```
# Import Required Libraries
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

# Read in white wine data
white = pd.read_csv("/content/winequality-white.csv", sep =';')

# Read in red wine data
red = pd.read_csv("/content/winequality-red.csv", sep =';')


# First rows of `red`
red.head()
```

	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

```
# Last rows of `white`
white.tail()
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	su
4893	6.2	0.21	0.29	1.6	0.039	24.0	92.0	0.99114	3.27	
4894	6.6	0.32	0.36	8.0	0.047	57.0	168.0	0.99490	3.15	
4895	6.5	0.24	0.19	1.2	0.041	30.0	111.0	0.99254	2.99	
4896	5.5	0.29	0.30	1.1	0.022	20.0	110.0	0.98869	3.34	
4897	6.0	0.21	0.38	0.8	0.020	22.0	98.0	0.98941	3.26	

```
# Take a sample of five rows of `red`
red.sample(5)
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	su
1395	8.6	0.685	0.10	1.6	0.092	3.0	12.0	0.99745	3.31	
718	8.4	0.560	0.04	2.0	0.082	10.0	22.0	0.99760	3.22	
750	8.3	0.650	0.10	2.9	0.089	17.0	40.0	0.99803	3.29	
1124	6.5	0.580	0.00	2.2	0.096	3.0	13.0	0.99557	3.62	
1238	6.0	0.500	0.00	1.4	0.057	15.0	26.0	0.99448	3.36	

```
# Describe `white`
white.describe()
```

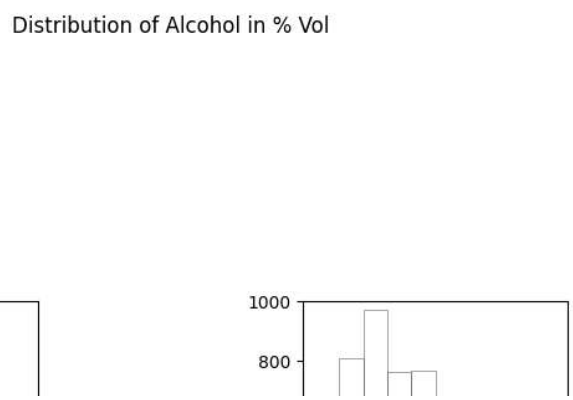
	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	s di
count	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.000000	4898.0
mean	6.854788	0.278241	0.334192	6.391415	0.045772	35.308085	138.3
std	0.843868	0.100795	0.121020	5.072058	0.021848	17.007137	42.4
min	3.800000	0.080000	0.000000	0.600000	0.009000	2.000000	9.0

```
# Double check for null values in `red`
pd.isnull(red)
```

	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	s
0	False	False	False	False	False	False	False	False	False	
1	False	False	False	False	False	False	False	False	False	
2	False	False	False	False	False	False	False	False	False	
3	False	False	False	False	False	False	False	False	False	
4	False	False	False	False	False	False	False	False	False	
...	...	...	...	...	...	...	...	...	...	
1594	False	False	False	False	False	False	False	False	False	
1595	False	False	False	False	False	False	False	False	False	
1596	False	False	False	False	False	False	False	False	False	
1597	False	False	False	False	False	False	False	False	False	
1598	False	False	False	False	False	False	False	False	False	

1599 rows × 12 columns

```
# Create Histogram
fig, ax = plt.subplots(1, 2)

ax[0].hist(red.alcohol, 10, facecolor ='red',
            alpha = 0.5, label ="Red wine")

ax[1].hist(white.alcohol, 10, facecolor ='white',
        ec ="black", lw = 0.5, alpha = 0.5,
        label ="White wine")

fig.subplots_adjust(left = 0, right = 1, bottom = 0,
            top = 0.5, hspace = 0.05, wspace = 1)

ax[0].set_ylim([0, 1000])
ax[0].set_xlabel("Alcohol in % Vol")
ax[0].set_ylabel("Frequency")
ax[1].set_ylim([0, 1000])
ax[1].set_xlabel("Alcohol in % Vol")
ax[1].set_ylabel("Frequency")

fig.suptitle("Distribution of Alcohol in % Vol")
plt.show()
```




```
# Add `type` column to `red` with price one
red['type'] = 1

# Add `type` column to `white` with price zero
white['type'] = 0

# Append `white` to `red`
wines = red.append(white, ignore_index = True)

# Import `train_test_split` from `sklearn.model_selection`
from sklearn.model_selection import train_test_split
X = wines.iloc[:, 0:11]
y = np.ravel(wines.type)

# Splitting the data set for training and validating
X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size = 0.34, random_state = 45)
```

```
<ipython-input-10-42380618fe8a>:8: FutureWarning: The frame.append method is deprecated and will be removed from pandas in a future vers
  wines = red.append(white, ignore_index = True)
```

```
# Import `Sequential` from `keras.models`
from keras.models import Sequential

# Import `Dense` from `keras.layers`
from keras.layers import Dense

# Initialize the constructor
model = Sequential()

# Add an input layer
model.add(Dense(12, activation ='relu', input_shape =(11, )))

# Add one hidden layer
model.add(Dense(9, activation ='relu'))

# Add an output layer
model.add(Dense(1, activation ='sigmoid'))

# Model output shape
model.output_shape

# Model summary
model.summary()

# Model config
model.get_config()

# List all weight tensors
model.get_weights()
model.compile(loss ='binary_crossentropy',
optimizer ='adam', metrics =['accuracy'])
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
```

```
 dense (Dense)               (None, 12)                144

 dense_1 (Dense)             (None, 9)                 117

 dense_2 (Dense)             (None, 1)                 10

=================================================================
Total params: 271 (1.06 KB)
Trainable params: 271 (1.06 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

```
# Training Model
model.fit(X_train, y_train, epochs = 3,
    batch_size = 1, verbose = 1)

# Predicting the Value
y_pred = model.predict(X_test)
print(y_pred)
```

```
Epoch 1/3
4288/4288 [==============================] - 11s 2ms/step - loss: 0.3876 - accuracy: 0.9076
Epoch 2/3
4288/4288 [==============================] - 9s 2ms/step - loss: 0.1840 - accuracy: 0.9363
Epoch 3/3
4288/4288 [==============================] - 9s 2ms/step - loss: 0.1577 - accuracy: 0.9471
70/70 [==============================] - 0s 2ms/step
[[0.02697089]
 [0.03689588]
 [0.00860741]
 ...
 [0.07302065]
 [0.02354082]
 [0.00142525]]
```