sciendo Journal of Social and Economic Statistics

Vol. 11, No. 1-2, Summer-Winter 2022
DOI: 10.2478/jses-2022-0007

OPEN ACCESS CC BY

A MACHINE LEARNING APPROACH TO IDENTIFY THE FEATURE IMPORTANCE FOR ADMISSION IN THE NATIONAL MILITARY HIGH SCHOOLS[*]

Dimitrie-Daniel PLĂCINTĂ[a*]

[a)] The Bucharest University of Economic Studies, Romania

Abstract

The article provides the impact of different averages (feature importance) within the admission exam for the national military high schools using and testing three supervised machine learning algorithms: logistic regression, K-Nearest Neighbors, and random forest. For this purpose, I have used the list with the results of candidates compounded by 430 rows, an unclassified document posted on the national military high school website, with details about: the final admission grade, the general grade for graduating of the secondary school, the general grade obtained at the national assessment, the mark obtained at admission test from Romanian language and mathematics items, etc. From the machine learning perspective, I have built a Jupyter notebook, a Python code using the specialized ML libraries (numpy, pandas, matplotlib, sklearn). In conclusion, the logistic regression algorithm identified the 'feature importance' (how each variable contributes to the predicted model) for admission in the national military high school: the mark obtained at admission test from Romanian language and Mathematics items - 4.821834, the general average obtained at the national assessment - 0.584434, the general average for graduating of the secondary school - 0.285446, etc. These are the expected results based on the admission methodology for the national military high schools.

Keywords: machine learning, feature importance, admission, national military high schools
JEL Classification: I21, L86, O39, P46

[*] Corresponding author, **Dimitrie-Daniel Plăcintă** – daniel.placinta@csie.ase.ro

[*] This paper was presented at the International Conference on Applied Statistics ICAS 2022. The authors thank participants for their useful feedback.

1. Introduction

Admission to high school is very important for every Romanian pupil, especially when the procedure is carried out for one of the five national military high schools in Alba-Iulia, Breaza, Câmpulung Moldovenesc, Constanța, and Craiova. Every candidate is attracted by the tradition, institutional brand, and excellent results obtained by military pupils are local, regional, national, or international level, at Olympiads, sport competitions, or military contests.

Before the admission exam for a national military high school, the candidate must complete the national evaluation assessment ("Evaluarea Națională") which is a mandatory examination for every graduate of secondary school. These results will consist of the general grade obtained in the national assessment (MEV).

The general average for graduating from secondary school (MVVIII) is another important factor for admission to high schools in Romania, including, of course, the national military high schools. The admission average (MA) is calculated based on MVIII (20%) and MEV (80%) with 2 decimals, without rounding, according to the admission methodology of (Ministerul Apărării Naționale Colegiul Național Militar 'Dimitrie Cantemir' 2022).

For military admission into a high school apart from mentioned averages, every candidate must pass the medical exam, the psychological assessment, and the evaluation organized within the examination centres for the five national military high schools to obtain the mark at admission test from Romanian language and Mathematics items - (NT).

The final admission average (MFA) is calculated based on MA (20%) and NT (80%) with 2 decimals, without rounding, according to the admission methodology of (Ministerul Apărării Naționale Colegiul Național Militar 'Dimitrie Cantemir' 2022) as we can see in the (equation no.1).

$$MFA=20\% *(20\%*MVIII + 80\%* MEV) + 80\%*NT \quad (1)$$

MFA will represent the last average which will constitute the criteria for admission with the national military high schools.

Considering the geopolitical context around the Black Sea Extent Region, the Romanian government, through the Ministry of Defense and Ministry of Education, adopted a new approach to prepare and educate the younger generation of military pupils by reopening two national military high schools.

Beginning with the school year 2020-2021, two national military high schools were reopened, one in Constanța (Colegiul Național Militar 'Alexandru Ioan Cuza') and the second in Craiova (Colegiul Național Militar 'Tudor Vladimirescu'), with classes for the vocational stream, military profile, mathematics-informatics specialization, with intensive teaching in English.

Before reopening the national military high schools in Constanța and Craiova, many good candidates remained outside the admission list for free places within the national military high schools. With about 240 new places available in the southern part of Romania (Breaza,

Constanța, Craiova) these good candidates can obtain the admitted status and start their military journey.

