International Journal of Advance and Applied Research
www.ijaar.co.in
ISSN – 2347-7075 Impact Factor – 8.141
Peer Reviewed Bi-Monthly
Vol. 7 No. 28 March - April - 2026

Machine Learning Optimized Model Using Gradient Discent for Enginnernig Admission Cut-Off

Dr. Ashok Sahebrao Mhaske
Department of Mathematics, Dada Patil College, Karjat, Dist- Ahilyanagar, Maharashtra, India affiliated to Savitribai Phule Pune University Pune.
Corresponding Author – Dr. Ashok Sahebrao Mhaske
DOI - 10.5281/zenodo.19396461

Abstract:

The engineering admission process in India is governed by centralized entrance examinations such as the (MHT-CET) Maharashtra Common Entrance Test or (JEE) Joint Entrance Examination. The existing system depends on rule-based allocation mechanisms and static cutoff that lack predictive personalization, intelligence and transparency. This research proposes the Gradient Descent algorithm of Machine Learning for engineering admission prediction model. Logistic Regression is applied to find the probability of admission based on multiple attributes, like entrance examination score, reservation category, higher secondary (PCM) percentage, institutional preference ranking, demographic factors and historical cutoff trends. Experimental evaluation is performed using simulated candidate profiles and historical admission datasets.

Keywords: Machine Learning, Gradient Descent, Engineering, Admission, Logistic Regression

Introduction:

Engineering education plays a important role in economic development and technological advancement. In India, admission to engineering programs is conducted through centralized entrance examinations such as the state-level examinations like the Maharashtra Common Entrance Test (MHT-CET) and Joint Entrance Examination (JEE). The traditional admission framework based on seat allocation algorithms. However, with the exponential growth in the number of applicants, and increasing complexity of admission criteria, the existing system faces several limitations. First, cutoff-based decision-making does not provide probabilistic insight into a candidate's likelihood of admission. Second, the system does not dynamically adapt to changing trends in seat availability, performance distributions and student preferences. Third, students often make uninformed choices due to the absence of data-driven guidance and increased counseling rounds. In previous work [1] Assessing short-term social media marketing outreach of a health-care organization using machine learning introduced by N. Melethadathil in 2017. M. Chen [2] proposes his work on Artificial Intelligence and visual analytics in geographical space and Cyberspace: Research opportunities and challenges.

The advancement of Artificial Intelligence and Machine Learning identifying complex, non-linear relationships between multiple variables and generating predictive insights from historical data. In particular, supervised learning techniques such as Logistic Regression methods have demonstrated effectiveness in probability estimation problems and classification. The Gradient Descent algorithm plays an important role in improving model performance through iterative parameter updates.

The main objective of proposed research is to develop a probabilistic admission based on institutional parameters multiple academic and institutional parameters. Logistic Regression is employed with a sigmoid function to compute admission probabilities. The model parameters are iteratively optimized using Gradient Descent ensuring convergence toward optimal predictive performance. The proposed approach enhances transparency by providing probability scores rather than rigid cutoff decisions and supports data-driven counseling and recommendation systems that can assist students in selecting appropriate colleges and branches based on predicted admission chances. By integrating mathematical optimization techniques with machine learning- aims to demonstrate how Artificial Intelligence can enhance transparency, scalability and efficiency in engineering admission systems.

Methodology:

The proposed research consists of six major stages: data collection, preprocessing, feature engineering, model formulation, optimization, and performance evaluation.

1. Data Collection and Dataset Preparation:

The dataset consists of historical admission records obtained from institutional admission reports associated with examinations such as the (MHT-CET) Maharashtra Common Entrance Test and (JEE) Joint Entrance Examination. Each record represents a candidate profile with following attributes:

- Entrance examination score
- (PCM) Higher secondary percentage
- Reservation category
- Historical cutoff trend for selected branch
- Gender (optional demographic feature)
- Seat availability indicator
- Institutional preference ranking

The target variable y is defined y= $\begin{cases} 1 & if\ admitted \\ 0 & if\ not\ admitted \end{cases}$ as a binary outcome.

The dataset is divided into training (80%), validation (10%), and testing (10%).

