INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH AND ANALYSIS

ISSN(print): 2643-9840, ISSN(online): 2643-9875

Volume 06 Issue 09 September 2023

DOI: 10.47191/ijmra/v6-i9-66, Impact Factor: 7.022

Page No. 4460-4466

Effect of Feature Selection on the Accuracy of Machine Learning Model

Asst. Professor Mohammad Salim Hamdard[1], Asst. Professor Hedayatullah Lodin[2]

[1,2] Faculty of Computer Science, Kabul University

ABSTRACT: In real life data science problems, it's almost rare that all the features in the dataset are useful for building a model. In machine learning, feature selection is the process of selecting a subset of relevant features or attributes for constructing a model. Removing irrelevant and redundant features and, selecting relevant features will improve the accuracy of a machine learning model. Furthermore, adding unnecessary variables to a model increases the overall complexity of the model. Our experiment indicates that the accuracy of a classification model is highly affected by the process of feature selection. We train three algorithms (K-Nearest Neighbors, Decision Tree, Multi-layer Perceptron) by selecting all the features and we got accuracies 49%, 84% and 71% accordingly. After doing some feature selection without any logical changes in models code the accuracy scores jumped to 82%, 86% and 78% accordingly which is quite impressive.

KEYWORDS: Machine Learning**,** Feature Selection, Accuracy, Dimensionality Reduction, Classification

1. INTRODUCTION

Feature selection is one of the core concept in machine learning which hugely impacts the performance of your model, especially in datasets with many input variables and a low variance [1]. The goal of feature selection process in machine learning is to find the best set of features that allows one to build optimized models that will have a great accuracy score [2]. The input data that we use to train our machine learning model have a huge influence on the model's performance. The increase in dimensionality of data can lead to big challenges in both supervised and unsupervised learning process. Training your model with redundant features reduces the model's overall capability and may also reduce model's accuracy. Moreover, adding extra variables to a model increases the overall complexity of the model [3]. Performing feature selection offers several benefits, it reduces overfitting, improves accuracy, and reduces training time. This paper will provide a great analysis of the importance of feature selection in constructing an optimized machine learning model [4].

1.1 RESEARCH QUESTION

In this research paper we will study the impact of feature selection process on the accuracy of a machine learning model by using three different machine learning algorithms (KNN, Decision Tree, Multi-layer Perceptron). We aim to answer the following research questions:

- Does the increase in input variables with a low variance decrease the overall capability of a machine learning model?
- Does the feature selection process improve the accuracy of machine learning model compared to using all features?

2. BACKGROUND

The field of machine learning is concerned with automated discoveries of regularities in data with use of computer algorithms. These regularities can then be used to take actions, such as classifying data into different categories or making predictions. As the data may be of different kinds, the machine learning algorithms that learn from these data may differ too [5]. The machine learning algorithms used for conducting this research are discussed below.

2.1 K-Nearest Neighbors

K-Nearest Neighbors is one of the simplest machine learning algorithms based on supervised learning technique. It is effective for classification as well as regression. However, it is more widely used for classification problems. It is a lazy learner algorithm because it does not learn from the training set immediately [6]. In case of KNN algorithm, a particular value of K is

fixed which helps us in classifying the unknown data point. When a new data point comes in KNN will predict its class by performing the following steps [7]:

Step 1: Store the training set.

Step 2: For each new unlabeled data

A. Calculate Euclidean distance with all training data points using the formula: $\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