The graduates of the national military high schools are the most important pool of candidates for the military academies (Direcția Generală Învățământ Universitar | Ministerul Educației, 2021) from Bucharest (Academia Tehnică Militară "Ferdinand I", Universitatea Naţională de Apărare "Carol I", Academia Naţională de Informaţii "Mihai Viteazul", Academia de Poliţie "Alexandru Ioan Cuza"), Constanța (Academia Navală "Mircea Cel Bătrân"), Brașov (Academia Forţelor Aeriene "Henri Coandă"), Sibiu (Academia Forţelor Terestre "Nicolae Bălcescu"), also for the non-commissioned officer academies.

At the beginning of the school year, is signed an educational contract between the legal representants of candidates and the minister of Defense through the national military high schools where are mentioned the conditions, restrictions, responsibilities for each stakeholder.

The main important aspect is that all conditions are ensured by the national military high school with the condition for the graduates to sustain the admission exam for the military academies or for the non-commissioned officer academies, otherwise the candidate should pay the fee for entire period of tuition. If a candidate will not enter within a military academy, there is the option to continue the military studies at the non-commissioned officer academies.

The impact of admission to national military high schools is important for Romanian society because behind every candidate is his family. We can multiply the number of candidates with at least three persons to get the social impact. From the military students along their careers is selected the leadership of the Romanian army.

In this paper, I will identify the 'feature importance'(how each variable contributes to the predicted model or best set of explanatory variables) of the independent variables (the general average for the graduation of secondary school (MVVIII), the general average obtained in the national assessment (MEV), the mark obtained in the Romanian language admission test and the mathematical elements (NT)) against the dependent variable, 'Status', which has two values 'Admitted' or 'Unadmitted'.

In this study, I will evaluate three supervised machine learning algorithms (logistic regression, K-Nearest Neighbors, and random forest.), compare them to identify the best algorithm which will extract the feature importance for the before mentioned independent variables.

2. Literature review

During the research phase of this paper, I have run multiple queries for statements, such as 'feature importance admission military schools', 'feature importance logistic regression admission exam', 'machine learning admission military high schools', against the most important scientific resources: Web of Science, Science Direct, Google Scholar, ProQuest, IEEE.

Unfortunately, for the high school level there are no relevant papers or studies on the above topics from other researchers or from different countries with a similar or closer approach.

This is a strong point of this paper, also an opportunity to open a path of investigation for the admission in the military high schools using machine learning models.

Instead, I found other views about the student admission and major selection, when machine learning algorithms were used, KNN (K-Nearest Neighbors) has the highest accuracy (Assiri et al., 2022).

In the paper (Pasic and Kucak, 2020) the Algebra details of high school pupils are used to predict the possibility of successfully completing some programs, for unsuccessful case the candidates are receiving the recommendations to study a different program.

Logistic regression was used with good results (accuracy of 78%), together with Naive Bayes, decision tree, random forest algorithms (over 90% precision) to explore the impact of results from high school at Mathematics, English, and high school average before college entry in the paper (Embarak, 2020).

Using Weka software against sixteen machine learning algorithms, the researchers predicted the enrolment of students at the university taking into consideration the averages of 'every year of high school, matura exam results' and the average of the admission exam (Ujkani et al., 2021).

Romanian military high schools act like isolated islands in the educational landscape from many points of view: orientation on educational excellence through national and European projects, initial military socialization, respect of military traditions, rituals, alignment of behaviors to the very strong military institutional brand, etc.

3. Methodology

For this exploratory machine learning process, I have performed quantitative research using a machine learning approach the identification of the best set of explanatory variables for the admission within the national military high schools after testing the performance of three supervised algorithms (logistic regression, K-Nearest Neighbors, and random forest) and selecting logistic regression model.

The machine learning pipeline is represented by the following phases: data cleaning and importing the data into a data frame, removing the variables (Current_number, Case, MFA) which will not impact the dependent variable (Status), encoding the categorial values into the numerical values, checking the imported data from the statistics point of view, shuffling the data set, splitting the data into training and test data sets, creating an instance of the model, comparing models, evaluating the performance of logistic regression model.