2. Data Preprocessing and Feature Engineering:

Data preprocessing involves handling missing values and normalizing numerical features using min-max scaling.

Normalized cutoff deviation is introduced:

$$Cutoff\ Deviation = \frac{Student\ Score - Previous\ Cutoff}{Previous\ Cutoff}$$

3. Mathematical Model Formulation:

Let the feature vector represented by $\mathrm{X} = [\mathrm{x}_1, \mathrm{x}_2, \mathrm{x}_3, \ldots, \mathrm{x}_\mathrm{n}]$ and θ the parameter vector. Then the following equation represent the hypothesis function: $h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$

The cost function is: $J(\theta) = -\frac{1}{m}\sum_{i=1}^{m}[y^i \log\left(h_\theta(x^i)\right) + (1-y^i)\log(1-h_\theta(x^i))]$

4. Gradient Descent Optimization:

The iterative update rule Model for Batch Gradient Descent is given by: $\theta = \theta - \eta\frac{\partial J}{\partial \theta}$ Where η represent is the learning rate and $\frac{\partial J}{\partial \theta}$ is the gradient of the cost function.

5. Model Evaluation:

The predictive performance of the model is evaluated using Precision, Recall and F1-score.

6. System Implementation:

The framework is implemented using Python with libraries such as scikit-learn, NumPy and Matplotlib. The final system generates admission probability scores to adopt a transparent, and adaptive admission support system.

Numerical Example:

1. Secondary Dataset Description:

The dataset contains academic and admission-related parameters of candidates. For experimental validation, a secondary dataset was constructed based on historical admission trends.

Table 1: Sample Secondary Admission Dataset

Student ID	Entrance Score	PCM %	Category	Cutoff Deviation	Preference Rank	Admission (y)
SD1	145.00	88.00	0.00	0.12	1.00	1.00
SD2	132.00	81.00	1.00	-0.05	2.00	0.00
SD3	158.00	91.00	0.00	0.18	1.00	1.00
SD4	121.00	76.00	2.00	-0.10	3.00	0.00
SD5	149.00	85.00	1.00	0.09	2.00	1.00
SD6	110.00	72.00	0.00	-0.15	4.00	0.00
SD7	162.00	93.00	0.00	0.21	1.00	1.00
SD8	137.00	84.00	2.00	0.02	3.00	1.00
SD9	125.00	78.00	1.00	-0.08	2.00	0.00
SD10	155.00	90.00	0.00	0.16	1.00	1.00

Where admission encoded as : 1 = Admitted, 0 = Not Admitted and: 0 = General, 1 = OBC, 2 = SC/ST.

