ШИНЖЛЭХ УХААН ТЕХНОЛОГИЙН ИХ СУРГУУЛЬ
Мэдээлэл, холбооны технологийн сургууль

БИЕ ДААЛТЫН АЖЛЫН
ТАЙЛАН
Өгөдлийн холбоо ба сүлжээний үндэс (F.CN204)
2020-2021 оны хичээлийн жилийн хаврын улирал

Бие даалтын ажлын сэдэв:

Хичээл заагч багш: Б.Золзаяа
Бие даалтын ажил гүйцэтгэсэн багийн гишүүд: Оюутан:
С.Тэмүүжин

Red Wine чанарыг олж мэдэхийн тулд машин сургалтыг ашиглах

Өнөө үед улаан дарсны ач холбогдлын талаар бид бүгд мэддэг. Мөн чанар, чанарт нөлөөлж болох хүчин зүйлийн талаар илүү сайн мэдэх нь дээр. Эдгээр нэр томъёогоор бид ML-ийн талаарх компьютерийн мэдлэгээ ашиглаж, өөр өөр физик-химийн шинж чанарт үндэслэн чанарыг урьдчилан таамаглах боломжтой. Одоо энэ нь чанарын үнэлгээ болж, хүмүүс үүнийг мэддэг байх болно.

Би хувьдаа хүмүүс ML-ийн тухай ойлголтыг эргэлзэхийн тулд биш, харин бидний алгоритм дээр үндэслэн харилцаа холбоо, таамаглалын мөн чанарыг судалж, сурахын тулд сурах ёстой гэж би боддог. Ингэснээр тэд үр дүндээ итгэлтэй байж, зөвхөн зарим нэг тал дээр бус зорилтот болон хариу үйлдэл бүхий бүх зүйл дээр илүү сайн үр дүнд хүрч чадна.

Энд улаан дарсны чанарын хувьд би мэдээллийн багцыг archive ics uci ашиглаж байна. Би өгөгдлийн visualization, preprocessing, model prediction онцолж, эцэст нь түүний ач холбогдлыг анхаарч үзэх болно.

IV. Дүгнэлт:

So, the question may arise about its significance. Companies can use this technique of ML for creating better models and by using the dataset with thousands of data. This is just a simple dataset for devising a method for relating the quality of red wine with its physicochemical properties. General laymen can indeed know about what factors can affect the quality of red wine by analyzing the dataset in the visualization part. These algorithms and techniques can not only be used here but in every field of response and target variables, which can indeed boost our capabilities.

Тиймээс түүний ач холбогдлын талаар асуулт гарч ирж магадгүй юм. Компаниуд ML-ийн энэхүү техникийг илүү сайн загвар бүтээх, олон мянган өгөгдөл бүхий өгөгдлийн багцыг ашиглахын тулд ашиглаж болно. Энэ бол улаан дарсны чанарыг физик-химийн шинж чанартай холбох аргыг боловсруулах энгийн мэдээллийн багц юм. Дүрслэх хэсэг дэх өгөгдлийн багцад дүн шинжилгээ хийснээр улаан дарсны чанарт ямар хүчин зүйл нөлөөлж болохыг ерөнхий энгийн хүмүүс мэдэх боломжтой. Эдгээр алгоритм, техникийг зөвхөн энд төдийгүй хариу үйлдэл болон зорилтот хувьсагчдад ашиглах боломжтой бөгөөд энэ нь бидний чадавхийг үнэхээр нэмэгдүүлж чадна.

Red Wine чанарыг олж мэдэхийн тулд машин сургалтыг ашиглах

Өнөө үед улаан дарсны ач холбогдлын талаар бид бүгд мэддэг. Мөн чанар, чанарт нөлөөлж болох хүчин зүйлийн талаар илүү сайн мэдэх нь дээр. Эдгээр нэр томъёогоор бид ML-ийн талаарх компьютерийн мэдлэгээ ашиглаж, өөр өөр физик-химийн шинж чанарт үндэслэн чанарыг урьдчилан таамаглах боломжтой. Одоо энэ нь чанарын үнэлгээ болж, хүмүүс үүнийг мэддэг байх болно.

Би хувьдаа хүмүүс ML-ийн тухай ойлголтыг эргэлзэхийн тулд биш, харин бидний алгоритм дээр үндэслэн харилцаа холбоо, таамаглалын мөн чанарыг судалж, сурахын тулд сурах ёстой гэж би боддог. Ингэснээр тэд үр дүндээ итгэлтэй байж, зөвхөн зарим нэг тал дээр бус зорилтот болон хариу үйлдэл бүхий бүх зүйл дээр илүү сайн үр дүнд хүрч чадна.

Энд улаан дарсны чанарын хувьд би мэдээллийн багцыг archive ics uci ашиглаж байна. Би өгөгдлийн visualization, preprocessing, model prediction онцолж, эцэст нь түүний ач холбогдлыг анхаарч үзэх болно.

I. Dataset

Мэдээллийн багцыг link:
https://archive.ics.uci.edu/dataset/186/wine+quality

pandas номын сан нь өгөгдлийн багцыг заасан URL-аас DataFrame руу ачаалах. Энэ өгөгдлийн багц нь улаан дарсны чанарын тухай бөгөөд UCI Machine Learning Repository дээр байрладаг. Энд pd.read_csv функцийг URL-аас CSV файлыг цэг таслалтайгаар уншихад ашигладаг; хязгаарлагч гэж заасан бөгөөд энэ нь CSV дахь утгууд нь цэгтэй таслалаар тусгаарлагдсан гэсэн үг юм. Өгөгдлийг ачаалсны дараа DataFrame-ийн эхний 10 мөрийг df.head(10)-аар харуулна.

```
import pandas as pd

url =
'http://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/
winequality-red.csv'

df = pd.read_csv(url, sep=';')

df.head(10)
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	5
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	5
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	6
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	5
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	5
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4	5
7	7.3	0.65	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0	7
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5	7
9	7.5	0.50	0.36	6.1	0.071	17.0	102.0	0.9978	3.35	0.80	10.5	5

Python дахь df.shape мөрийг панда DataFrame-тэй ашиглах үед DataFrame-ийн хэмжээсийг төлөөлсөн хэлхээг буцаана.

```
df.shape
```

```
(1599, 12)
```

df.info() оролтын тоо, багана тус бүрийн хоосон оруулгын тоо, багана бүрийн өгөгдлийн төрөл, ашиглаж буй санах ойн хэмжээ зэрэг мэдээлэл.