The main research hypothesis of this article is that the mark obtained in the Romanian language admission test and the mathematical elements (NT) is the most important factor for the admission in military high schools within the admission formula (equation no.1).

For this machine learning approach, I have used a public data set from that official website of the National Military High School "Dimitrie Cantemir"(Colegiul Național Militar 'Dimitrie Cantemir' Breaza, 2022) where an examination centre was organized.

This data set was composed by 430 rows and 11 columns as we can see in the (fig. no.1), 159 rows with 'Admitted' status and 271 rows with the 'Unadmitted' status.

	Current_number	Sex	County	Case	MFA	NT	MEV	MATE	MWIII	Status
0	1	M	PH	NU	9.94	10.0	9.67	9.85	9.99	Admis
1	2	M	B	NU	9.91	10.0	9.52	9.60	9.87	Admis
2	3	F	PH	NU	9.89	10.0	9.40	9.45	9.89	Admis
3	4	M	PH	NU	9.89	10.0	9.37	10.00	9.94	Admis
4	5	M	DB	NU	9.72	9.7	9.80	9.90	9.99	Admis
...	...	...	...	...	...	...	...	...	...	...
425	426	F	B	NU	6.39	6.1	7.35	7.30	8.35	Neadmis
426	427	F	B	NU	6.36	6.1	7.00	7.20	9.08	Neadmis
427	428	M	B	NU	6.32	6.1	6.70	5.75	9.44	Neadmis
428	429	M	BZ	NU	6.17	6.1	6.05	5.40	8.23	Neadmis
429	430	M	B	NU	6.13	6.1	5.90	4.85	7.76	Neadmis

430 rows × 10 columns

Figure no.1: Representation of the data set in a Jupyter notebook file

Source: Authors' own research results/contribution

Because the source file was a PDF(portable document format) and not well formatted for data exploration analysis, I had converted this initially into an Excel file, with .xlsx extension, using an online converter, https://www.ilovepdf.com/pdf_to_excel.

After the conversion to.xlsx file, not all content was prepared for data analysis, and I manually deleted the saved pictures, unnecessary columns, table headers, etc.

After importing the next Python libraries (NumPy, fundamental package for scientific computing in Python, pandas, open-source library for data transformation, matplotlib, the Python library for generating plots, sklearn, or Scikit-learn, efficient machine learning library for Python used for predictive data analysis) (fig. no.2) within a Jupyter notebook file created from Anaconda, I encountered a lot of issues at data frame creation in pandas, which were solved manually based on the number of rows displayed on the error messages, and validated the problematic data format.

Anaconda(Anaconda.com, 2022) is the most popular data science platform/distribution, with a package manager and environment manager, the single place where data scientists can perform their Python/R approaches, with thousands of open-source packages and libraries.

```
# Import libraries
import numpy as np
import pandas as pd
%matplotlib inline
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')

from sklearn.utils import shuffle
from sklearn import neighbors, datasets, preprocessing
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.ensemble import RandomForestClassifier

# Read the dataset
df = pd.read_csv('dataset_work_final.csv', sep=",")
df
```

Figure no.2: Importing the Python library used in the Jupyter notebook file

Source: Authors' own research results/contribution

To implement the machine learning approach, I have created a data frame in pandas, by reading the details from the dataset_work_final.csv file and displayed the first and last rows details of the candidates (fig. no.1) in alphabetical order of the final admission average, then I have validated if all rows from the data set are completed by applying count() function on the data frame, df.

Because the columns Current_number, Candidate, Case, and MFA did not impact the status of candidates, we can remove these from the data frame (fig. no.3).

```
# Drop columns
df_final=df.drop(columns=['Current_number','Candidate','Case','MFA'])
df_final
```

	Sex	County	NT	MEV	MATE	MWIII	Status
0	M	PH	10.0	9.67	9.85	9.99	Admis
1	M	B	10.0	9.52	9.60	9.87	Admis
2	F	PH	10.0	9.40	9.45	9.89	Admis
3	M	PH	10.0	9.37	10.00	9.94	Admis
4	M	DB	9.7	9.80	9.90	9.99	Admis
...	...	...	...	...	...	...	...
425	F	B	6.1	7.35	7.30	8.35	Neadmis
426	F	B	6.1	7.00	7.20	9.08	Neadmis
427	M	B	6.1	6.70	5.75	9.44	Neadmis
428	M	BZ	6.1	6.05	5.40	8.23	Neadmis
429	M	B	6.1	5.90	4.85	7.76	Neadmis