B. Find the k- nearest neighbors.

C. Assign class containing the maximum number of nearest neighbors.

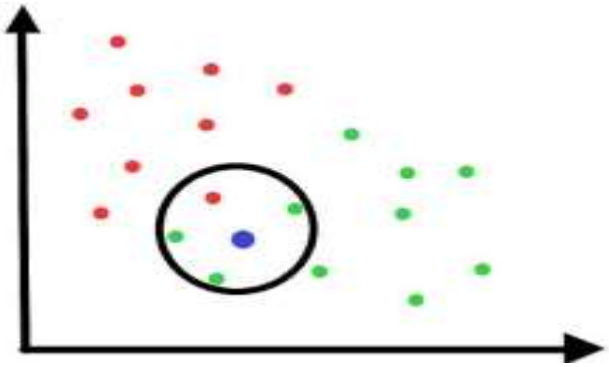

Figure 1: Working of KNN

2.2 DECISION TREE

A decision tree is a supervised learning algorithm used for both classification and regression problems. It takes the form of a tree with branches representing the potential answers to a given question [8]. In decision tree it is very important to select the right attribute or feature for splitting the dataset. Random selection is not a good idea it will generate bad result and low accuracy of prediction. In order to find the best splitting attribute, we need to consider feature selection measures like information gain. Information gain is based on entropy [9]. Entropy measures the extent of impurity or randomness in a dataset. If all the observations of subsets belong to one class, the entropy of that dataset would be 0. The entropy of the whole set of data can be calculated by using the following equation:

$$H(S) = -\sum_{i=1}^{N} Pi \log_2(Pi)$$

In the above equation, S represent set of all instances, N represent number of distinct class values and Pi represent event probability. Information gain indicates how much information a particular variable or feature gives us about the final outcome. It can be found out by subtracting the entropy of a particular attribute inside the data set from the entropy of the whole data set [8], [9].

$$\text{Gain}(A,S) = H(S) - \sum_{j=1}^{V} \frac{|Sj|}{|S|} . H(S_j) = H(S) - H(A,S)$$

2.3 ARTIFICIAL NEURAL NETWORK

The Artificial Neural Network (ANN) is a deep learning method that arose from the concept of working of the human brain. The workings of ANN are extremely similar to those of biological neural networks, although they are not identical [10]. There are three layers in the network architecture: the input layer, the hidden layer (more than one), and the output layer. Because of the numerous layers are sometimes referred to as the Multi-Layers Perceptron [10], [11].

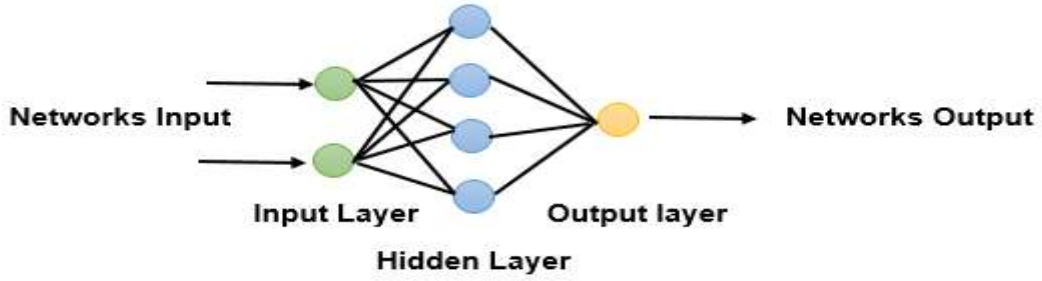

Figure 2: Architecture of Artificial Neural Network

3. FEATURE SELECTION METHODS

In general, feature selection algorithms are categorized into Supervised and Unsupervised feature selection [12].

- Supervised feature selection method uses the output label class for feature selection. Supervised feature selection methods can be further categorized as: Filter, Wrapper and Embedded approach.
- Unsupervised feature selection method refers to the method which does not need the output label class for feature selection.

3.1 Filter Method

In this method, we use correlation to check if the features are positively or negatively correlated to the output labels and drop or select features accordingly. These methods are faster and less computationally expensive than wrapper methods. When dealing with high-dimensional data, it is computationally cheaper to use filter methods. Eg: Information Gain, Chi-Square Test, Fisher's Score, etc [12], [13].