```
df.info()

df.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   fixed acidity         1599 non-null   float64
 1   volatile acidity      1599 non-null   float64
 2   citric acid           1599 non-null   float64
 3   residual sugar        1599 non-null   float64
 4   chlorides             1599 non-null   float64
 5   free sulfur dioxide   1599 non-null   float64
 6   total sulfur dioxide  1599 non-null   float64
 7   density               1599 non-null   float64
 8   pH                    1599 non-null   float64
 9   sulphates             1599 non-null   float64
 10  alcohol               1599 non-null   float64
 11  quality               1599 non-null   int64  
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
fixed acidity           0
volatile acidity        0
citric acid             0
residual sugar          0
chlorides               0
free sulfur dioxide     0
total sulfur dioxide    0
density                 0
pH                      0
sulphates               0
alcohol                 0
quality                 0
dtype: int64
```

Python дахь df.describe() аргыг панда DataFrame-д ашиглах үед DataFrame-ийн бүх тоон баганын статистик хураангуйг өгдөг.

```
df.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000	1599.000000
mean	8.319637	0.527821	0.270976	2.538806	0.087467	15.874922	46.467792	0.996747	3.311113	0.658149	10.422983	5.636023
std	1.741096	0.179060	0.194801	1.409928	0.047065	10.460157	32.895324	0.001887	0.154386	0.169507	1.065668	0.807569
min	4.600000	0.120000	0.000000	0.900000	0.012000	1.000000	6.000000	0.990070	2.740000	0.330000	8.400000	3.000000
25%	7.100000	0.390000	0.090000	1.900000	0.070000	7.000000	22.000000	0.995600	3.210000	0.550000	9.500000	5.000000
50%	7.900000	0.520000	0.260000	2.200000	0.079000	14.000000	38.000000	0.996750	3.310000	0.620000	10.200000	6.000000
75%	9.200000	0.640000	0.420000	2.600000	0.090000	21.000000	62.000000	0.997835	3.400000	0.730000	11.100000	6.000000
max	15.900000	1.580000	1.000000	15.500000	0.611000	72.000000	289.000000	1.003690	4.010000	2.000000	14.900000	8.000000

```
#Importing required packages.

import matplotlib.pyplot as plt

import seaborn as sb

import numpy as np

%matplotlib inline
```

"чанар" гэсэн шошготой баганаас өвөрмөц утгуудыг гаргаж авдаг.

```
df['quality'].unique()
```

```
array([5, 6, 7, 4, 8, 3])
```

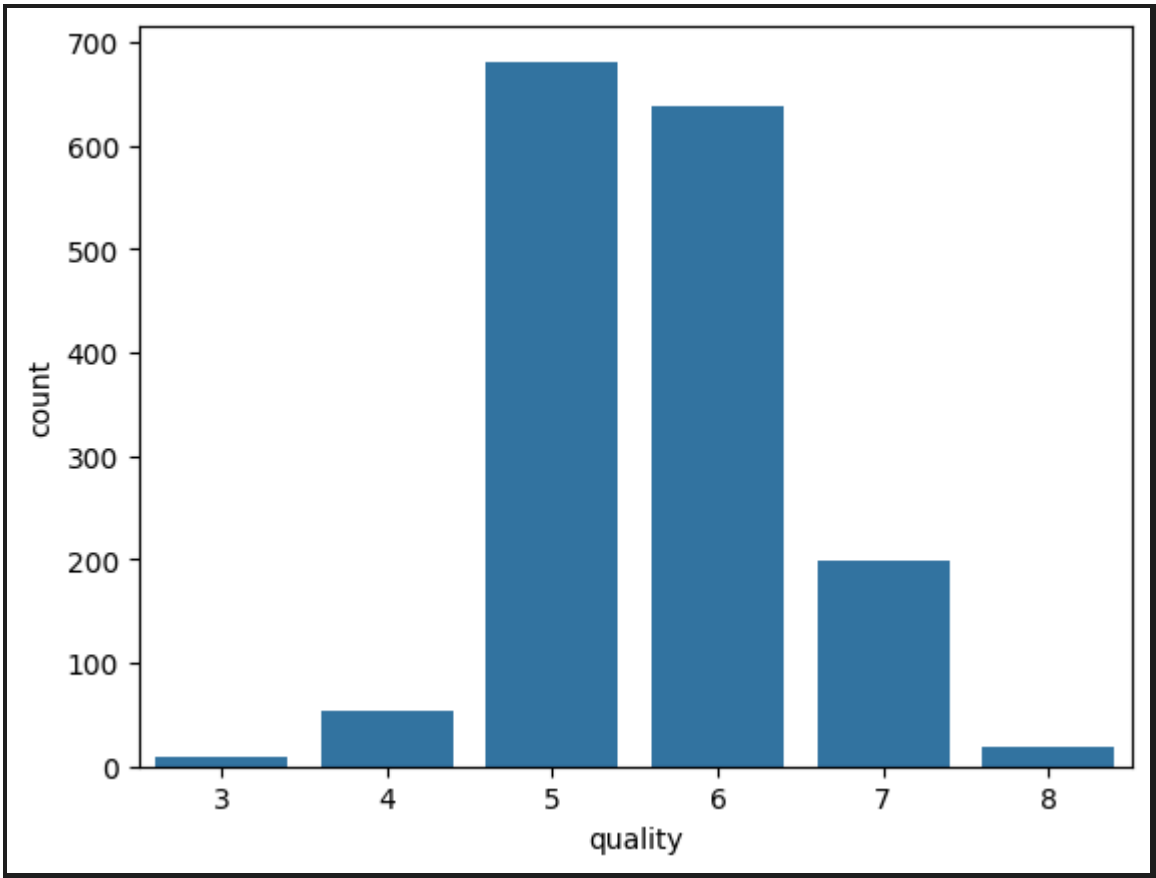
"Чанар" баганад байгаа өвөрмөц утга бүрийн давтамжийг тоолох DataFrame.

```
df['quality'].value_counts()
```

```
quality
5    681
6    638
7    199
4     53
8     18
3     10
Name: count, dtype: int64
```

