AUTOMATED ESSAY SCORING

Main project final presentation

Varsha U
S4 MCA
MAC22MCA-2028

Guide : Prof. Sonia Abraham

Introduction

Automated Essay Scoring (AES) addresses the need for fast , effective and affordable solutions to automate the grading of student essays. The overarching problem is to develop scoring algorithms that can accurately asses essays , mimicking human expert grader’s evaluations.

Literature review

Paper 1

Ramesh, Dadi, and Suresh Kumar Sanampudi. "An Improved Approach for Automated Essay Scoring with LSTM and Word Embedding." Evolution in Computational Intelligence: Proceedings of the 9th International Conference on Frontiers in Intelligent Computing: Theory and Applications (FICTA 2021). Singapore: Springer Nature Singapore, 2022.

Objective

Improve Automated Essay Scoring (AES) efficiency and reliability.

Algorithm

LSTM with Word2Vec.

Dataset

ASAP dataset from Kaggle.

Conclusion

Outperforms other models on Kaggle dataset.

Paper 2

Chimingyang, Huang. "An automatic system for essay questions scoring based on LSTM and word embedding." 2020 5th International Conference on Information Science, Computer Technology and Transportation (ISCTT). IEEE, 2020.

.

Objective

Develop an Automated Essay Scoring (AES) model for efficient evaluation of student responses.

Algorithm

Logistic Regression
LSTM

Dataset

ASAP dataset from Kaggle.

Conclusion

LSTM outperformed Logistic Regression.

Paper 3

Sharma, Shakshi, and Anjali Goyal. "Automated essay grading: An empirical analysis of ensemble learning techniques." *Computational Methods and Data Engineering: Proceedings of ICMDE 2020, Volume 2*. Singapore: Springer Singapore, 2020. 343-362.

Objective

Evaluate traditional machine learning and ensemble learning for essay grading.

Algorithms

Logistic Regression, kNN, Naive Bayes, Decision Tree, SVM, Random Forest, AdaBoost, Gradient Boosting, XGBoost

Dataset

Uses ASAP dataset from Kaggle.

Conclusion

Ensemble learning enhances automated essay grading

Insights of three studies

	Algorithms	Advantages	Disadvantages
Paper 1	**LSTM (QWK:85.35%)**	Outperform other neural network models on the Kaggle dataset.	The model extracts word-level features, potentially missing the semantic nuances of essays.
Paper 2	**Logistic Regression(QWK: 65%) LSTM(QWK: 95%)**	Demonstrate the effectiveness of deep learning for accurate essay scoring.	The model's interpretability might be challenging due to the complexity of LSTM architectures, making it less straightforward to understand the specific factors contributing to essay scores.
Paper 3	**Traditional ML: logistic regression-85% Ensemble technique : gradient boosting – 86.53%**	Ensemble learning improves automated essay grading accuracy and efficiency, surpassing traditional methods.	Still faces challenges related to subjectivity and requires ongoing research for further improvement.

Proposed Algorithms

We are looking at both traditional machine learning methods and neural networks and compare the results.

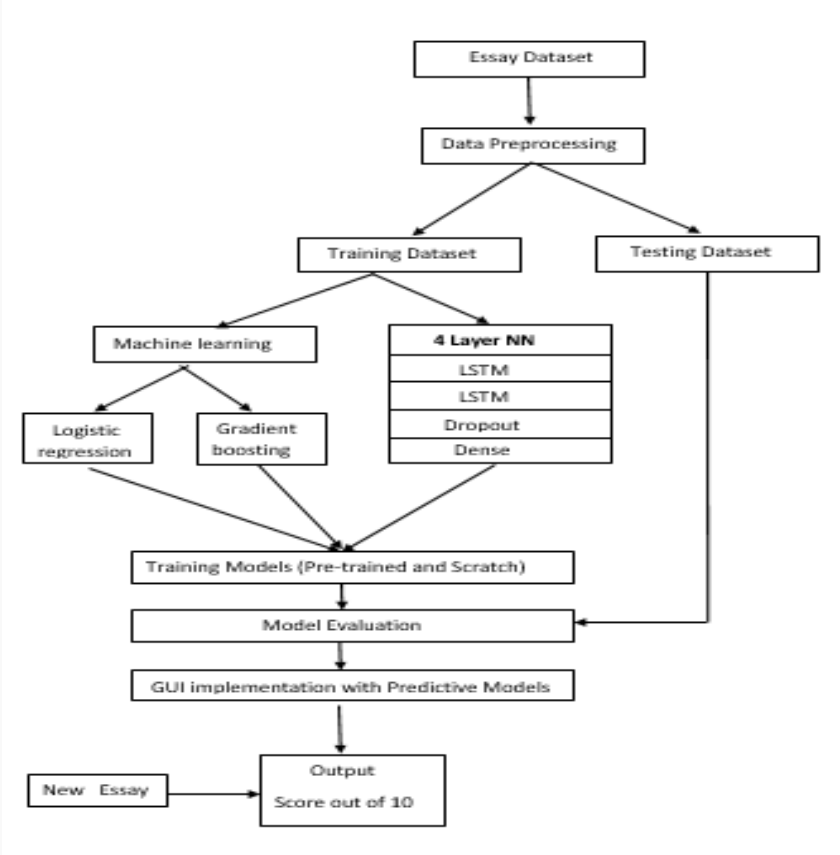
The models that will be using are:

Logistic regression

Gradient boosting

LSTM

Project pipeline



Dataset

Dataset: Automated Student Assessment Prize (ASAP) on Kaggle.
https://www.kaggle.com/c/asap-aes

Essays span 150-550 words

Authored by Grade 7 to Grade 10 students

hand-graded and double-scored

8 essay sets

12976 entries, 28 columns

size:120.15 MB ， TSV file

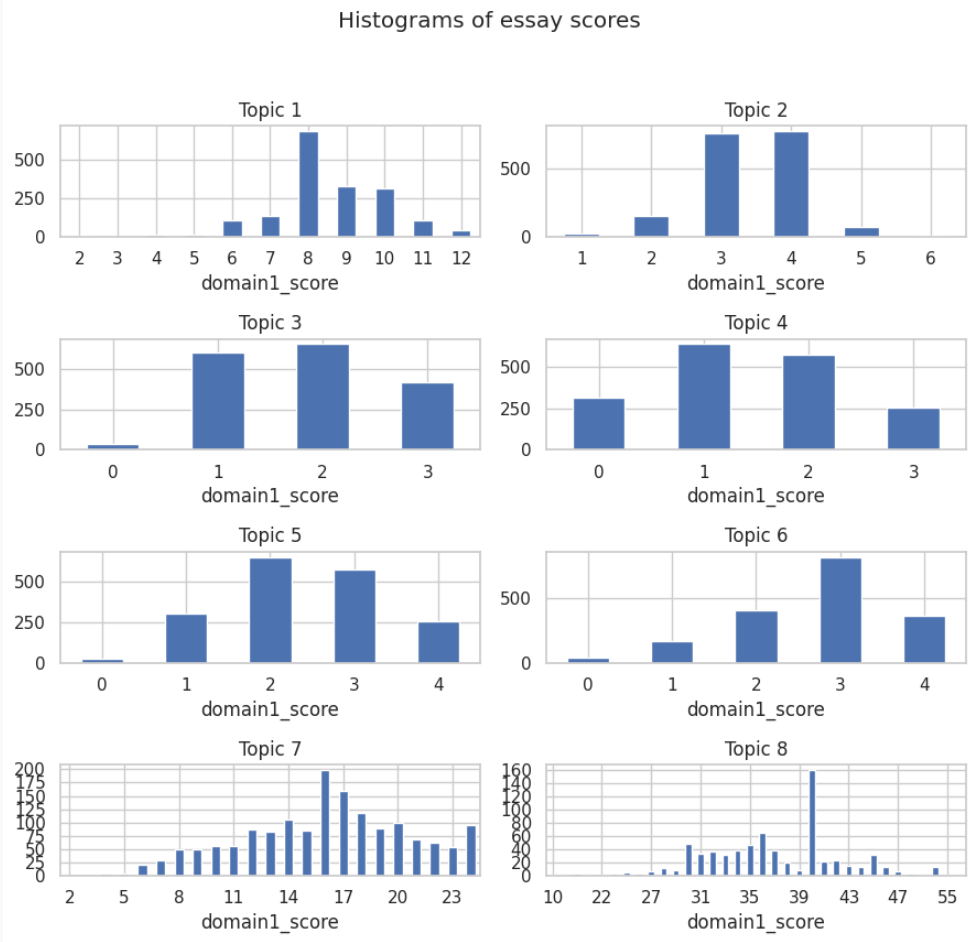
- **essay_id**: A unique identifier for each individual student essay
- **essay_set**: 1-8, an id for each set of essays
- **essay**: The ascii text of a student's response
- **rater1_domain1**: Rater 1's domain 1 score; all essays have this
- **rater2_domain1**: Rater 2's domain 1 score; all essays have this
- **rater3_domain1**: Rater 3's domain 1 score; only some essays in set 8 have this.
- **domain1_score**: Resolved score between the raters; all essays have this
- **rater1_domain2**: Rater 1's domain 2 score; only essays in set 2 have this
- **rater2_domain2**: Rater 2's domain 2 score; only essays in set 2 have this
- **domain2_score**: Resolved score between the raters; only essays in set 2 have this
- **rater1_trait1 score - rater3_trait6 score**: trait scores for sets 7-8

