**CLASSICAL MACHINE LEARNING**

**Table of Contents**

[Data 2](#_Toc108891671)

[Clustering 2](#_Toc108891672)

[Method 1: K means clustering 2](#_Toc108891673)

[Method 2: DBscan cluster 2](#_Toc108891674)

[Method 3: Hirerechcal cluster (Agglomerative Clustering) 3](#_Toc108891675)

[Evaluation 3](#_Toc108891676)

[Classification 4](#_Toc108891677)

[Methods 4](#_Toc108891678)

[Model 1: Random forest classifier 4](#_Toc108891679)

[Model 2: Logistic regression 4](#_Toc108891680)

[Evaluation 4](#_Toc108891681)

[Cross-validation 5](#_Toc108891682)

[Conclusion 6](#_Toc108891683)

[Discussion 6](#_Toc108891684)

[Clustering 6](#_Toc108891685)

[Conclusion 6](#_Toc108891686)

[Discussion 7](#_Toc108891687)

[Classification 7](#_Toc108891688)

[Conclusion 7](#_Toc108891689)

[Appendix 8](#_Toc108891690)

[Reference list 13](#_Toc108891691)

# Data

A dataset is the set or assemblage of the staistcs. This kind of set is normally represented in a tubular sequence. In every list that is describing a circumstance variable. Each of the lists coincides with a given number set of data. That is the portion of staistcs management. Staistcs sets which is describing the values for individual variables for the nameless abundance. Here, In the dataset “***WINE.CSV***” different types of column has present and the quality has selected as the target column. In the dataset various types of column has present as “***fixed\_acidity”,”density”,”PH”*** and others. The statistics set which is consisting of statistics of one or more members analogous in each row.

# Clustering

## Method 1: K means clustering

The K means conclusion that is taking the outputs criterion K from the user and dissolution the dataset which is containing N equipment into the K cluster so that the outcomes analogy in between the statistics objects. Inside the groups which are very high but the affinity of the statistics objects with the statistics from the outside of the cluster becomes very low. The affinity of the cluster is determined by the appreciation of the mean value of the cluster. That is the kind of square failure conclusion (Alam *et al.* 2020). At the initial start randomly the k equipment from the dataset is choosing in which exclusive of the equipment that can be representing a cluster mean. The rest of the statics equipment can be defined as the convenient cluster that is based on the area from the mean cluster. The new median of the individual of the cluster is then that is calculating with the help of added statistics objects.

## Method 2: DBscan cluster

DBscan is a famous density that is based on statistics clustering conclusion. To the cluster statistics points, this kind of conclusion which is separating the high-frequency locality of the statistics from the low-frequency region. Unlike the k means conclusion, the good thing with this kind of conclusion is that it is not an obligation to provide the number of clusters that are required for the proceeding. DBscan's conclusion is based on measuring distance and the least possible of points (Burns *et al.* 2018). The crucial equity of this conclusion is helping to track down the aberration as the points in the low-frequency locality.

## Method 3: Hirerechcalcluster (Agglomerative Clustering)

That is looking at the adequate methods for the K means clustering which is called hierarchical clustering. In the analogy to the K means or the K mode hierarchical clustering that is having an adequate elemental conclusion. Hierarchical clustering that is using the disruptive approach, where k means that it uses a consolidation of centroid and the distance from the cluster. Here we are using python to analyze the Hierarchical clustering design. It is having 200 mall consumer statistics in the dataset (Pilnenskiy *et al.* 2020). Each of the consumers who is having a customer ID and it is spending score are all combined in the dataset. The extent computing for the individual clients is to spend the scores which are based on considerable criteria as like its compensation and the per week number of time to appointment the mall and the amount of the money that is spending in annually.

# Evaluation

Machine learning is the region of study which gives computers the capacity to learn without being absolutely programmed. Automation learning is one of the most appealing automation that one can become beyond. As that is conspicuous from the name that is giving the computer which is making it more analogous to the human. Automation learning is ardent and is being used today. The organization is using automation learning to improve commercial decisions, expand capacity, detect disease, and ambiguous weather, and do many more things.