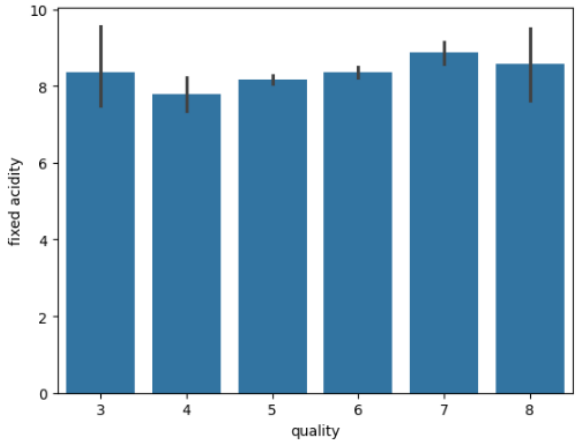
'чанар' баганад ангилсан өгөгдлийн тархалтыг төсөөлөх

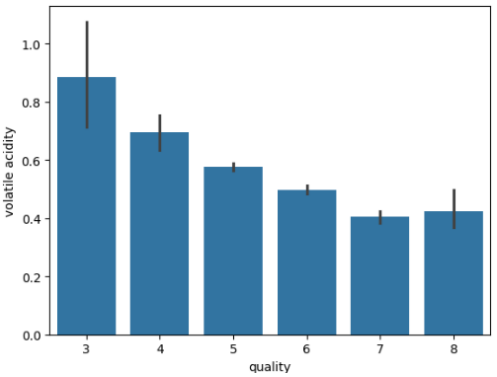
```
sb.countplot(x='quality', data=df)
```

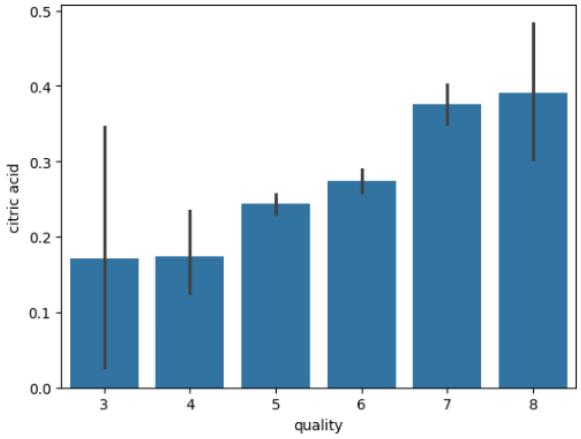


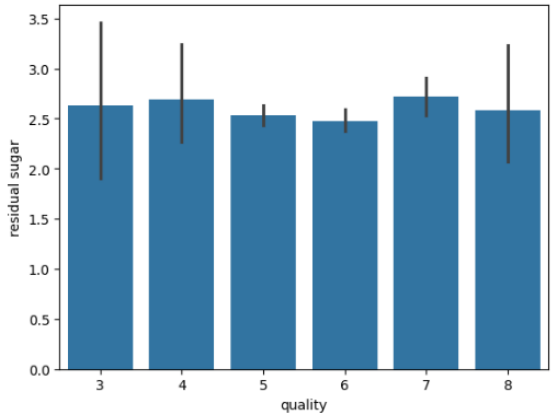
Seaborn's barplot функцийг ашиглан DataFrame df-ийн "чанар" баганын эсрэг тоон багана бҮрийн зураасыг ҮҮсгэнэ.

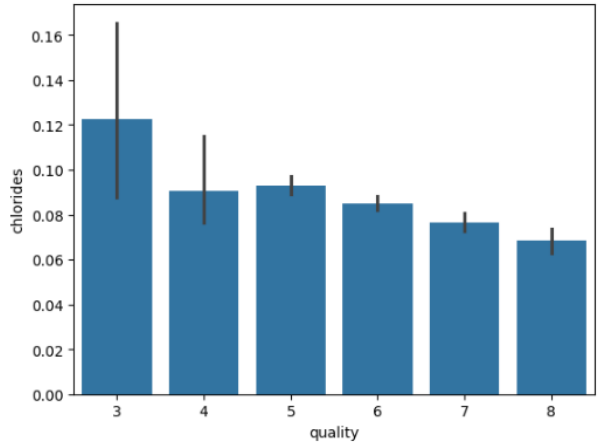
```
df1 = df.select_dtypes([int, float])

for i, col in enumerate(df1.columns):
    plt.figure(i)
    sb.barplot(x='quality', y =col, data=df1)
```