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12976 entries, 0 to 12975
Data columns (total 28 columns):
 #   Column          Non-Null Count  Dtype  
---  ------          --------------  -----  
 0   essay_id        12976 non-null  int64  
 1   essay_set       12976 non-null  int64  
 2   essay           12976 non-null  object 
 3   rater1_domain1  12976 non-null  int64  
 4   rater2_domain1  12976 non-null  int64  
 5   rater3_domain1  128 non-null    float64
 6   domain1_score   12976 non-null  int64  
 7   rater1_domain2  1800 non-null   float64
 8   rater2_domain2  1800 non-null   float64
 9   domain2_score   1800 non-null   float64
 10  rater1_trait1   2292 non-null   float64
 11  rater1_trait2   2292 non-null   float64
 12  rater1_trait3   2292 non-null   float64
 13  rater1_trait4   2292 non-null   float64
 14  rater1_trait5   723 non-null    float64
 15  rater1_trait6   723 non-null    float64
 16  rater2_trait1   2292 non-null   float64
 17  rater2_trait2   2292 non-null   float64
 18  rater2_trait3   2292 non-null   float64
 19  rater2_trait4   2292 non-null   float64
 20  rater2_trait5   723 non-null    float64
 21  rater2_trait6   723 non-null    float64
 22  rater3_trait1   128 non-null    float64
 23  rater3_trait2   128 non-null    float64
 24  rater3_trait3   128 non-null    float64
 25  rater3_trait4   128 non-null    float64
 26  rater3_trait5   128 non-null    float64
 27  rater3_trait6   128 non-null    float64
dtypes: float64(22), int64(5), object(1)
memory usage: 2.8+ MB
```

```
df.head()
```

	essay_id	essay_set	essay	domain1_score
0	1	1	Dear local newspaper, I think effects computer...	8
1	2	1	Dear @CAPS1 @CAPS2, I believe that using compu...	9
2	3	1	Dear, @CAPS1 @CAPS2 @CAPS3 More and more peopl...	7
3	4	1	Dear Local Newspaper, @CAPS1 I have found that...	10
4	5	1	Dear @LOCATION1, I know having computers has a...	8

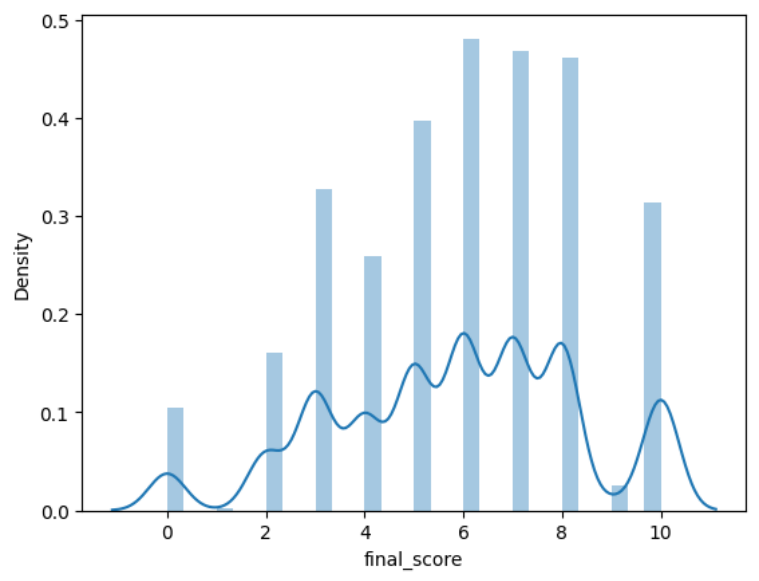
essay_set	count	min	max	nunique
1	1783	2	12	11
2	1800	1	6	6
3	1726	0	3	4
4	1770	0	3	4
5	1805	0	4	5
6	1800	0	4	5
7	1569	2	24	23
8	723	10	60	34



Data preprocessing

Normalization

```
min_range = [2,1,0,0,0,0,0,10]
max_range = [12,6,3,3,4,4,30,60]

def normalize(x,mi,ma):
    x = (x-mi)/(ma-mi)
    return round(x*10)

