Python 3.8.1 documentation, https://docs.python.org/3/
- Graphical User Interfaces with Tk, https://docs.python.org/3/library/tk.html
- Stack Overflow, https://stackoverflow.com
- Why Python is Good for Artificial Intelligence and Machine Learning, retrieved from https://djangostars.com/blog/
- Scikit-learn User Guide, https://scikit-learn.org/stable/user_guide.html