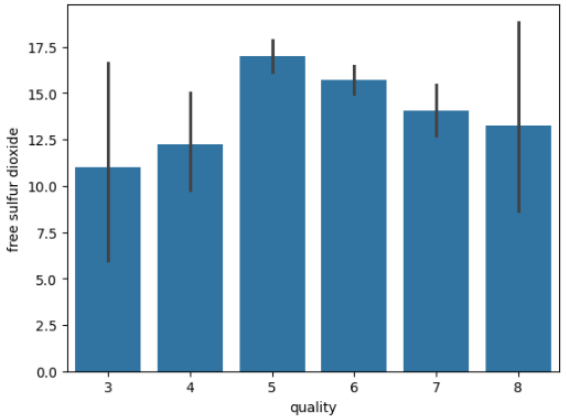

fixed acidity
quality

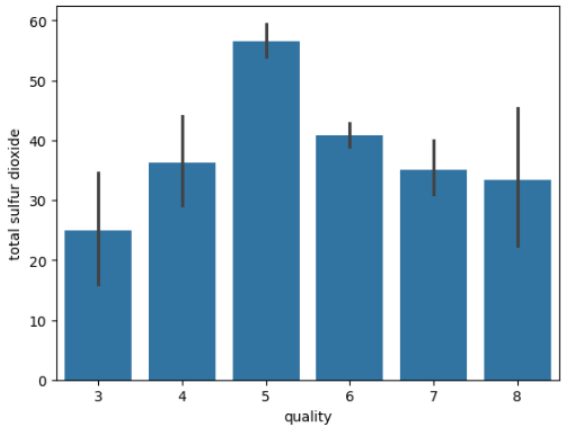

volatile acidity
quality

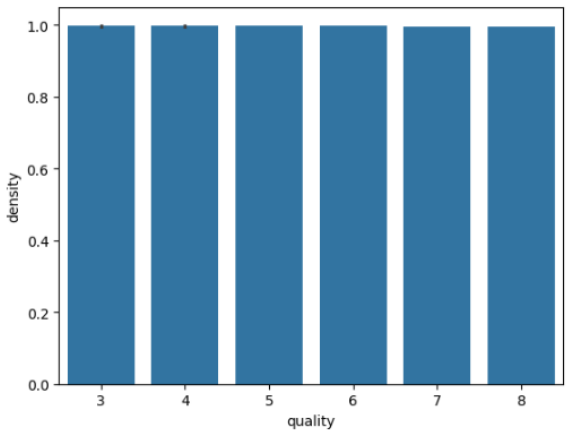

citric acid
quality

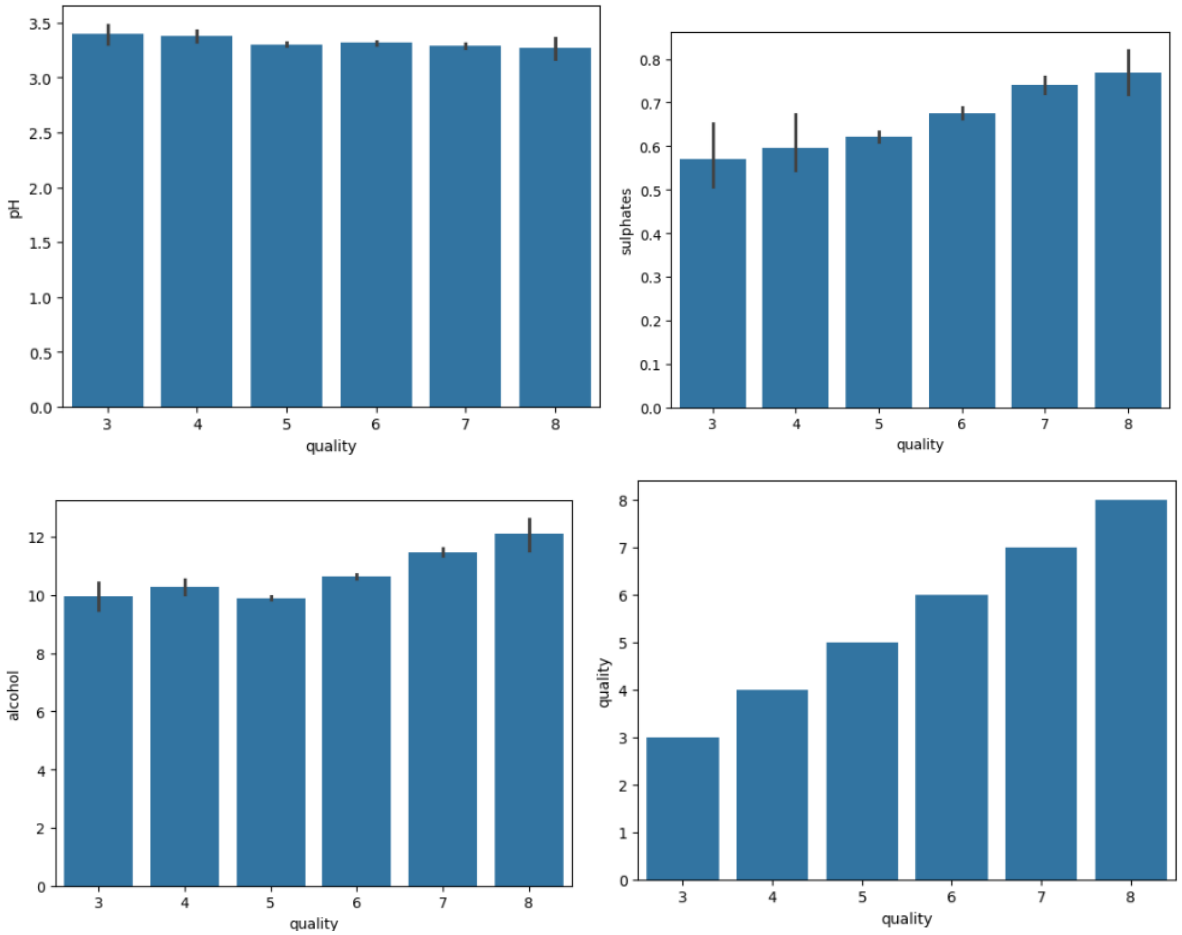

residual sugar
quality


chlorides
quality


free sulfur dioxide
quality

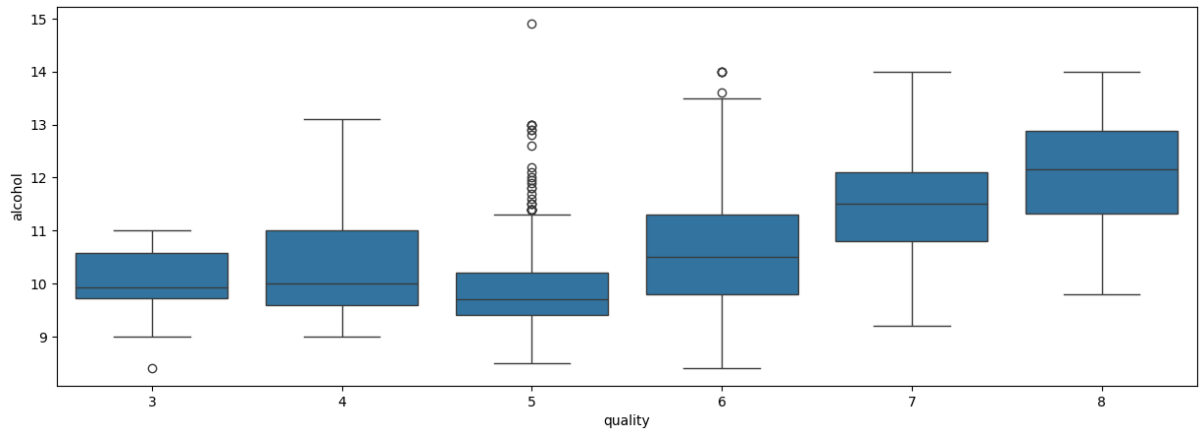

total sulfur dioxide
quality


density
quality



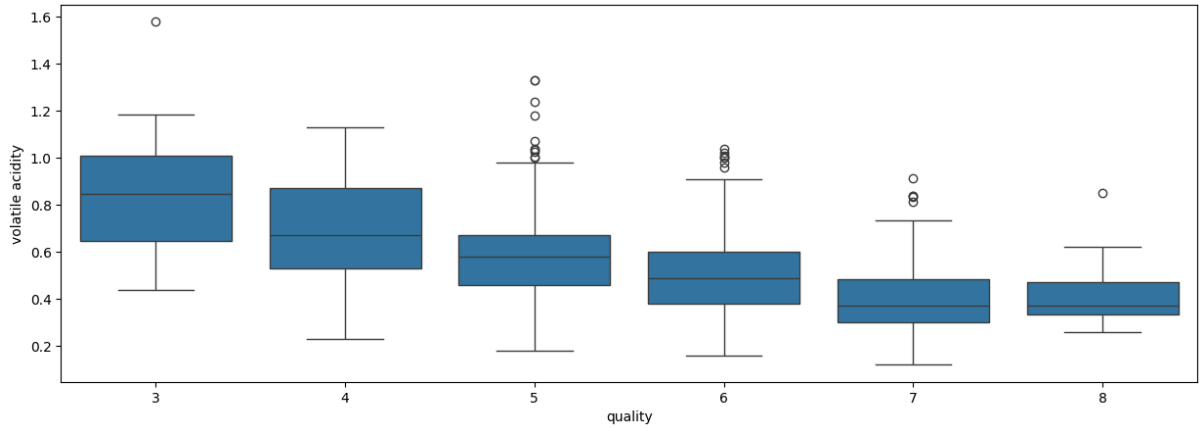