Without the aggressive growth of automation, it cannot need the better components which is to understand the staistcs which is recently having. To achieve this kind of ambition it needs to build intelligent automation. But most of the time, the hand-working acumen in it is ambitious (Raschka *et al.* 2020). If the automation learning from the input then that cannot be a hard job for us. The typical appliance that is adding the web click statistics for the better **UX**, medical that is the best archives machine in health maintenance. It is having some works which cannot be automatic as the computers which are not used to create that way. For example, it is including self-determining driving and recognition work from the disordered statistics.

# Classification

## Methods

### Model 1: Random forest classifier

The random forest or the random arrangements forest is the managed automation learning conclusion that is used for the allocation.Regression, and the other work which is used in the arrangements tree (Zahorodko *et al.* 2021). The random forest allocation is developing a set of arrangements tree from the randomly elevated subset of the practice arrangements. That is basically an arrangement of arrangements tree from the selection of randomly, a subgroup of the practice arrangements, and then that is collecting the votes from the adequate decision tree which is to decide the final forecast. Random forests which are having an assortment of the appliance as like the endorsement image allocation and the selection of choice. That can be used to classify the candidates. To identify the deceitful enterprise and the forecast contamination. The advantages of random forests are considered as decidedly authentic and booming the arrangements in the reason of the number of decision trees to the coordinated in the process. That is not enduring from the issue of overnighting. The important reason is that it is taking the balance in all of the forecasts.

### Model 2: Logistic regression

A logistic regression design “**predicts P(Y=1) as the function of X”**. That is one of the elemental automation learning conclusions which can be used for different allocation issues as spam discloser. As the aggregate of the available staistcs the power of computing strength and the number which is having conclusion advancement which continues to the rising, so it can be crucial of the data science and the automation learning. Allocation is between the most crucial region of automation learning and the logistic regression which is one of its essential arrangements (Zahorodko *et al.* 2021).

## Evaluation

The nature of the reliant variable comprehends regression and the allocation issue. The regression issue is having the continued and commonly absolute outputs. An example is when it can be guessed the salary as the activity of involvement and the level of education. On the other hand, the allocation issue is having disconnected and definite outputs which are known as classes. For example, it is predicting if a worker is going which is to be approved or not. Here total dataset has splitted test and train data with 80:20 proportion. Depending on the splitting data the prediction result has analyzed.

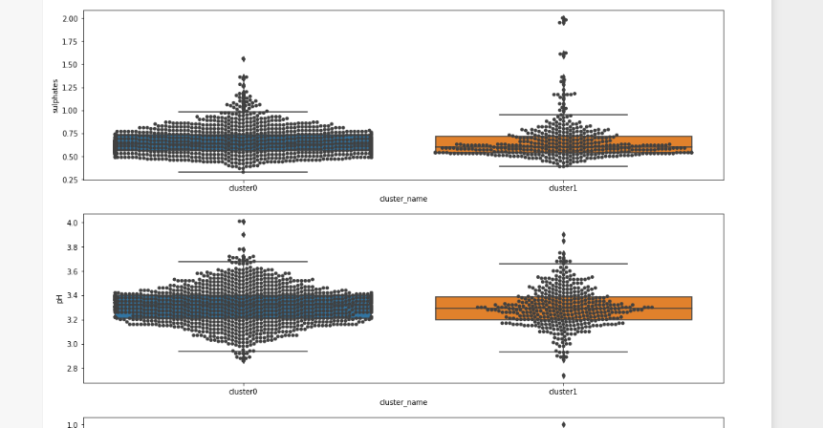
## Cross-validation

This kind of clustering method that is classifying the details into multiple kinds of groups which is based on the aspect and analogy of the statistics. It's the statistics investigation which is to specify the number of clusters that has to generate for the clustering approach. In the partitioning approach when the database which is containing different kinds of equipment. Then the partitioning approach is to construct the user described (Lippeveld *et al.* 2020). Dissolution of the staistcs in which each of the dissolutions can be representing a cluster and specific field. It is having any kind of conclusion which is coming under the dissolution approach in some of the attractive ones.