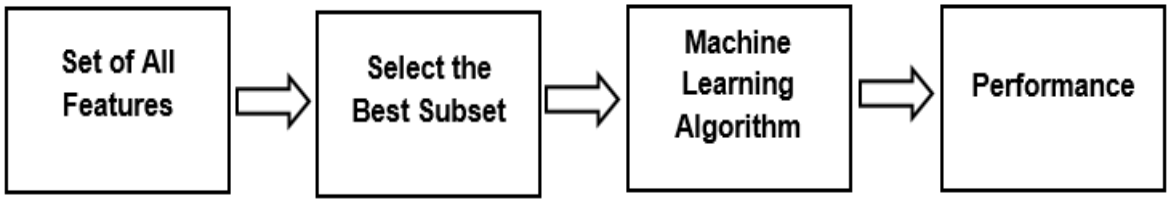

Figure 3: Filter Method flowchart

3.2 Wrapper Method

We split our data into subsets and train a model using this. Based on the output of the model, we add and subtract features and train the model again. It forms the subsets using a greedy approach and evaluates the accuracy of all the possible combinations of features. Eg: Forward Selection, Backwards Elimination, etc [13].

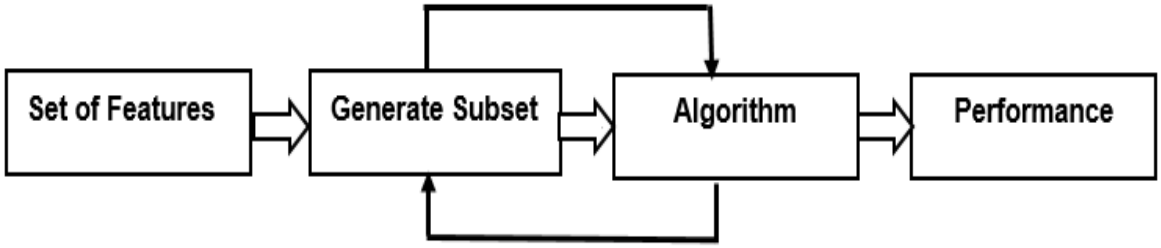

Figure 4: Wrapper Method flowchart

3.3 Embedded Method

This method combines the qualities of both the Filter and Wrapper method to create the best subset. This method takes care of the machine training iterative process while maintaining the computation cost to be minimum. Eg: Lasso and Ridge Regression [12], [14].

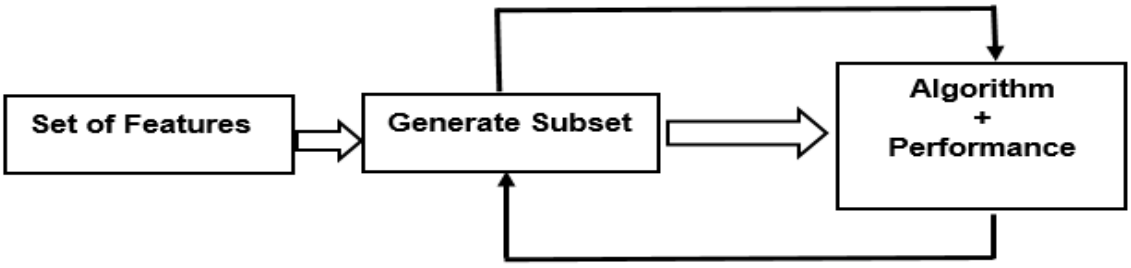

Figure 5: Embedded Method flowchart

4. METHODOLOGY

To study the impacts of feature selection on the accuracy of machine learning model we will use a dataset named mobile price prediction which is available on kaggle machine learning repository. There is total 2000 instances, 20 features and one output

variable which is mobile price range in dataset. In this project, based on the mobile specifications (battery power, 3G enabled, wifi, bluetooth, ram etc.) we are predicting price range of the mobile as output variable. We will train 3 classification algorithms (KNN, Decision Tree, Multi-layer Perceptron) to predict the output by selecting all the features. After training the algorithms by using all 20 features, we will now perform feature selection in order to find 5 best features which are highly correlated with output variable and having huge impacts on accuracy of our model. Finally, we will train the previously used 3 algorithms by using 5 best features without doing any logical changes in our models code, it clearly shows that the accuracy of a machine learning model is highly effected by the process of feature selection.