df['final_score']=df.apply(lambda x:normalize(x['domain1_score'],min_range[x['essay_set']-1],max_range[x['essay_set']-1]),axis=1)
```



Removal of stop words,punctuations

Before

"Dear local newspaper, I think effects computers have on people are great learning skills/affects because they give us time to chat with friends/new people, helps us learn about the globe(astronomy) and keeps us out of troble! Thing about! Dont you think so? How would you feel if your teenager is always on the phone with friends! Do you ever time to chat with your friends or buisness partner about things. Well now - there's a new way to chat the computer, theirs plenty of sites on the internet to do so: @ORGANIZATION1, @ORGANIZATION2, @CAPS1, facebook, myspace ect. Just think now while your setting up meeting with your boss on the computer, your teenager is having fun on the phone not rushing to get off cause you want to use it. How did you learn about other countrys/states outside of yours? Well I have by computer/internet, it's a new way to learn about what going on in our time! You might think your child spends a lot of time on the computer, but ask them so question about the economy, sea floor spreading or even about the @DATE1's you'll be surprise at how much he/she knows. Believe it or not the computer is much interesting then in class all day reading out of books. If your child is home on your computer or at a local library, it's better than being out with friends being fresh, or being perpressured to doing something they know isnt right. You might not know where your child is, @CAPS2 forbidde in a hospital bed because of a drive-by. Rather than your child on the computer learning, chatting or just playing games, safe and sound in your home or community place. Now I hope you have reached a point to understand and agree with me, because computers can have great effects on you or child because it gives us time to chat with friends/new people, helps us learn about the globe and believe or not keeps us out of troble. Thank you for listening."

After

'dear local newspaper i think effects computers have on people are great learning skillsaffects because they give us time to chat with friendsnew people helps us learn about the globeastronomy and keeps us out of troble thing about dont you think so how would you feel if your teenager is always on the phone with friends do you ever time to chat with your friends or buisness partner about things well now theres a new way to chat the computer theirs plenty of sites on the internet to do so facebook myspace ect just think now while your setting up meeting with your boss on the computer your teenager is having fun on the phone not rushing to get off cause you want to use it how did you learn about other countrysstates outside of yours well i have by computerinternet its a new way to learn about what going on in our time you might think your child spends a lot of time on the computer but ask them so question about the economy sea floor spreading or even about the youll be surprise at how much heshe knows believe it or not the computer is much interesting then in class all day reading out of books if your child is home on your computer or at a local library its better than being out with friends being fresh or being perpressured to doing something they know isnt right you might not know where your child is forbidde in a hospital bed because of a driveby rather than your child on the computer learning chatting or just playing games safe and sound in your home or community place now i hope you have reached a point to understand and agree with me because computers can have great effects on you or child because it gives us time to chat with friendsnew people helps us learn about the globe and believe or not keep

Dataset split

```
y = df['domain1_score']
df.drop('domain1_score',inplace=True,axis=1)
X=df
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

Dataset in vector form

```
training_vectors.shape
```

```
(9083, 300)
```

```
training_vectors
```

```
array([[-0.33153877,  0.12550558,  0.10141426, ..., -0.3191929 ,
         0.08542251, -0.04179949],
       [ 0.00291279,  0.04955867, -0.7039538 , ..., -0.1545468 ,
        -0.20332299,  0.34156355],
       [-0.24314483, -0.04799713, -0.2974147 , ..., -0.29020917,
        -0.16595323,  0.25235656],
       ...,
       [-0.04950241,  0.21167101, -0.571376  , ..., -0.38394287,
        -0.14878643,  0.22840413],
       [-0.10984093,  0.0235496 ,  0.1152441 , ...,  0.04483803,
        -0.04337973,  0.25489426],
       [-0.04279533,  0.04706671, -0.31105715, ..., -0.21455787,
        -0.1289501 ,  0.41291294]], dtype=float32)
```

```
testing_vectors.shape
```

```
(3893, 300)
```

Training and Testing

Logistic regression

Training

```
from sklearn.linear_model import LogisticRegression

# Train multinomial logistic regression model
log_reg = LogisticRegression(solver='lbfgs', penalty='l2', max_iter=100, C=1)
log_reg.fit(training_vectors.reshape(training_vectors.shape[0], -1), y_train)
```

Testing

```
# Predict on the test set
y_pred_logistic = logistic_regressor.predict(testing_vectors.reshape(
    testing_vectors.shape[0], -1))

# Calculate evaluation metrics
mse_logistic = mean_squared_error(y_test, y_pred_logistic)
mae_logistic = mean_absolute_error(y_test, y_pred_logistic)
rmse_logistic = mean_squared_error(y_test, y_pred_logistic, squared=False)
r2_logistic = r2_score(y_test, y_pred_logistic)
qwk_logistic = cohen_kappa_score(y_test, y_pred_logistic, weights='quadratic')
```

```
print("Logistic Regression Metrics:")
print("Mean Squared Error (MSE):", mse_logistic)
print("Mean Absolute Error (MAE):", mae_logistic)
print("Root Mean Squared Error (RMSE):", rmse_logistic)
print("R-squared (R2) Score:", r2_logistic)
print("Quadratic Weighted Kappa (QWK) Score:", qwk_logistic)
print("\n")
```

```
Logistic Regression Metrics:
Mean Squared Error (MSE): 4.877839254513686
Mean Absolute Error (MAE): 1.279994175888177
Root Mean Squared Error (RMSE): 2.2085830875277677
R-squared (R2) Score: 0.5082061026654981
Quadratic Weighted Kappa (QWK) Score: 0.8215152396282389
```

Gradient boosting

Training

```
from sklearn.ensemble import GradientBoostingRegressor

# Train Gradient Boosting Regression model
gb_reg = GradientBoostingRegressor(n_estimators=100, learning_rate=0.1,
                                   max_depth=5, min_samples_split=2,
                                   min_samples_leaf=1, random_state=0)
gb_reg.fit(training_vectors.reshape(training_vectors.shape[0], -1), y_train)
```