# Conclusion

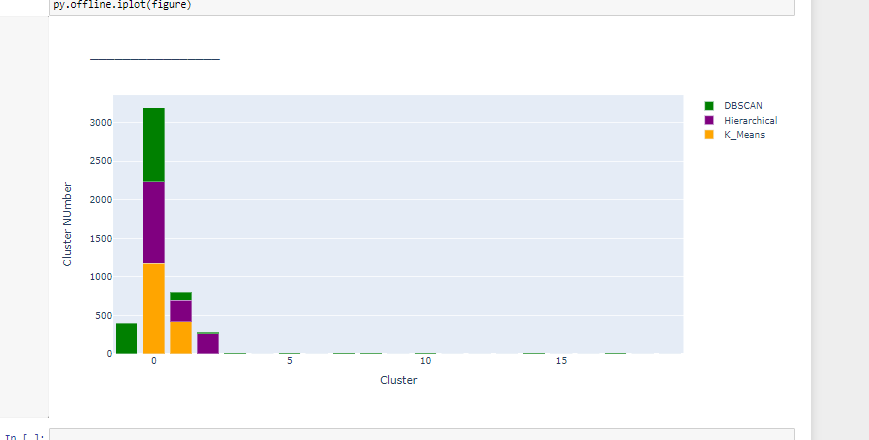
# Discussion

## Clustering



**Figure 1: Cluster result**

(Source: Develop in Jupyter Notebook)



**Figure 2: Cluster result Analysis**

(Source: Developed in Jupyter Notebook)

## Discussion

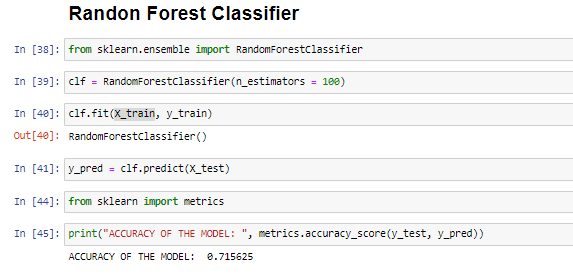
Here, cluster result and analysis of that result has been provided. It has been identified that the cluster numbers reached till 3000. This process of clustering or K-means clustering has been done, where the affinity of the cluster is determined by the appreciation of the mean value of the cluster here.

## Conclusion

* Based on above discussion, here three types of clustering analysis has performed on the wine dataset.
* ***K-means*** clustering has performed, here K defines the number of predefied cluster created the process, s if K=2, there will be two clusters, and for K=3, there will be three clusters.
* Hierarchical cluster has perfomed to define the ***Agglomerative clustering*** is the most well-known kind of various leveled clustering used to gather objects in bunches based on their comparability.
* DBSCAN cluster has perfomed for getting the neighbourhood of the each point of the cluster that define the minimum number of points.

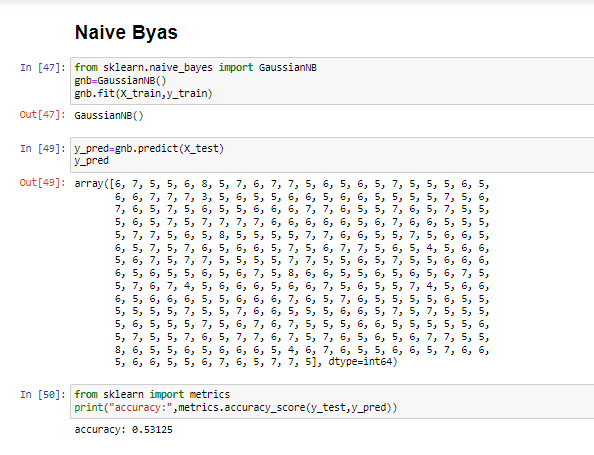
# Discussion

## Classification



**Figure 3: Result of Random Forest Classifier**

(Source: Developed in Jupyter Notebook)



**Figure 4: Result of Naïve Byas Classification**

(Source: Developed in Jupyter Notebook)

## Discussion

The result of Random forest classifier and Naive byas classification has been performed here. In the case of random forest, the accuracy score is 0.7156. On the other hand, the accuracy score of Naive byas classification is 0.5312. Comparing the result of these two classifiers, random forest is better than the Naive byas.

## Conclusion

* Based on above discussion here three types of classification has performed “***Random forestclassification”, “Naïve Byas”,”Logistic Regression***”.
* From the logistic regression get the ROC curve as actual value and predictive value.
* Naïve byes classification depending on the target column prediction result has produced.
* Random forest classification has performed for get the highest level of accuracy score.