"Чанар" гэсэн категорийн хувьсагчийн янз бүрийн категориудад "архи" тасралтгүй хувьсагчийн тархалтыг төсөөлөх

```
import seaborn as sns

plt.figure(figsize=(15,5))

sns.boxplot(x="quality", y="alcohol",    data=df )
```



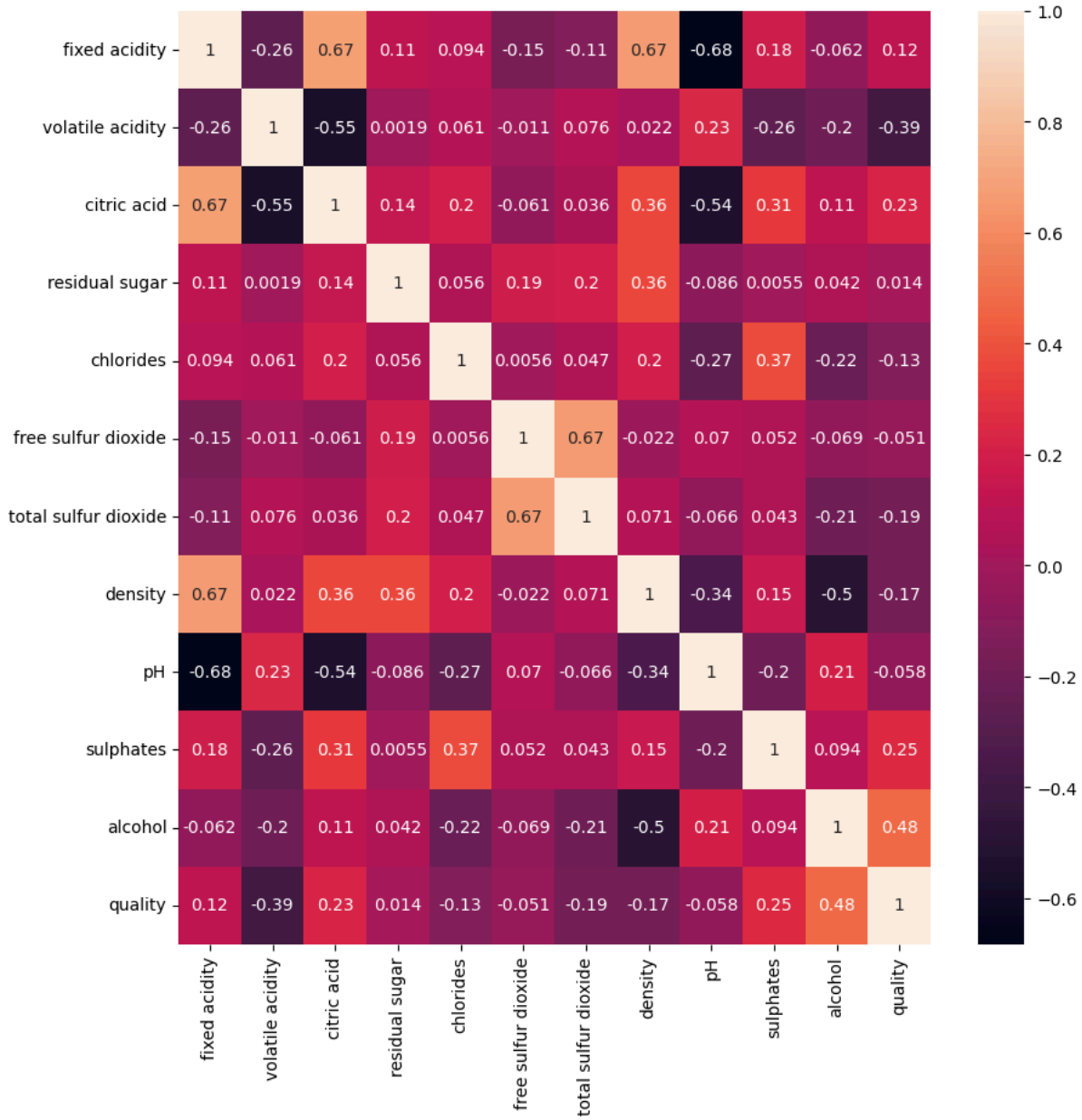
"Чанар" гэсэн категорийн хувьсагчийн янз бүрийн категориудад "volatile acidity" тасралтгүй хувьсагчийн тархалтыг төсөөлөх

```
import seaborn as sns

plt.figure(figsize=(15,5))

sns.boxplot(x="quality", y="volatile acidity",    data=df )
```