Testing

```
# Predict on the test set
y_pred_gb = gb_regressor.predict(testing_vectors.reshape(
    testing_vectors.shape[0], -1))

# Calculate evaluation metrics
mse_gb = mean_squared_error(y_test, y_pred_gb)
mae_gb = mean_absolute_error(y_test, y_pred_gb)
r2_gb = r2_score(y_test, y_pred_gb)
qwk_gb = cohen_kappa_score(y_test, np.round(y_pred_gb), weights='quadratic')
```

```
print("Gradient Boosting Metrics:")
print("Mean Squared Error (MSE):", mse_gb)
print("Mean Absolute Error (MAE):", mae_gb)
print("Root Mean Squared Error (RMSE):", rmse_gb)
print("R-squared (R2) Score:", r2_gb)
print("Quadratic Weighted Kappa (QWK) Score:", qwk_gb)
print("\n")
```

```
Gradient Boosting Metrics:
Mean Squared Error (MSE): 2.7730420481474662
Mean Absolute Error (MAE): 1.264358030731615
Root Mean Squared Error (RMSE): 1.67547491581083
R-squared (R2) Score: 0.7204161340353851
Quadratic Weighted Kappa (QWK) Score: 0.885626795350687
```

LSTM

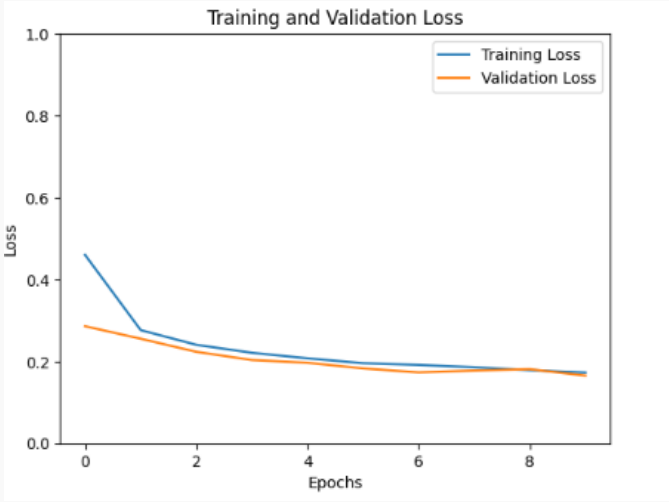
Training

```
def get_model():
    model = Sequential()
    model.add(LSTM(300, dropout=0.4, recurrent_dropout=0.4,
                   input_shape=[1, 300], return_sequences=True))
    model.add(LSTM(64, recurrent_dropout=0.4))
    model.add(Dropout(0.5))
    model.add(Dense(1, activation='relu'))
    model.compile(loss='mean_squared_error', optimizer='rmsprop',metrics=['mae'])
    model.summary()
    return model

lstm_model = get_model()

history = lstm_model.fit(training_vectors, y_train,batch_size=64,epochs=10,validation_split=0.2)
```

Layer (type)	Output Shape	Param #
lstm_1 (LSTM)	(None, 1, 300)	721200
lstm_2 (LSTM)	(None, 64)	93440
dropout_1 (Dropout)	(None, 64)	0
dense_1 (Dense)	(None, 1)	65

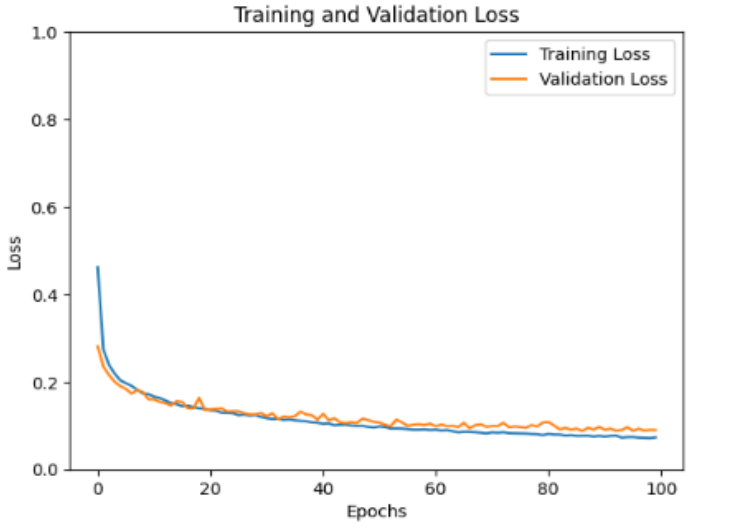
10 EPOCHS


```
Epoch 1/10
201/201 [==============================] - 25s 65ms/step - loss: 0.4626 - mae: 2.5165 - val_loss: 0.2852 - val_mae: 1.9343
Epoch 2/10
201/201 [==============================] - 10s 49ms/step - loss: 0.2733 - mae: 1.9439 - val_loss: 0.2484 - val_mae: 1.6956
Epoch 3/10
201/201 [==============================] - 9s 43ms/step - loss: 0.2393 - mae: 1.7957 - val_loss: 0.2153 - val_mae: 1.6038
Epoch 4/10
201/201 [==============================] - 5s 24ms/step - loss: 0.2188 - mae: 1.7069 - val_loss: 0.2069 - val_mae: 1.5535
Epoch 5/10
201/201 [==============================] - 6s 32ms/step - loss: 0.2093 - mae: 1.6618 - val_loss: 0.1958 - val_mae: 1.4957
Epoch 6/10
201/201 [==============================] - 5s 24ms/step - loss: 0.2001 - mae: 1.6413 - val_loss: 0.1897 - val_mae: 1.4753
Epoch 7/10
201/201 [==============================] - 6s 29ms/step - loss: 0.1893 - mae: 1.5887 - val_loss: 0.2076 - val_mae: 1.5441
Epoch 8/10
201/201 [==============================] - 6s 28ms/step - loss: 0.1806 - mae: 1.5549 - val_loss: 0.1687 - val_mae: 1.3897
Epoch 9/10
201/201 [==============================] - 5s 25ms/step - loss: 0.1787 - mae: 1.5576 - val_loss: 0.1666 - val_mae: 1.3714
Epoch 10/10
201/201 [==============================] - 7s 33ms/step - loss: 0.1707 - mae: 1.5216 - val_loss: 0.1674 - val_mae: 1.3697
```