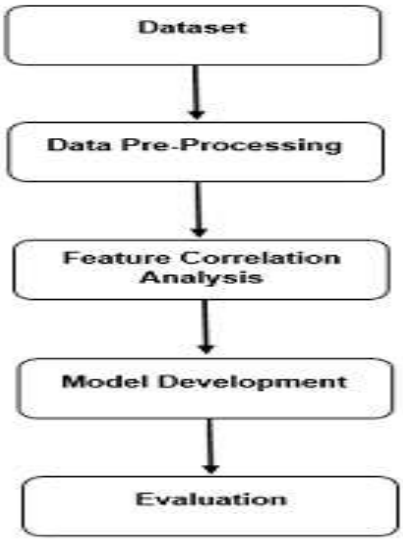

Figure 6: Flowchart of the Methodology

5. EXPERIMENTAL RESULT AND DISCUSSION

In order to select those features that have strongest relationship with the output variable, we use scikit-learn library it provides the SelectKBest class that can be used with a suite of different statistical tests to select a specific number of features. This technique belongs to filter method of feature selection as it uses statistical tools to evaluate the relationship of each input variable and the output variable and then drop the irrelevant features. But before starting practical work let check the first 20 rows of our dataset.

0	842	0	2.2	0	1	0	7	0.6	188	2	2	20	756	2549	9	7	19	0	0	1	1
1	1021	1	0.5	1	0	1	53	0.7	136	3	6	905	1988	2631	17	3	7	1	1	0	2
2	563	1	0.5	1	2	1	41	0.9	145	5	6	1263	1716	2603	11	2	9	1	1	0	2
3	615	1	2.5	0	0	0	10	0.8	131	6	9	1216	1786	2769	16	8	11	1	0	0	2
4	1821	1	1.2	0	13	1	44	0.6	141	2	14	1208	1212	1411	8	2	15	1	1	0	1
5	1859	0	0.5	1	3	0	22	0.7	164	1	7	1004	1654	1067	17	1	10	1	0	0	1
6	1821	0	1.7	0	4	1	10	0.8	139	8	10	381	1018	3220	13	8	18	1	0	1	3
7	1954	0	0.5	1	0	0	24	0.8	187	4	0	512	1149	700	16	3	5	1	1	1	0
8	1445	1	0.5	0	0	0	53	0.7	174	7	14	386	836	1099	17	1	20	1	0	0	0
9	509	1	0.6	1	2	1	9	0.1	93	5	15	1137	1224	513	19	10	12	1	0	0	0
10	769	1	2.9	1	0	0	9	0.1	182	5	1	248	874	3946	5	2	7	0	0	0	3
11	1520	1	2.2	0	5	1	33	0.5	177	0	18	151	1005	3826	14	9	13	1	1	1	3
12	1815	0	2.8	0	2	0	33	0.6	159	4	17	607	748	1482	18	0	2	1	0	0	1
13	803	1	2.1	0	7	0	17	1	198	4	11	344	1440	2680	7	1	4	1	0	1	2
14	1866	0	0.5	0	13	1	52	0.7	185	1	17	356	563	373	14	9	3	1	0	1	0
15	775	0	1	0	3	0	46	0.7	159	2	16	862	1864	568	17	15	11	1	1	1	0
16	838	0	0.5	0	1	1	13	0.1	196	8	4	984	1850	3554	10	9	19	1	0	1	3
17	595	0	0.9	1	7	1	23	0.1	121	3	17	441	810	3752	10	2	18	1	1	0	3
18	1131	1	0.5	1	11	0	49	0.6	101	5	18	658	878	1835	19	13	16	1	1	0	1
19	682	1	0.5	0	4	0	19	1	121	4	11	902	1064	2337	11	1	18	0	1	1	1

Figure 7: First 20 rows of the dataset

Effect of Feature Selection on the Accuracy of Machine Learning Model

Now we will select 5 best features from our dataset:

```
import pandas as pd
import numpy as np
from sklearn.feature_selection import SelectKBest
from sklearn.feature_selection import f_classif
data = pd.read_csv("train.csv")
X = data.iloc[:,0:20]
y = data.iloc[:,-1]
bestfeatures = SelectKBest(score_func=f_classif, k=5)
fit = bestfeatures.fit(X,y)
dfscores = pd.DataFrame(fit.scores_)
dfcolumns = pd.DataFrame(X.columns)
featureScores = pd.concat([dfcolumns,dfscores],axis=1)
featureScores.columns = ['Feature','Score']
print(featureScores.nlargest(5,'Score'))
```

