In [1]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import accuracy_score

from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
```

In [2]:

```
#A IS THE DATEFRAME OF WATER QUALITY DATASET
A=pd.read_csv("C:/Users/dell/Desktop/NANDU/PG/SEM2/PROJECT/WATERQUALITY.csv")
A.head(5)
```

Out[2]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	NaN	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	NaN	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	NaN	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

In [3]:

```
A.info()
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 3276 entries, 0 to 3275
Data columns (total 10 columns):
 #   Column           Non-Null Count  Dtype  
---  ------           --------------  -----  
 0   ph               2785 non-null   float64
 1   Hardness         3276 non-null   float64
 2   Solids           3276 non-null   float64
 3   Chloramines      3276 non-null   float64
 4   Sulfate          2495 non-null   float64
 5   Conductivity     3276 non-null   float64
 6   Organic_carbon   3276 non-null   float64
 7   Trihalomethanes  3114 non-null   float64
 8   Turbidity        3276 non-null   float64
 9   Potability       3276 non-null   int64  
dtypes: float64(9), int64(1)
memory usage: 256.1 KB
```

In [4]:

```
#size of A
A.shape
```

Out[4]:

```
(3276, 10)
```

In [ ]:

In [5]:

```
A.isnull().sum()
```

Out[5]:

```
ph                 491
Hardness             0
Solids               0
Chloramines          0
Sulfate            781
Conductivity         0
Organic_carbon       0
Trihalomethanes    162
Turbidity            0
Potability           0
dtype: int64
```

In [6]:

```
A['ph'] = A['ph'].fillna(A.groupby(['Potability'])['ph'].transform('mean'))
A['Sulfate'] = A['Sulfate'].fillna(A.groupby(['Potability'])['Sulfate'].transform('mean'))
A['Trihalomethanes'] = A['Trihalomethanes'].fillna(A.groupby(['Potability'])['Trihalomethanes'].transform('mean'))
```

In [7]:

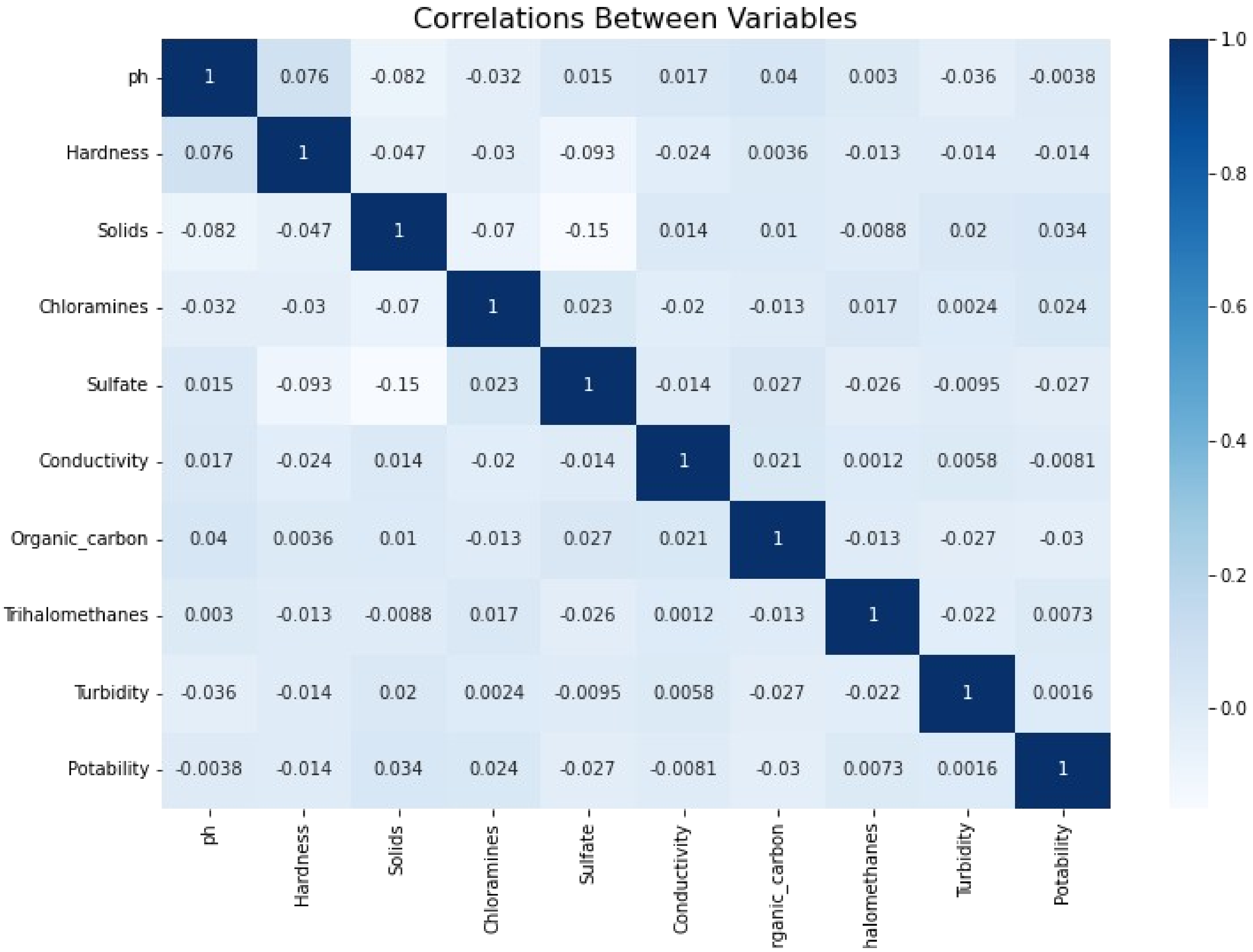
```
A.head(5)
```

Out[7]:

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carbon	Trihalomethanes	Turbidity	Potability
0	7.085378	204.890455	20791.318981	7.300212	368.516441	564.308654	10.379783	86.990970	2.963135	0
1	3.716080	129.422921	18630.057858	6.635246	334.564290	592.885359	15.180013	56.329076	4.500656	0
2	8.099124	224.236259	19909.541732	9.275884	334.564290	418.606213	16.868637	66.420093	3.055934	0
3	8.316766	214.373394	22018.417441	8.059332	356.886136	363.266516	18.436524	100.341674	4.628771	0
4	9.092223	181.101509	17978.986339	6.546600	310.135738	398.410813	11.558279	31.997993	4.075075	0

In [8]:

```
#CORRELATION BETWEEN VARIABLES
plt.figure(figsize=(12, 8))
sns.heatmap(A.corr(), annot=True, cmap="Blues")
plt.title("Correlations Between Variables", size=16)
plt.show()
```



Correlations Between Variables

	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	rganic_carbon	halomethanes	Turbidity	Potability
ph	1	0.076	-0.082	-0.032	0.015	0.017	0.04	0.003	-0.036	-0.0038
Hardness	0.076	1	-0.047	-0.03	-0.093	-0.024	0.0036	-0.013	-0.014	-0.014
Solids	-0.082	-0.047	1	-0.07	-0.15	0.014	0.01	-0.0088	0.02	0.034
Chloramines	-0.032	-0.03	-0.07	1	0.023	-0.02	-0.013	0.017	0.0024	0.024
Sulfate	0.015	-0.093	-0.15	0.023	1	-0.014	0.027	-0.026	-0.0095	-0.027
Conductivity	0.017	-0.024	0.014	-0.02	-0.014	1	0.021	0.0012	0.0058	-0.0081
Organic_carbon	0.04	0.0036	0.01	-0.013	0.027	0.021	1	-0.013	-0.027	-0.03
Trihalomethanes	0.003	-0.013	-0.0088	0.017	-0.026	0.0012	-0.013	1	-0.022	0.0073
Turbidity	-0.036	-0.014	0.02	0.0024	-0.0095	0.0058	-0.027	-0.022	1	0.0016
Potability	-0.0038	-0.014	0.034	0.024	-0.027	-0.0081	-0.03	0.0073	0.0016	1

In [9]:

```
#SPLITTING X AN Y
X = A.drop("Potability", axis=1)
y = A["Potability"]
```

In [10]:

```
#TEST AND TRAIN DATAEST SPLIT
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

In [11]:

```
#LOGISTIC REGRESSION MODEL
LR=LogisticRegression(random_state=42).fit(X_train, y_train)
y_pred1 = LR.predict(X_test)
score_LR = accuracy_score(y_test, y_pred1)
print(score_LR)
```

```
0.6280487804878049
```

In [12]:

```
#DESICION TREE CLASSIFIER MODEL
DT=DecisionTreeClassifier(random_state=42).fit(X_train, y_train)
y_pred2=DT.predict(X_test)
score_DT=accuracy_score(y_test,y_pred2)
print(score_DT)
```

```
0.7469512195121951
```

In [13]:

```
#KNN model
KNN=KNeighborsClassifier(n_neighbors=2).fit(X_train,y_train)
y_pred3=KNN.predict(X_test)
score_KNN=accuracy_score(y_test,y_pred3)
print(score_KNN)
```

```
0.5899390243902439
```