100 EPOCHS

```
Epoch 95/100
201/201 [==============================] - 5s 25ms/step - loss: 0.0739 - mae: 1.0860 - val_loss: 0.0964 - val_mae: 1.0761
Epoch 96/100
201/201 [==============================] - 7s 33ms/step - loss: 0.0741 - mae: 1.0846 - val_loss: 0.0883 - val_mae: 1.0450
Epoch 97/100
201/201 [==============================] - 5s 25ms/step - loss: 0.0727 - mae: 1.0782 - val_loss: 0.0931 - val_mae: 1.0636
Epoch 98/100
201/201 [==============================] - 6s 31ms/step - loss: 0.0723 - mae: 1.0895 - val_loss: 0.0889 - val_mae: 1.0547
Epoch 99/100
201/201 [==============================] - 5s 26ms/step - loss: 0.0717 - mae: 1.0863 - val_loss: 0.0904 - val_mae: 1.0627
Epoch 100/100
201/201 [==============================] - 5s 25ms/step - loss: 0.0732 - mae: 1.0798 - val_loss: 0.0901 - val_mae: 1.0535
```



Testing

```
# Predict on the test set
y_pred_lstm = lstm_model.predict(testing_vectors)

# Calculate evaluation metrics
mse_lstm = mean_squared_error(y_test, y_pred_lstm)
mae_lstm = mean_absolute_error(y_test, y_pred_lstm)
rmse_lstm = mean_squared_error(y_test, y_pred_lstm, squared=False)
r2_lstm = r2_score(y_test, y_pred_lstm)
qwk_lstm = cohen_kappa_score(y_test, np.round(y_pred_lstm), weights='quadratic')
```

```
print("LSTM Metrics:")
print("Mean Squared Error (MSE):", mse_lstm)
print("Mean Absolute Error (MAE):", mae_lstm)
print("Root Mean Squared Error (RMSE):", rmse_lstm)
print("R-squared (R2) Score:", r2_lstm)
print("Quadratic Weighted Kappa (QWK) Score:", qwk_lstm)
print("\n")
```

```
LSTM Metrics:
Mean Squared Error (MSE): 2.117104270251798
Mean Absolute Error (MAE): 0.9825118318694792
Root Mean Squared Error (RMSE): 1.4550272403813607
R-squared (R2) Score: 0.7865491448560553
Quadratic Weighted Kappa (QWK) Score: 0.945626795350687
```


Performance Metrics of Different Models
Logistic Regression
Gradient Boosting
LSTM
Values
0
1
2
3
4
5
MSE
MAE
RMSE
R2 Score
QWK
Metrics

Scratch Code Implemenation

Logistic Regression

```
lr = MultiClassLogisticRegression()
lr.fit(X,y, lr=0.0001)
```

```
pre = lr.predict_classes(X)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y, pre)
print('MSE:', mse)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y, pre)
print('MAE:', mae)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)
print('RMSE:', rmse)

# Calculate R2 Score
r2 = r2_score(y, pre)
print('R2 Score:', r2)

# Calculate Quadratic Weighted Kappa (QWK)
qwk_score = cohen_kappa_score(y, pre, weights='quadratic')
print("QWK Score:", qwk_score)
```

```
MSE: 5.383477188655981
MAE: 1.5393033292231812
RMSE: 2.3202321411134665
R2 Score: 0.114688493755564
QWK Score: 0.5600226361772777
```