Output of the program:

```
In [1]: runfile('C:/Users/DELL/Desktop/RESEARCH/
RESEARCH/archive')
          Feature        Score
13            ram  3520.110824
0   battery_power    31.598158
12       px_width    22.620882
11      px_height    19.484842
8       mobile_wt     3.594318

In [2]:
```

The output of the above program clearly show that ram is the highly correlated feature with price range followed by battery power, pixel width and height. Now we will see how the accuracy and prediction power of (KNN, Decision Tree, Multi-layer Perceptron) can be affected by the process of feature selection, to do the experiments first we will train the algorithms by selecting all the features:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score
data = pd.read_csv("train.csv")
X = data.iloc[:,0:20]
y = data.iloc[:,-1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
knn_classifier = KNeighborsClassifier(n_neighbors = 5)
knn_classifier.fit(X_train, y_train)
knn_pred=knn_classifier.predict(X_test)
knn_score=accuracy_score(y_test,knn_pred)
print('Accuracy of K-Nearest Neighbors using all 20 features: ',knn_score)
tree_classifier=DecisionTreeClassifier(criterion='entropy')
tree_classifier.fit(X_train,y_train)
tree_pred=tree_classifier.predict(X_test)
tree_score=accuracy_score(y_test,tree_pred)
```

```
print('Accuracy of Decision Tree using all 20 features: ',tree_score)
ANN_classifier=Perceptron(random_state=1)
ANN_classifier.fit(X_train, y_train)
ANN_pred=ANN_classifier.predict(X_test)
ANN_score=accuracy_score(y_test,ANN_pred)
print('Accuracy of Artificial Neural Network using all 20 features: ',ANN_score)
```

Output of the program:

```
In [2]: runfile('C:/Users/DELL/Desktop/RESEARCH/archive/all_features.py',
RESEARCH/archive')
Accuracy of K-Nearest Neighbors using all 20 features:  0.498
Accuracy of Decision Tree using all 20 features:  0.848
Accuracy of Artificial Neural Network using all 20 features:  0.712
```

Now it is time to train the algorithms by using 5 best features and ignore the rest of the features which are not important for our model construction and can see the improvement in accuracy:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import Perceptron
from sklearn.metrics import accuracy_score
data = pd.read_csv("train.csv")
best_data=data[['ram','battery_power','px_width','px_height','mobile_wt','price_range']].copy()
X = best_data.iloc[:,0:5]
y = best_data.iloc[:,-1]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 0)
from sklearn.preprocessing import StandardScaler
sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
knn_classifier = KNeighborsClassifier(n_neighbors = 5)
knn_classifier.fit(X_train, y_train)
knn_pred=knn_classifier.predict(X_test)
knn_score=accuracy_score(y_test,knn_pred)
print('Accuracy of K-Nearest Neighbors using 5 best features: ',knn_score)
tree_classifier=DecisionTreeClassifier(criterion='entropy')
tree_classifier.fit(X_train,y_train)
tree_pred=tree_classifier.predict(X_test)
tree_score=accuracy_score(y_test,tree_pred)
print('Accuracy of Decision Tree using 5 best features: ',tree_score)
ANN_classifier=Perceptron(random_state=1)
ANN_classifier.fit(X_train, y_train)
ANN_pred=ANN_classifier.predict(X_test)
ANN_score=accuracy_score(y_test,ANN_pred)
print('Accuracy of Artificial Neural Network using 5 best features: ',ANN_score)
```

Output of the program:

```
In [3]: runfile('C:/Users/DELL/Desktop/RESEARCH/archive/best_data.py',
archive')
Accuracy of K-Nearest Neighbors using 5 best features:  0.826
Accuracy of Decision Tree using 5 best features:  0.866
Accuracy of Artificial Neural Network using 5 best features:  0.78
```

6. CONCLUSION

Feature selection is an important concept in machine learning, because it may have huge effect on accuracy and prediction power of a machine learning model. Moreover, removing extra variables from a dataset decreases the overall complexity of the model. It reduces overfitting, improves accuracy, and reduces training time.