# Appendix

|  |
| --- |
| import pandas as pd  import numpy as np  import seaborn as sns  import matplotlib.pyplot as plt  import matplotlib.gridspec as gridspec  %matplotlib inline  import warnings  warnings.simplefilter("ignore")  df=pd.read\_csv('winequality-red.csv')  plt.figure(figsize=[9, 9])  sns.heatmap(df.corr(), xticklabels=df.columns[:-1], yticklabels=df.columns[:-1],  square=True, cmap="Spectral\_r", center=0);  defdrawing\_two\_barplots(column, ylims):  fig = plt.figure(figsize=(14, 12))  gs = gridspec.GridSpec(2, 2)  ax0 = fig.add\_subplot(gs[0, :])  ax1 = fig.add\_subplot(gs[1, 0])  ax2 = fig.add\_subplot(gs[1, 1])  sns.distplot(df[df.columns[column]], kde=False, ax=ax0)  sns.barplot("quality", df.columns[column], data=df, ax=ax1)  sns.barplot("bin\_quality", df.columns[column], data=df, ax=ax2)  ax1.set\_ylim(ylims[0], ylims[1])  ax2.set\_ylim(ylims[0], ylims[1])  ax1.set\_yticks(np.linspace(ylims[0], ylims[1], 11))  ax2.set\_yticks(np.linspace(ylims[0], ylims[1], 11))  ax1.yaxis.grid()  ax2.yaxis.grid()  ax1.set\_axisbelow(True)  ax2.set\_axisbelow(True)  drawing\_two\_barplots(0, [0, 10])  from sklearn.linear\_model import LogisticRegression  from sklearn.model\_selection import train\_test\_split  X\_train,X\_test,y\_train,y\_test=train\_test\_split(df.drop('quality',axis=1),df['quality'],test\_size=0.20,random\_state=100)  print("no of train set is:\n",X\_train)  print("no of test set is:\n",y\_test)  logmodel=LogisticRegression()  logmodel.fit(X\_train,y\_train)  logmodel.score(X\_train,y\_train)  from sklearn.ensemble import RandomForestClassifier  from sklearn import metrics  print("ACCURACY OF THE MODEL: ", metrics.accuracy\_score(y\_test, y\_pred))  X = df.iloc[:,:]  X.head()  from sklearn.cluster import KMeans  wcss = []  for i in range(1,11):  kmeans = KMeans(n\_clusters = i, init = 'k-means++', random\_state=42)  kmeans.fit(X)  wcss.append(kmeans.inertia\_)  plt.plot(range(1,11),wcss, scalex=True)  plt.title('The Elbow Method')  plt.xlabel('Number of Clusters')  plt.ylabel('WCSS')  plt.show()  plt.scatter(X[y==0,0], X[y==0,1], c='red', s=50)  plt.scatter(X[y==1,0], X[y==1,1], c='blue', s=50)  plt.scatter(X[y==2,0], X[y==2,1], c='green', s=50)  plt.title('KMeans Clustering')  plt.xlabel('Alcohol')  plt.ylabel('Malic\_Acid')  plt.show()  import plotly.express as px  clusters=pd.DataFrame(X,columns=df.columns)  clusters['label']=kmeans.labels\_  pie=clusters.groupby('label').size().reset\_index()  pie.columns=['label','value']  px.pie(pie,values='value',names='label',color=['blue','red','green'])  import scipy.cluster.hierarchy as sch  dendrogram = sch.dendrogram(sch.linkage(X,method='ward'))  plt.title('Dendrogram')  plt.xlabel('Wines')  plt.ylabel('Euclidean Distance')  plt.show()  kmeans = KMeans(  n\_clusters=3, init="k-means++",  n\_init=10,  tol=1e-04, random\_state=42  )  kmeans.fit(X)  clusters=pd.DataFrame(X,columns=df.columns)  clusters['label']=kmeans.labels\_  polar=clusters.groupby("label").mean().reset\_index()  polar=pd.melt(polar,id\_vars=["label"])  fig4 = px.line\_polar(polar, r="value", theta="variable", color="label", line\_close=True,height=800,width=1400)  fig4.show()  defdoKmeans(X, nclust=2):  