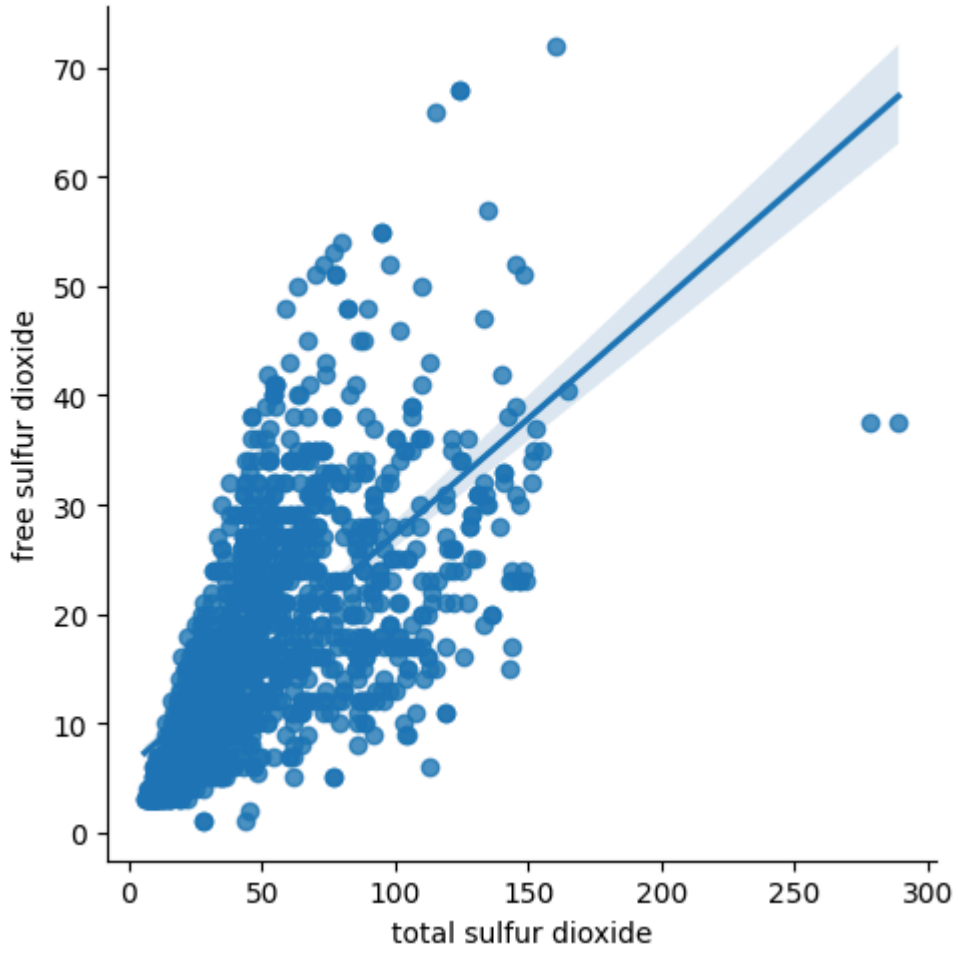
Корреляцийн матрицыг дүрслэхийн тулд Seaborn болон Matplotlib ашиглан дулааны зураглалыг үүсгэдэг

```
plt.figure(figsize=(10,10))

sns.heatmap(df.corr(),color = "k", annot=True)
```



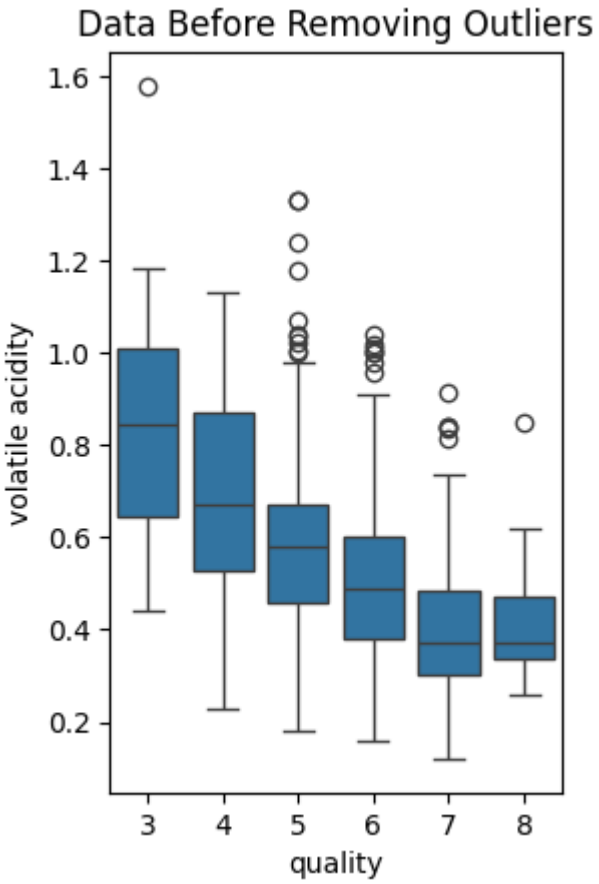
Seaborn's lmplot, энэ нь өгөгдлийн цэгүүдэд тохирох регрессийн шугам бүхий тархалтын график үүсгэдэг функц юм.

```
sns.lmplot(x="total sulfur dioxide", y="free sulfur dioxide", data=df)
```



Matplotlib болон Seaborn-ийг ашиглан дэд зураглалыг тохируулж, гарчигтай хайрцагны зураг үүсгэдэг.

```
plt.subplot(1, 2, 2)

sns.boxplot(x='quality', y='volatile acidity', data=df)

plt.title('Data Before Removing Outliers')
```

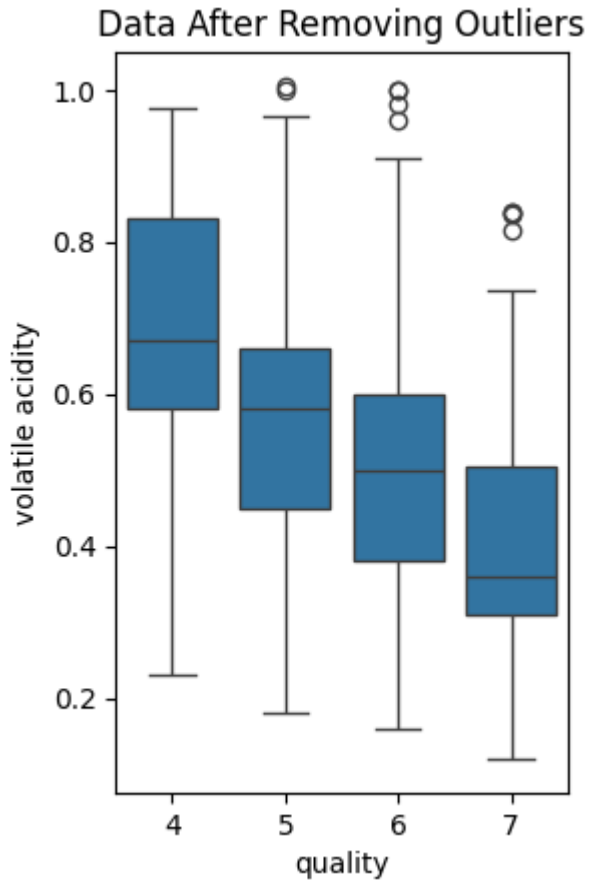


Квартиль хоорондын зай (IQR) аргад тулгуурлан панда DataFrame-аас гадуурх утгыг тодорхойлж устгах.

```
Q1 = df.quantile(0.25)

Q3 = df.quantile(0.75)

IQR = Q3 - Q1

outlier_condition = (df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))

df = df[~outlier_condition.any(axis=1)]
```