430 rows × 7 columns

Figure no.3: Drop unnecessary columns and display the new data set

Source: Authors' own research results/contribution

As we can see from above details (fig. no.3) the columns Sex, County, Status are represented as categorial values. Because machine learning algorithms use numerical values, we need to convert the mentioned columns from categorial values to numerical values by using the replace() function against our data frame.

In this process, we will encode 'F' with '0', 'M' with '1', counties from 'B' to 'PH' with '0' to '9' and 'Neadmis' – 'Unadmitted' with '0' and 'Admis' – 'Admitted' with '1'.

An important aspect of the machine learning approach is to check the data from data frame against the most important statistics functions as I have represented in below representation (fig. no.4) to fully make sure that data is validated also from the qualitative point of view.

We can remark that all rows are completed (430), also that the values for the NT, MEV, MATE, MVIII grades are in the anticipated ranges, etc.

```
# Summary statistics
df.describe()
```

	Sex	County	NT	MEV	MATE	MWIII	Status
count	430.000000	430.000000	430.000000	430.000000	430.000000	430.000000	430.000000
mean	0.672093	3.783721	7.496744	8.407558	8.305233	9.505256	0.369767
std	0.469998	3.640259	0.968496	0.750087	0.934424	0.418943	0.483304
min	0.000000	0.000000	6.100000	5.400000	4.850000	7.760000	0.000000
25%	0.000000	0.000000	6.700000	7.970000	7.750000	9.260000	0.000000
50%	1.000000	2.000000	7.300000	8.470000	8.375000	9.620000	0.000000
75%	1.000000	9.000000	8.200000	8.950000	9.050000	9.847500	1.000000
max	1.000000	9.000000	10.000000	9.820000	10.000000	10.000000	1.000000

Figure no.4: Summary statistics of transformed data frame

Source: Authors' own research results/contribution

Because the data set was extracted in a descending order by the MFA column, we need to shuffle the values from the data frame with the shuffle() function using as first parameter the name of the data frame – 'df' and the assignment of value '1' for the second parameter 'random_state'.

The shuffled data frame can be observed in (fig. no.5), rows of the data frame are displayed in randomized order, and in these conditions, the data frame is prepared for the next transformations of our machine learning approach.

	Sex	County	NT	MEV	MATE	MWIII	Status
227	1	1	7.3	8.40	8.15	9.24	0
208	1	9	7.3	9.02	9.35	9.80	0
164	0	9	7.6	8.87	8.35	9.76	0
277	1	8	7.0	8.05	8.60	9.32	0
62	0	0	8.5	9.57	9.35	9.93	1
...	...	...	...	...	...	...	...
255	1	0	7.0	8.92	8.80	9.63	0
72	1	8	8.8	7.72	8.35	9.10	1
396	1	8	6.1	8.25	7.70	8.96	0
235	1	0	7.3	8.15	8.10	9.47	0
37	0	2	8.8	9.65	9.55	9.86	1

430 rows × 7 columns

Figure no.5: Shuffled data frame use in the machine learning approach

Source: Authors' own research results/contribution

Having the data prepared for manipulation, I extracted in the 'X' data frame the independent variables (Sex, County, NT, MEV, MATE, MVIII) and in the 'y' data frame the target variable – Status (fig. no.6).

```
# Modelling - extracting the independent and target variabels
X=df.drop("Status",axis=1)
y=df["Status"]
```

Figure no.6: Extracting the independent and target variables

Source: Authors' own research results/contribution

For every machine learning approach to be effective, the entire data set should be divided into training and test data sets. c Currently, I reserved 70% from data set to training set and 30% to the test data set by setting the value 0.3 for the 'test_size' parameter as is described in (fig. no.7).

```
# Split the data into train and test datasets
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3)

# Scale the X values
scaler = StandardScaler()

X_train = scaler.fit_transform(X_train)
X_test = scaler.fit_transform(X_test)
```