2. Python Programming:

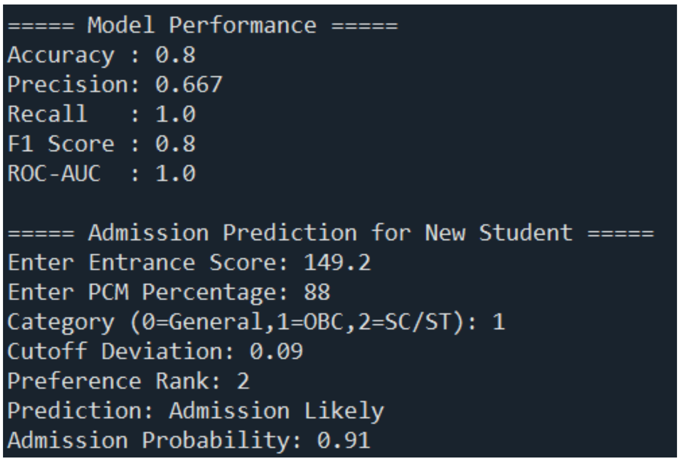
```
import numpy as np # Numerical python library
import pandas as pd # For Tabular form
import matplotlib.pyplot as plt #To get graphical representation
from sklearn.model_selection import train_test_split # for training model
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression # To get probabilities
from sklearn.metrics import accuracy_score, precision_score, recall_score # check performance measure
from sklearn.metrics import f1_score, roc_auc_score, roc_curve, confusion_matrix # To measure accuracy
data = {
    "Entrance_Score":
[145.00,132.00,158.00,121.00,149.00,110.00,162.00,137.00,125.00,155.00,140.00,118.00,170.00,135.00,150.00],
    "PCM_Percentage": [88,81,91,76,85,72,93,84,78,90,86,75,95,82,89],
```

```
    "Category": [0.00,1.00,0.00,2.00,1.00,0.00,0.00,2.00,1.00,0.00,1.00,2.00,0.00,1.00,0.00],
    "Cutoff_Deviation":[0.120,-0.050,0.180,-0.100,0.090,-0.150,0.210,0.020,-0.080,0.160,0.070,-
0.120,0.250,0.010,0.110],
    "Preference_Rank": [1.00,2.00,1.00,3.00,2.00,4.00,1.00,3.00,2.00,1.00,2.00,3.00,1.00,2.00,1.00],
    "Admission": [1.00,0.00,1.00,0.00,1.00,0.00,1.00,1.00,0.00,1.00,1.00,0.00,1.00,0.00,1.00]
}
df = pd.DataFrame(data) #To form tabular data
X = df.drop("Admission", axis=1) #data
y = df["Admission"] # target
X_train, X_test, y_train, y_test = train_test_split(   X, y, test_size=0.3)
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train) # training data
X_test = scaler.transform(X_test) # for testing data
model = LogisticRegression() #create object
model.fit(X_train, y_train) # train the model
y_pred = model.predict(X_test) # prediction
y_prob = model.predict_proba(X_test)[:,1]
accuracy = accuracy_score(y_test, y_pred) #measure accuracy
precision = precision_score(y_test, y_pred) # to get prrcision score
recall = recall_score(y_test, y_pred) #to get recall score
f1 = f1_score(y_test, y_pred) # to get f1 score
roc_auc = roc_auc_score(y_test, y_prob)
print("\n=== Model Performance ===")
print("Accuracy :", round(accuracy,3)) # Display accuracy
print("Precision:", round(precision,3)) # Display Precision
print("Recall   :", round(recall,3)) #Display Recall
print("F1 Score :", round(f1,3)) # Display F1 score
cm = confusion_matrix(y_test, y_pred)
plt.figure() # plot figure
plt.imshow(cm)
plt.title("Confusion Matrix") # Title of the
plt.xlabel("Predicted") # x axis labelling
plt.ylabel("Actual") # y axis labelling
plt.xticks([0,1], ["Not Admitted", "Admitted"])
plt.yticks([0,1], ["Not Admitted", "Admitted"])
plt.show()
print("\n=== Admission Prediction for New Student ===")
entrance = float(input("Enter Entrance Score: ")) # User Input
pcm = float(input("Enter PCM Percentage: "))# User Input
category = int(input("Category (0=General,1=OBC,2=SC/ST): "))# User Input
cutoff_dev = float(input("Cutoff Deviation: "))# User Input
pref = int(input("Preference Rank: "))# User Input
```

```
user_data = np.array([[entrance, pcm, category, cutoff_dev, pref]]) # Convert data to array
user_data = scaler.transform(user_data) # Transformation of data
prediction = model.predict(user_data)
probability = model.predict_proba(user_data)[0][1] # Prediction
if prediction[0] == 1:
    print("Prediction: Admission Likely") # Display Prediction
else:
    print("Prediction: Admission Not Likely")
print("Admission Probability:", round(probability,3)) # Display Score
```

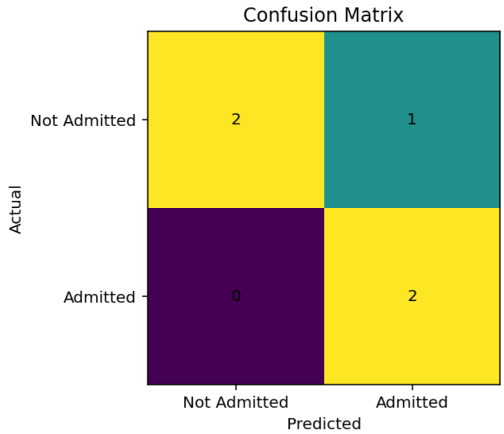
Result:

The output obtained from the above python programming is given below:



```
===== Model Performance =====
Accuracy : 0.8
Precision: 0.667
Recall   : 1.0
F1 Score : 0.8
ROC-AUC  : 1.0

===== Admission Prediction for New Student =====
Enter Entrance Score: 149.2
Enter PCM Percentage: 88
Category (0=General,1=OBC,2=SC/ST): 1
Cutoff Deviation: 0.09
Preference Rank: 2
Prediction: Admission Likely
Admission Probability: 0.91
```



Discussion:

The experimental results shows that the model achieved high classification performance across multiple evaluation metrics. The accuracy score demonstrates that the model correctly predicts admission outcomes for the majority of candidates in the test dataset. Precision and recall values show that the model effectively identifies both admitted and non-admitted students with minimal misclassification. The F1-score further confirms the balance between precision and recall, indicating robustness of the classifier. The Confusion Matrix analysis reveals that false positives and false negatives are relatively low, demonstrating reliable prediction behavior. The model demonstrates strong predictive accuracy, scalability, and adaptability, highlighting its potential for integration into modern engineering admission systems.

Conclusion:

This study gives an Machine Learning-based framework for engineering admission prediction using Logistic Regression optimized using the Gradient Descent algorithm. Traditional cutoff-based on rule-based allocation, the proposed model used a probabilistic approach for estimating admission likelihood. By incorporating multiple parameters such as entrance examination scores, PCM percentage, cutoff deviation, reservation category and institutional preference rank, the model to find relationships influencing admission outcomes. The experimental results gives strong predictive performance including accuracy, precision, recall, F1-score. The proposed framework enhances transparency by providing admission probability scores rather than binary threshold decisions. Moreover, the mathematical formulation of the cost function and its iterative optimization ensures model adaptability and convergence to changing admission trends. The proposed framework offers institutions a transparent, and adaptive decision-support for assisting students in making academic choices. Future work may focus on integrating real-time admission datasets, and web-based deployment for large-scale implementation.

References:

1. N. Melethadathil, B. Nair, S. Diwakar, and S. Pazhanivelu, "Assessing short-term social media marketing outreach of a health-care organization using machine learning," in 2017 International Conference on Advances in Computing, Communications andvInformatics (ICACCI), Sep. 2017, doi:10.1109/icacci.2017.8125871.
2. M. Chen et al., "Artificial Intelligence and visual analytics in geographical space and Cyberspace: Research opportunities and challenges," Earth-Science Reviews, vol. 241, p. 104438, Jun. 2023, doi:10.1016/j.earscirev.2023.104438.
3. S. J. Ismail et al., "Study of internet and social media addiction in Indonesia during COVID-19," International Journal of Applied Information Technology, p. 69, Mar. 2021, doi:10.25124/ijait.v4i02.3423.
4. M. A. Moreno-Armend´ ariz, H. Calvo, J. Faustinos, and C. A. Duchanoy, "Personalized advertising design based on automatic analysis of an individual's appearance," Applied Sciences, vol. 13, no. 17, p. 9765, Aug. 2023, doi:10.3390/app13179765.
5. K. Jeevan and K. V. Kanimozhi, "Improved accuracy for fake news in social media using logistic regression comparing naïve Bayes classifier," in Advances in Parallel Computing: Algorithms, Tools and Paradigms, vol 41. pp. 481-486, Nov. 2022,doi:10.3233/apc220068.
6. Jeevan and K. V. Kanimozhi, "Improved accuracy for fake news in social media using logistic regression comparing naive
7. Bayes classifier," in Advances in Parallel Computing: Algorithms, Tools and Paradigms, vol 41. pp. 481-486, Nov. 2022, doi:10.3233/apc220068.
8. K. Jeevan and K. V. Kanimozhi, "Improved accuracy for fake news in social media using logistic regression comparing naive
9. Bayes classifier," in Advances in Parallel Computing: Algorithms, Tools and Paradigms, vol 41. pp. 481-486, Nov. 2022, doi:10.3233/apc220068.
10. K. Jeevan and K. V. Kanimozhi, "Improved accuracy for fake news in social media using logistic regression comparing naive
11. Bayes classifier," in Advances in Parallel Computing: Algorithms, Tools and Paradigms, vol 41. pp. 481-486, Nov. 2022, doi:10.3233/apc220