model = KMeans(nclust)  model.fit(X)  clust\_labels = model.predict(X)  cent = model.cluster\_centers\_  return (clust\_labels, cent)  clust\_labels, cent = doKmeans(df, 2)  kmeans = pd.DataFrame(clust\_labels)  df.insert((df.shape[1]),'kmeans',kmeans)  from sklearn.metrics import accuracy\_score  print("Accuracy score for kmeans algorithm is",accuracy\_score(clust\_labels,kmeans ))  def condition(x):  if x == 0:  x = "cluster0"  elif x == 1:  x = "cluster1"  return x  df["cluster\_name"] = df["kmeans"].apply(lambda x: condition(x))  df[df['cluster\_name'] == 'cluster0'].hist(figsize = (15,10))  plt.show()  fig, (ax2,ax3,ax4,ax5) = plt.subplots(nrows = 4)  fig.set\_figwidth(16)  fig.set\_figheight(20)  # Cluster analysis of wine  x = sns.boxplot(x="cluster\_name", y='alcohol', data=df, ax= ax2)  x = sns.swarmplot(x="cluster\_name", y='alcohol', data=df,color=".25",ax= ax2)  # Cluster analysis of wine  x = sns.boxplot(x="cluster\_name", y='sulphates', data=df, ax= ax3)  x = sns.swarmplot(x="cluster\_name", y='sulphates', data=df,color=".25",ax= ax3)  x = sns.boxplot(x="cluster\_name", y='pH', data=df, ax= ax4)  x = sns.swarmplot(x="cluster\_name", y='pH', data=df,color=".25",ax= ax4)  x = sns.boxplot(x="cluster\_name", y='citric acid', data=df, ax= ax5)  x = sns.swarmplot(x="cluster\_name", y='citric acid', data=df,color=".25",ax= ax5)  import scipy.cluster.hierarchy as sch  from sklearn.cluster import AgglomerativeClustering  plt.figure(figsize=(8, 6))  sns.scatterplot(x='alcohol', y='sulphates', data=df1, color='green')  sns.scatterplot(x='alcohol', y='sulphates', data=df2, color='red')  sns.scatterplot(x='alcohol', y='sulphates', data=df3, color='blue')  plt.xlabel('Width')  plt.ylabel('Length')  plt.show()  df1 = df[df['DBSCAN\_Cluster']==0]  df2 = df[df['DBSCAN\_Cluster']==1]  df3 = df[df['DBSCAN\_Cluster']==-1]  sb = df[['alcohol', 'sulphates', 'kmeans', 'Hierarchical\_Cluster', 'DBSCAN\_Cluster']]  sb.head()  data1 = pd.DataFrame({  'KM\_Cluster' : km.index,  'K\_Means' : km.values  }, columns=['KM\_Cluster', 'K\_Means'])  data1.set\_index('KM\_Cluster', inplace=True)  data1  data2 = pd.DataFrame({  'HC\_Cluster' : hc.index,  'Hierarchial' : hc.values  }, columns=['HC\_Cluster', 'Hierarchial'])  data2.set\_index('HC\_Cluster', inplace=True)  data2  data3 = pd.DataFrame({  'DB\_Cluster': db.index,  'DBSCAN' : db.values  }, columns=['DB\_Cluster', 'DBSCAN'])  data3.set\_index('DB\_Cluster', inplace=True)  data3  import plotly.graph\_objs as go  import pandas as pd  import numpy as np  import matplotlib.pyplot as plt  import seaborn as sns  import plotly as py  trace1 = go.Bar(  x=result.index,  y=result['K\_Means'],  marker\_color='orange',  name='K\_Means'  )  trace2 = go.Bar(  x=result.index,  y=result['Hierarchial'],  marker\_color='purple',  name='Hierarchical'  )  trace3 = go.Bar(  x=result.index,  y=result['DBSCAN'],  marker\_color='green',  name='DBSCAN'  )  data = [trace1, trace2, trace3]  layout = go.Layout(  barmode='stack',  title=" \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_",  xaxis={  'title':"Cluster",  },  yaxis={  'title':"ClusterNUmber",  }  )  figure=go.Figure(data=data,layout=layout)  py.offline.iplot(figure) |