Figure no.7: Split the data into training and test data sets, applying the StandardScaler()

Source: Authors' own research results/contribution

From the sklearn library I imported the LogisticRegression module, created an instance of the logistic regression model, and trained the model on the data (fig. no.8).

```
#Fitting Logistic Regression to the training set
from sklearn.linear_model import LogisticRegression
classifier= LogisticRegression(random_state=0)
classifier.fit(X_train, y_train)
```

```
LogisticRegression(random_state=0)
```

Figure no.8: Fitting LogisticRegression () to the training data set

Source: Authors' own research results/contribution

My goal is to compare the performance of logistic regression, KNN and random forest algorithms and for this reason I built the 'models' dictionary in Python, and the fit_and_score() function which will receive the model, X_train, X_test, y_train, y_test as parameters (fig. no.9).

```
# Put models into a dictionary
models={"Logistic Regression":LogisticRegression(),
         "KNN": KNeighborsClassifier(),
         "Random Forest":RandomForestClassifier()}

# Create a function for fit and score
def fit_and_score(model,X_train,X_test,y_train,y_test):
    model_scores={}
    for name, model in models.items():
        # Fit the model
        model.fit(X_train,y_train)
        # Evaluate the model
        model_scores[name]=model.score(X_test,y_test)
    return model_scores # return the dictionary model_scores
```

Figure no.9: Create a dictionary and a function which retrieves the model scores

Source: Authors' own research results/contribution

After calling the function fit_and_score on the models, I retrieved the next score / accuracy at that moment of running the Jupyter notebook file: logistic regression: 0.9844, KNN: 0.9612, random forest: 0.9767 (fig. no.10).

An important observation should be done in this point of discussion is that the next results presented as figures can differ with small changes at every running of the Jupyter notebook file due to multiple factors like different training data, because we are shuffling every time the initial data set, 'because of stochastic learning algorithms, because of stochastic evaluation procedures, because of differences in platform'(Brownlee, 2020).

```
# Calculate the score for models
model_scores=fit_and_score(model=models,X_train=X_train,X_test=X_test,y_train=y_train,y_test=y_test)
model_scores
```

```
{'Logistic Regression': 0.9844961240310077,
 'KNN': 0.9612403100775194,
 'Random Forest': 0.9767441860465116}
```

Figure no.10: Call the function fit_and_score () and display the results

Source: Authors' own research results/contribution

From the evaluated, compared models, logistic regression obtained the highest score, and I will continue to explore the importance feature (the importance of every independent variable on the dependent variable (Status)) (fig. no.11).

The result (importance feature) is stored in the column 'Importance' of the 'importances' data frame, which is receiving the values of the 'coef_' attribute of the fitted logistic regression algorithm.

```
# Apply Logistic Regression
X_train = pd.DataFrame(X_train, columns = X.columns)
model = LogisticRegression()
model.fit(X_train, y_train)

# Get the importance of features
importances = pd.DataFrame(data={
    'Attribute': X_train.columns,
    'Importance': model.coef_[0]
})
```

Figure no.11: Get the importance of feature data frame

Source: Authors' own research results/contribution

In the current step I can unveil the key results of this paper, the coefficients for the feature importance in descending order: NT - 4.821834, MATE - 0.901912, MEV -0.584434, MWIII -0.285446, Sex -0.183213, County - '-0.073013' (fig. no.12).

The results obtained are the expected ones; a small observation can be made on MATE, which was second place, because the MATE average is used to order the candidates with the

same MFA average, and this was not used frequently in the current data set. The gender and county criteria did not influence the result, the status of the candidates – Admitted or Unadmitted.

```
# Ordered Importance feature
importances.sort_values(by="Importance", ascending=False)
```

	Attribute	Importance
2	NT	4.821834
4	MATE	0.901912
3	MEV	0.584434
5	MWIII	0.285446
0	Sex	0.183213
1	County	-0.073013