In this research paper, we have investigated the impacts of feature selection on the accuracy of a machine learning model by using three different classification algorithms (K-Nearest Neighbors, Decision Tree and Artificial Neural Network). We observed that the accuracy scores of these algorithms are highly affected by the process of feature selection. The experiment clearly shows that the accuracy scores of the algorithms will increase from 49%, 84% and 71% to 82%, 86% and 78% accordingly. Therefore, feature selection is highly recommended especially in high dimensional datasets.

REFERENCES

1) J. Miao and L. Niu, "A Survey on Feature Selection," *Procedia Comput. Sci.*, vol. 91, no. Itqm, pp. 919–926, 2016, doi: 10.1016/j.procs.2016.07.111.
2) E. M. Karabulut, S. A. Özel, and T. İbrikçi, "A comparative study on the effect of feature selection on classification accuracy," *Procedia Technol.*, vol. 1, pp. 323–327, 2012, doi: 10.1016/j.protcy.2012.02.068.
3) R. C. Chen, C. Dewi, S. W. Huang, and R. E. Caraka, "Selecting critical features for data classification based on machine learning methods," *J. Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00327-4.
4) Y. Akhiat, Y. Manzali, M. Chahhou, and A. Zinedine, "A New Noisy Random Forest Based Method for Feature Selection," *Cybern. Inf. Technol.*, vol. 21, no. 2, pp. 10–28, 2021, doi: 10.2478/cait-2021-0016.
5) A. Cardew, *Antiquity and anxiety: Freud, jung, and the impossibility of the archaic*. 2018. doi: 10.4324/9780203733332.
6) K. Taunk, S. De, S. Verma, and A. Swetapadma, "A brief review of nearest neighbor algorithm for learning and classification," *2019 Int. Conf. Intell. Comput. Control Syst. ICCS 2019*, no. May, pp. 1255–1260, 2019, doi: 10.1109/ICCS45141.2019.9065747.
7) M. Suyal and P. Goyal, "A Review on Analysis of K-Nearest Neighbor Classification Machine Learning Algorithms based on Supervised Learning," *Int. J. Eng. Trends Technol.*, vol. 70, no. 7, pp. 43–48, 2022, doi: 10.14445/22315381/IJETT-V70I7P205.
8) S. Raschka, *Python machine learning*. Packt publishing ltd., 2015.
9) Pooja Gulati, Amita Sharma, and Manish Gupta, "Theoretical Study of Decision Tree Algorithms to Identify Pivotal Factors for Performance Improvement: A Review Pooja Gulati," *Int. J. Comput. Appl.*, vol. 141, no. 14, pp. 975–8887, 2016.
10) M. Stephen, *Machine Learning An Algorithmic Perspective Second Edition*. 2014. [Online]. Available: https://b-ok.cc/book/2543746/ef80cb
11) T. Price and N. Lindqvist, "Evaluation of Feature Selection Methods for Machine Learning Classification of Breast Cancer," pp. 1–40, 2018.
12) Y. Bouchlaghem, Y. Akhiat, and S. Amjad, "Feature Selection: A Review and Comparative Study," *E3S Web Conf.*, vol. 351, pp. 1–6, 2022, doi: 10.1051/e3sconf/202235101046.
13) Y. B. Wah, N. Ibrahim, and H. A. Hamid, "Feature selection methods : Case of filter and wrapper approaches for maximising classification accuracy SCIENCE & TECHNOLOGY Feature Selection Methods : Case of Filter and Wrapper Approaches for Maximising Classification Accuracy," no. May 2020, 2018.
14) A. Jović, K. Brkić, and N. Bogunović, "A review of feature selection methods with applications," *2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc.*, no. May, pp. 1200–1205, 2015, doi: 10.1109/MIPRO.2015.7160458.