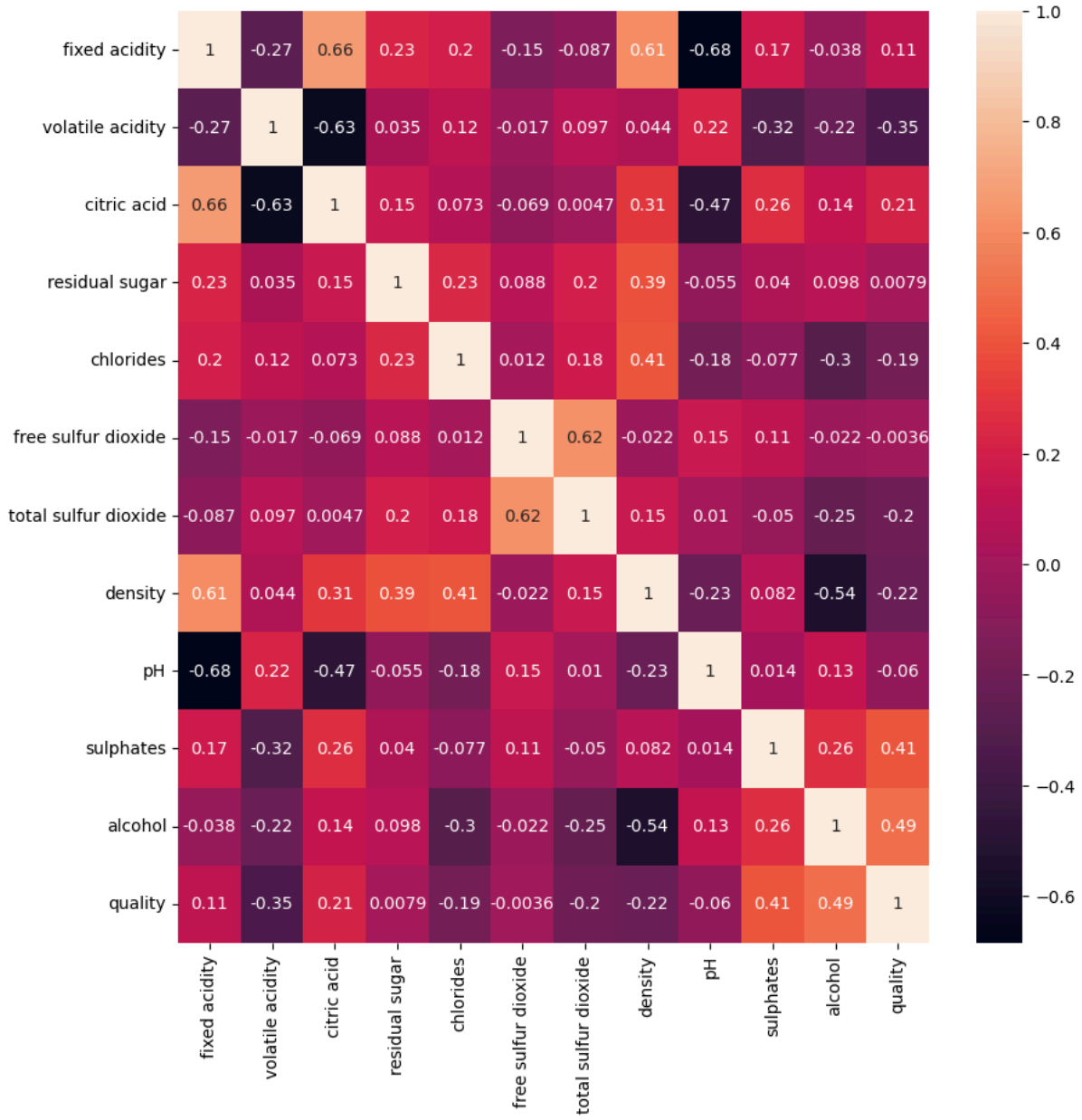
Хоёрдахь дэд графикийг 1 мөр, 2 баганатай зурганд байрлуулж, гадуурх утгыг арилгасны дараа DataFrame df-ийн өөр өөр "чанарын" зэрэглэлд "дэгдэмхий хүчил"-ийн тархалтыг дүрслэн харуулахын тулд хайрцагны график үүсгэнэ.

```
plt.subplot(1, 2, 2)

sns.boxplot(x='quality', y='volatile acidity', data=df)

plt.title('Data After Removing Outliers')
```



Шинэчлэгдсэн кодын блок нь 10-аас 10 инчийн хэмжээтэй дүрсийг тохируулж, хэтийн утгыг арилгасны дараа DataFrame df-ийн хамаарлын матрицыг дүрслэн харуулах дулааны зураглалыг үүсгэдэг.

```
plt.figure(figsize=(10,10))

sns.heatmap(df.corr(),color = "k", annot=True)
```




```
df.describe()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
count	1179.000000	1179.000000	1179.000000	1179.000000	1179.000000	1179.000000	1179.000000	1179.000000	1179.000000	1179.000000	1179.000000	1179.000000
mean	8.162002	0.523066	0.246760	2.185411	0.078586	15.020356	42.268024	0.996584	3.324623	0.631264	10.350792	5.623410
std	1.458270	0.164231	0.179441	0.440972	0.014317	8.792916	26.106438	0.001593	0.131731	0.116098	0.963954	0.721248
min	5.100000	0.120000	0.000000	1.200000	0.041000	1.000000	6.000000	0.992360	2.940000	0.330000	8.700000	4.000000
25%	7.100000	0.390000	0.080000	1.900000	0.069000	8.000000	22.000000	0.995520	3.230000	0.550000	9.500000	5.000000
50%	7.800000	0.520000	0.240000	2.100000	0.078000	13.000000	36.000000	0.996600	3.330000	0.610000	10.100000	6.000000
75%	9.000000	0.630000	0.390000	2.500000	0.087000	20.000000	56.000000	0.997600	3.410000	0.700000	11.000000	6.000000
max	12.300000	1.005000	0.730000	3.600000	0.119000	42.000000	122.000000	1.001000	3.680000	0.980000	13.400000	7.000000

DataFrame df доторх "чанар"-ын утгуудыг "муу" ба "сайн" гэсэн хоёр бүлэгт ангилж эсвэл ангилна уу.