Figure no.12: Show the importance feature on descending order

Source: Authors' own research results/contribution

The power of machine learning libraries is that we can apply the predict function to anticipate the result of dependent variable (Status), based on the entire X_test data set utilizing the learnt information by the logistic regression algorithm within the training phase (fig. no.13). After applying the predict function and saving the result on y_pred array I have displayed the predicted results for the test data set.

```
#Predicting the test set result
y_pred= classifier.predict(X_test)
y_pred
#y_pred.size  #129
```

```
array([0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
       1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 1, 1, 1, 0, 1, 1,
       0, 1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0, 1,
       1, 1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0,
       1, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 1, 1, 1, 1, 0, 1, 0, 0, 0, 1,
       1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0],
      dtype=int64)
```

Figure no.13: Predict the result based on X test data set

Source: Authors' own research results/contribution

The machine learning approaches are evaluated from the performance perspective, and for this purpose I have built, calculated, and displayed (using seaborn library) the expected values for confusion matrix (fig. no.14) where on rows we can see the actual, existing values and on the columns, we can see the predicted values by the classification model.

For the discussed model, 80 false actual values were predicted as false – True Negative, 47 true actual values were predicted as true – True Positive, only two true actual values were identified wrong as false values – False Negative, 0 values were identified as False Positive

(top-right corner of confusion matrix) and in this manner we can conclude that the extracted classification model is performant.

```
#Creating the Confusion matrix
from sklearn.metrics import confusion_matrix
from sklearn import metrics
import seaborn as sns

cm= confusion_matrix(y_test,y_pred)

ax = sns.heatmap(cm, annot=True, fmt='g');
ax.set_title('Confusion Matrix');
ax.set_xlabel('Predicted values')
ax.set_ylabel('Actual values');

plt.show()
```

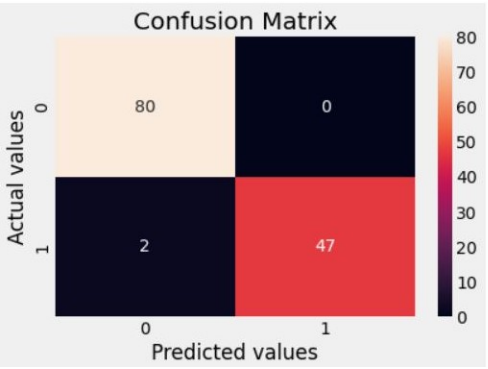


Figure no.14: Build and show the confusion matrix results

Source: Authors' own research results/contribution

To continue the evaluation of the model we can calculate other metrics, like accuracy((True Positive+True Negative)/Total Predictions)=0.9844, precision(True Positive/(True Positive+False Positive))=0.9545, sensitivity_recall (True Positive/(True Positive+False Negative))=1.0, specificity(True Negative/(True Negative+False Positive))=0.9770, f1_score(2*((Precision*Sensitivit)/(Precision+Sensitivit))=0.9767 (fig. no.15), (https://www.w3schools.com/, 2022).

'Accuracy measures how often the model is correct, precision does not evaluate correctly predicted negative cases, sensitivity measures how good the model is in predicted positives, specificity measures how good the model is in predicted negatives, and F-score is the harmonic mean of precision and sensitivity"(https://www.w3schools.com/, 2022). All calculated metrics confirm the performance of the identified logistic regression model.

```
# Applying all metrics from https://www.w3schools.com/python/python_ml_confusion_matrix.asp

actual=y_test
predicted=y_pred

Accuracy = metrics.accuracy_score(actual, predicted)
Precision = metrics.precision_score(actual, predicted)
Sensitivity_recall = metrics.recall_score(actual, predicted)
Specificity = metrics.recall_score(actual, predicted, pos_label=0)
F1_score = metrics.f1_score(actual, predicted)

# Print metrics:
print("Accuracy=",Accuracy)
print("Precision=",Precision)
print("Sensitivity_recall=",Sensitivity_recall)
print("Specificity=",Specificity)
print("F1_score=",F1_score)
```

```
Accuracy= 0.9844961240310077
Precision= 0.9545454545454546
Sensitivity_recall= 1.0
Specificity= 0.9770114942528736
F1_score= 0.9767441860465117
```

Figure no.15: Apply multiple metrics against actual and predicted results

Source: Authors' own research results/contribution

4. Results and discussions

After applying the previous approach, I have obtained the expected results in descending order the feature importance for admission in the national military high school: the mark obtained at admission test from Romanian language and Mathematics items - 4.821834, the general average obtained at the national assessment - 0.584434, the general average for graduating of the secondary school - 0.285446. These results can differ with small changes at every running of the Jupyter notebook file, as I have highlighted in the methodology section.