Gradient Boosting

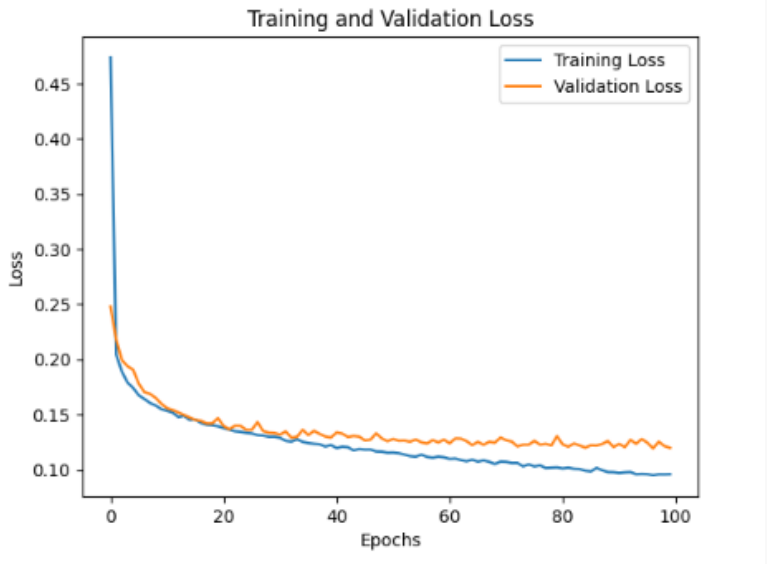
```
G = GradientBooster()
models, losses, pred_0 = G.train(X_train,y_train)
```

```
# Generate predictions using the model and test data
y_pred = G.predict(models, y_train, X_test)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print('MSE:', mse)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)
print('MAE:', mae)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)
print('RMSE:', rmse)

# Calculate R2 Score
r2 = r2_score(y_test, y_pred)
print('R2 Score:', r2)

# Calculate Quadratic Weighted Kappa (QWK)
def quadratic_weighted_kappa(y_true, y_pred):
    return cohen_kappa_score(y_true, y_pred, weights='quadratic')

qwk = quadratic_weighted_kappa(y_test, np.round(y_pred))
print('QWK:', qwk)
```

```
MSE: 2.635588152745999
MAE: 1.2610186988500363
RMSE: 1.6234494611000365
R2 Score: 0.5587778495660762
QWK: 0.7046731798700849
```