```
bins = (2, 6.5, 8)

group_names = ['bad', 'good']

df['quality'] = pd.cut(df['quality'], bins = bins, labels =
group_names)
```

```
from sklearn.ensemble import RandomForestClassifier,
GradientBoostingClassifier

from sklearn.svm import SVC, LinearSVC

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, StratifiedKFold

from sklearn.linear_model import LogisticRegression

from sklearn.neighbors import KNeighborsClassifier

from sklearn.naive_bayes import GaussianNB

from sklearn.metrics import accuracy_score
```

Scikit-learn-ийн урьдчилан боловсруулах модулийн LabelEncoder нь мөрийн шошгыг бүхэл тоо болгон кодлох.

```
# Assigning a label to our quality variable

label_quality = LabelEncoder()




# Now changing our dataframe to reflect our new label

df['quality'] = label_quality.fit_transform(df['quality'])
```

```
df.head(10)
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alcohol	quality
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	0
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8	0
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8	0
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8	0
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4	0
5	7.4	0.66	0.00	1.8	0.075	13.0	40.0	0.9978	3.51	0.56	9.4	0
6	7.9	0.60	0.06	1.6	0.069	15.0	59.0	0.9964	3.30	0.46	9.4	0
7	7.3	0.65	0.00	1.2	0.065	15.0	21.0	0.9946	3.39	0.47	10.0	1
8	7.8	0.58	0.02	2.0	0.073	9.0	18.0	0.9968	3.36	0.57	9.5	1
10	6.7	0.58	0.08	1.8	0.097	15.0	65.0	0.9959	3.28	0.54	9.2	0

өгөгдлийн багцыг онцлог (X) болон зорилтот хувьсагч (Y) болгон хуваж, дараа нь сургалт, туршилтын багц болгон хуваах замаар машин сурахад бэлтгэдэг.

```
Y = df.quality

X = df.drop('quality', axis=1)
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size =
0.2, random_state = 0)
```

```
#Feature Scaling

# from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X_train = sc.fit_transform(X_train)

X_test = sc.transform(X_test)
```

Python функц нь өгөгдсөн сургалтын өгөгдлийн багц дээр олон машин сургалтын загварыг сургаж, сургалтын нарийвчлалыг хэвлэдэг загваруудыг нэрлэсэн.

def models(X_train, Y_train):: Функцийн загварууд нь X_train (сургалтын өгөгдлийн онцлог) ба Y_train (сургалтын өгөгдлийн зорилтот хувьсагч) гэсэн хоёр параметрийг авдаг.

Загварын сургалт
Функцын дотор scikit-learn номын сангаас хэд хэдэн ангиллын загварыг бий болгож, сургасан:

K Nearest Neighbors (KNN)
Minkowski хэмжигдэхүүнийг ашиглан 5 хөрштэй KNeighborsClassifier ашигладаг p=2 (Евклидийн зайтай тэнцүү).
Загварыг .fit(X_train, Y_train) ашиглан сургадаг.

Support Vector Machine (SVM)
Шугаман цөмтэй SVC классыг ашигладаг.
Загварыг .fit(X_train, Y_train) ашиглан сургадаг.

Gaussian Naive Bayes
GaussianNB ангиллыг ашигладаг.
Загварыг .fit(X_train, Y_train) ашиглан сургадаг.

Decision Tree
Хуваахдаа энтропийн шалгуур бүхий DecisionTreeClassifier ашигладаг.
Загварыг .fit(X_train, Y_train) ашиглан сургадаг.

DecisionTree
10 мод болон энтропийн шалгуур бүхий RandomForestClassifier ашигладаг.
Загварыг .fit(X_train, Y_train) ашиглан сургадаг.

```
#Create a function within many Machine Learning Models
```

```
def models(X_train,Y_train):


  #Using KNeighborsClassifier Method of neighbors class to use Nearest
Neighbor algorithm

  from sklearn.neighbors import KNeighborsClassifier

  knn = KNeighborsClassifier(n_neighbors = 5, metric = 'minkowski', p =
2)

  knn.fit(X_train, Y_train)


  #Using SVC method of svm class to use Support Vector Machine
Algorithm

  from sklearn.svm import SVC

  svc_lin = SVC(kernel = 'linear', random_state = 0)

  svc_lin.fit(X_train, Y_train)


  #Using GaussianNB method of naïve_bayes class to use Naïve Bayes
Algorithm

  from sklearn.naive_bayes import GaussianNB

  gauss = GaussianNB()

  gauss.fit(X_train, Y_train)


  #Using DecisionTreeClassifier of tree class to use Decision Tree
Algorithm

  from sklearn.tree import DecisionTreeClassifier

  tree = DecisionTreeClassifier(criterion = 'entropy', random_state =
0)

  tree.fit(X_train, Y_train)


  #Using RandomForestClassifier method of ensemble class to use Random
Forest Classification algorithm

  from sklearn.ensemble import RandomForestClassifier
```

```
  forest = RandomForestClassifier(n_estimators = 10, criterion =
'entropy', random_state = 0)

  forest.fit(X_train, Y_train)



  #print model accuracy on the training data.

  print('[0]K Nearest Neighbor Training Accuracy:', knn.score(X_train,
Y_train))

  print('[1]Support Vector Machine Training Accuracy:',
svc_lin.score(X_train, Y_train))

  print('[2]Gaussian Naive Bayes Training Accuracy:',
gauss.score(X_train, Y_train))

  print('[3]Decision Tree Classifier Training Accuracy:',
tree.score(X_train, Y_train))

  print('[4]Random Forest Classifier Training Accuracy:',
forest.score(X_train, Y_train))




  return knn, svc_lin, gauss, tree, forest
```

Үр дүн нь харагдаж байна.

```
#Get and train all of the models

model = models(X_train,Y_train)
```

```
[0]K Nearest Neighbor Training Accuracy: 0.9310710498409331
[1]Support Vector Machine Training Accuracy: 0.911983032873807
[2]Gaussian Naive Bayes Training Accuracy: 0.8769883351007424
[3]Decision Tree Classifier Training Accuracy: 1.0
[4]Random Forest Classifier Training Accuracy: 0.9936373276776246
```