I have remarked that the independent variables of gender and county did not influence the result for the Status column. I have eliminated the MFA from our analysis because this is the result after applying the admission formula. and to offer the opportunity for the machine learning algorithms to learn from labelled data.

As I mentioned in the literature review section this is an original (data set, unique topic) approach with machine learning in the background that was applied to the admission to the military high schools from Romania.

5. Conclusions

In this paper I have created a machine learning model with the help of logistic regression algorithm that then predicts with high accuracy (over 98%) the status for a data set composed by the main important averages of candidates for admission on the national military high schools: the general average for graduating of the secondary school, the general average obtained at the national assessment, the mark obtained at admission test from Romanian language and mathematics items, etc.

The approach is new up to the moment of writing this article and can be reconfirmed/tuned with the new data from the next years using the candidates' details.

National military high schools can create simulators and share these with potential candidates to measure their chances of success. The data set utilized was processed only from one of the five military examination centres, and future research can consolidate the data from all five examination centres to offer a clearer picture for admission to the national military high schools.

A potential limitation can arise from the new methodologies of admission in case the new educational laws are implemented, where it is mentioned that the general average for graduating of the secondary school should not be taken into consideration for high school admission. Another aspect which cannot be fully controlled is the fact that the results are provided at every running of a Jupyter notebook file and can differ with small changes at every iteration.

For this machine learning approach, I used the public data from the examination centre where GDPR rules were respected. All statements related to military experience are based on my personal and professional interactions with military life, and this paper is an opportunity to share these details with the larger public and does not reflect the official position of the National Military High School 'Dimitrie Cantemir' Breaza, unless specified in references.

References

Anaconda.com (2022). Anaconda. [Accessed 6 November 2022]. Available from: https://www.anaconda.com/.

Assiri, B., Bashraheel, M. and Alsuri, A. (2022). Improve the Accuracy of Students Admission at Universities Using Machine Learning Techniques *In*: *2022 7th International Conference on Data Science and Machine Learning Applications (CDMA)*. IEEE, pp.127–132.

Brownlee, J. (2020). Why Do I Get Different Results Each Time in Machine Learning? *https://machinelearningmastery.com/*. [Online]. [Accessed 1 December 2022]. Available from: https://machinelearningmastery.com/different-results-each-time-in-machine-learning/.

Colegiul Național Militar ‘Dimitrie Cantemir’ Breaza (2022). Tabel nominal cu rezultatele candidaților arondați să susțină admiterea în Colegiul Național Militar ‘Dimitrie Cantemir’ în ordinea descrescătoare a mediilor de admitere, cu unitatea școlară la care au fost adimiși - Rezultate finale.

Direcția Generală Învățământ Universitar | Ministerul Educației (2021). Lista Instituțiilor de Învățământ Superior de Stat Militare. *edu.ro*. [Online]. [Accessed 30 October 2022]. Available from: https://www.edu.ro/sites/default/files/fisiere%20articole/Lista%20IISSM%202021.pdf.

Embarak, O. (2020). Apply Machine Learning Algorithms to Predict At-Risk Students to Admission Period, *In*: *2020 Seventh International Conference on Information Technology Trends (ITT)*. IEEE, pp.190–195.

https://www.w3schools.com/ 2022. Machine Learning - Confusion Matrix. *https://www.w3schools.com/*.

Ministerul Apărării Naționale Colegiul Național Militar ‘Dimitrie Cantemir’ (2022). Metodologie de admitere, pp.15–20. [Accessed 23 October 2022]. Available from: https://www.cantemircml.ro/.

Pasic, D., Kucak, D. (2020). Machine learning model for detecting high school students as candidates for drop-out from a study program, *In*: *2020 43rd International Convention on Information, Communication and Electronic Technology (MIPRO)*. IEEE, pp.1140–1144.

Ujkani, B., Minkovska, D. and Stoyanova, L. (2021). A Machine Learning Approach for Predicting Student Enrollment in the University, *In*: *2021 XXX International Scientific Conference Electronics (ET)*. IEEE, pp.1–4.