# Reference list

Al-Sharo, Y.M., Abu-Jassar, A.T., Sotnik, S. and Lyashenko, V., 2021. Neural Networks As A Tool For Pattern Recognition of Fasteners. *studies*, *4*(11), p.13.

Chawla, A., Singh, A., Lamba, A., Gangwani, N. and Soni, U., 2019. Demand forecasting using artificial neural networks—a case study of American retail corporation. In *Applications of artificial intelligence techniques in engineering* (pp. 79-89). Springer, Singapore.

Choudhary, A., Lindner, J.F., Holliday, E.G., Miller, S.T., Sinha, S. and Ditto, W.L., 2020. Physics-enhanced neural networks learn order and chaos. *Physical Review E*, *101*(6), p.062207.

Grattarola, D. and Alippi, C., 2021. Graph neural networks in TensorFlow and keras with spektral [application notes]. *IEEE Computational Intelligence Magazine*, *16*(1), pp.99-106.

Haghighat, E. and Juanes, R., 2021. Sciann: A keras/tensorflow wrapper for scientific computations and physics-informed deep learning using artificial neural networks. *Computer Methods in Applied Mechanics and Engineering*, *373*, p.113552.

Haque, M.R., Islam, M.M., Iqbal, H., Reza, M.S. and Hasan, M.K., 2018, February. Performance evaluation of random forests and artificial neural networks for the classification of liver disorder. In *2018 international conference on computer, communication, chemical, material and electronic engineering (IC4ME2)* (pp. 1-5). IEEE.

Hegde, R.S., 2019. Photonics inverse design: pairing deep neural networks with evolutionary algorithms. *IEEE Journal of Selected Topics in Quantum Electronics*, *26*(1), pp.1-8.

Janani, S., Thenmozhi, R. and Jayagopal, L.S., 2019. Theoretical investigations for the verification of shear centre and deflection of sigma section by back propagation neural network using Python. *Archives of Civil Engineering*, *65*(2).

Kannan, S., Dhiman, G., Natarajan, Y., Sharma, A., Mohanty, S.N., Soni, M., Easwaran, U., Ghorbani, H., Asheralieva, A. and Gheisari, M., 2021. Ubiquitous vehicular ad-hoc network computing using deep neural network with iot-based bat agents for traffic management. *Electronics*, *10*(7), p.785.

Koryagin, A., Khudorozkov, R. and Tsimfer, S., 2019. PyDEns: A python framework for solving differential equations with neural networks. *arXiv preprint arXiv:1909.11544*.

Lanovaz, M.J. and Bailey, J.D., 2022. Tutorial: Artificial neural networks to analyze single-case experimental designs. *Psychological Methods*.

Mähringer‐Kunz, A., Wagner, F., Hahn, F., Weinmann, A., Brodehl, S., Schotten, S., Hinrichs, J.B., Düber, C., Galle, P.R., Pinto dos Santos, D. and Kloeckner, R., 2020. Predicting survival after transarterial chemoembolization for hepatocellular carcinoma using a neural network: A Pilot Study. *Liver International*, *40*(3), pp.694-703.

Mishra, A., 2020. Artificial intelligence algorithms for the analysis of mechanical property of friction stir welded joints by using python programming. *Welding Technology Review*, *92*(6), pp.7-16.

Nusrat, I. and Jang, S.B., 2018. A comparison of regularization techniques in deep neural networks. *Symmetry*, *10*(11), p.648.

Rahmat, B. and Nugroho, B., 2019. Fuzzy and artificial neural networks-based intelligent control systems using Python. *Nusantara Science and Technology Proceedings*, pp.152-170.

Sajanraj, T.D. and Beena, M.V., 2018, April. Indian sign language numeral recognition using region of interest convolutional neural network. In *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)* (pp. 636-640). IEEE.

Shao, Y., Hellström, M., Mitev, P.D., Knijff, L. and Zhang, C., 2020. PiNN: A python library for building atomic neural networks of molecules and materials. *Journal of chemical information and modeling*, *60*(3), pp.1184-1193.

Siddique, F., Sakib, S. and Siddique, M.A.B., 2019, September. Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers. In *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)* (pp. 541-546). IEEE.

Srivastava, S., Sharma, L., Sharma, V., Kumar, A. and Darbari, H., 2019. Prediction of diabetes using artificial neural network approach. In *Engineering Vibration, Communication and Information Processing* (pp. 679-687). Springer, Singapore.

Viet, D.T., Phuong, V.V., Duong, M.Q. and Tran, Q.T., 2020. Models for short-term wind power forecasting based on improved artificial neural network using particle swarm optimization and genetic algorithms. *Energies*, *13*(11), p.2873.