LSTM

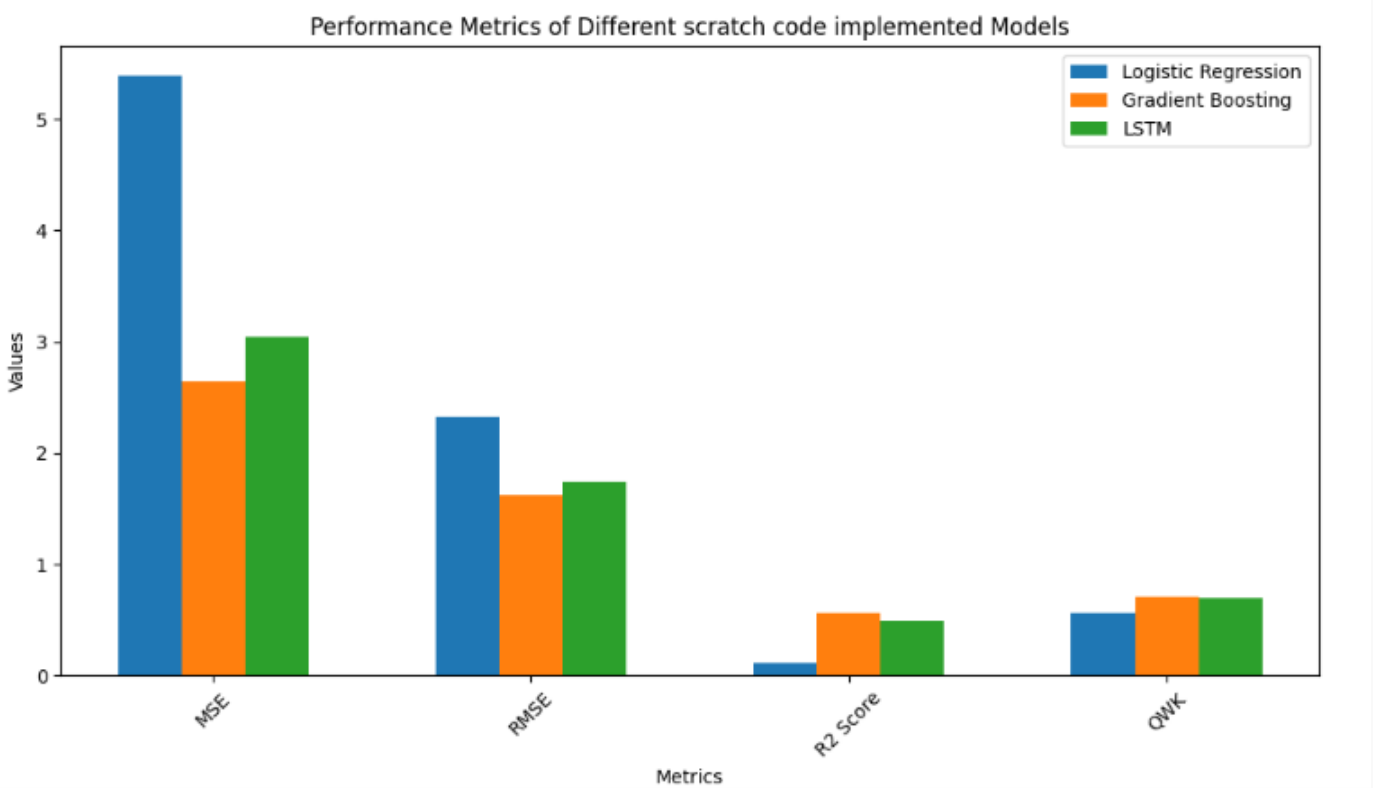
```
# Define the model
model = Sequential([
    custom_lstm_layer,
    CustomDropout(rate=0.2),
    CustomDense(units=1, activation=None)
])

# Compile the model
model.compile(optimizer='adam', loss='mean_squared_error')

# Train the model
history = model.fit(training_vectors, y_train, epochs=10, batch_size=64,
 validation_split=0.1, verbose=1)
```




```
Epoch 95/100
128/128 [==============================] - 1s 8ms/step - loss: 0.0962 - mae: 1.3322 - val_loss: 0.1278 - val_mae: 1.3394
Epoch 96/100
128/128 [==============================] - 1s 9ms/step - loss: 0.0960 - mae: 1.3279 - val_loss: 0.1245 - val_mae: 1.3795
Epoch 97/100
128/128 [==============================] - 1s 9ms/step - loss: 0.0951 - mae: 1.3249 - val_loss: 0.1193 - val_mae: 1.3655
Epoch 98/100
128/128 [==============================] - 1s 6ms/step - loss: 0.0958 - mae: 1.3330 - val_loss: 0.1255 - val_mae: 1.3628
Epoch 99/100
128/128 [==============================] - 1s 6ms/step - loss: 0.0958 - mae: 1.3317 - val_loss: 0.1215 - val_mae: 1.3923
Epoch 100/100
128/128 [==============================] - 1s 6ms/step - loss: 0.0960 - mae: 1.3311 - val_loss: 0.1199 - val_mae: 1.3833
```

```python
# Predict on the test set
y_pred1 = model.predict(testing_vectors)

# Round predictions to the nearest integer
y_pred = np.rint(y_pred).astype(int)

# Calculate Mean Squared Error (MSE)
mse = mean_squared_error(y_test, y_pred)
print('MSE:', mse)

# Calculate Mean Absolute Error (MAE)
mae = mean_absolute_error(y_test, y_pred)
print('MAE:', mae)

# Calculate Root Mean Squared Error (RMSE)
rmse = np.sqrt(mse)
print('RMSE:', rmse)

# Calculate R2 Score
r2 = r2_score(y_test, y_pred)
print('R2 Score:', r2)

# Calculate Quadratic Weighted Kappa (QWK) score
qwk_score = cohen_kappa_score(y_test, y_pred_rounded, weights='quadratic')
print("QWK Score:", qwk_score)
```

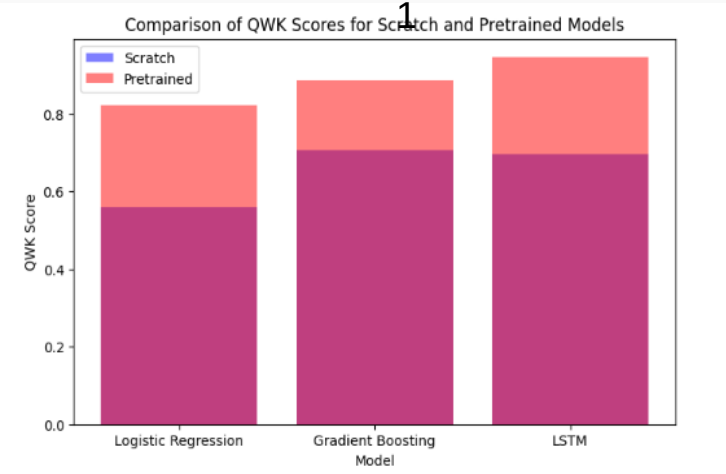
```
122/122 [==============================] - 0s 3ms/step
MSE: 3.041356280503468
MAE: 2369219634434826.5
RMSE: 1.743948474153829
R2 Score: 0.4941787192642465
QWK Score: 0.6962957421604512
```


Performance Metrics of Different scratch code implemented Models
Logistic Regression
Gradient Boosting
LSTM
Values
0
1
2
3
4
5
MSE
RMSE
R2 Score
QWK
Metrics

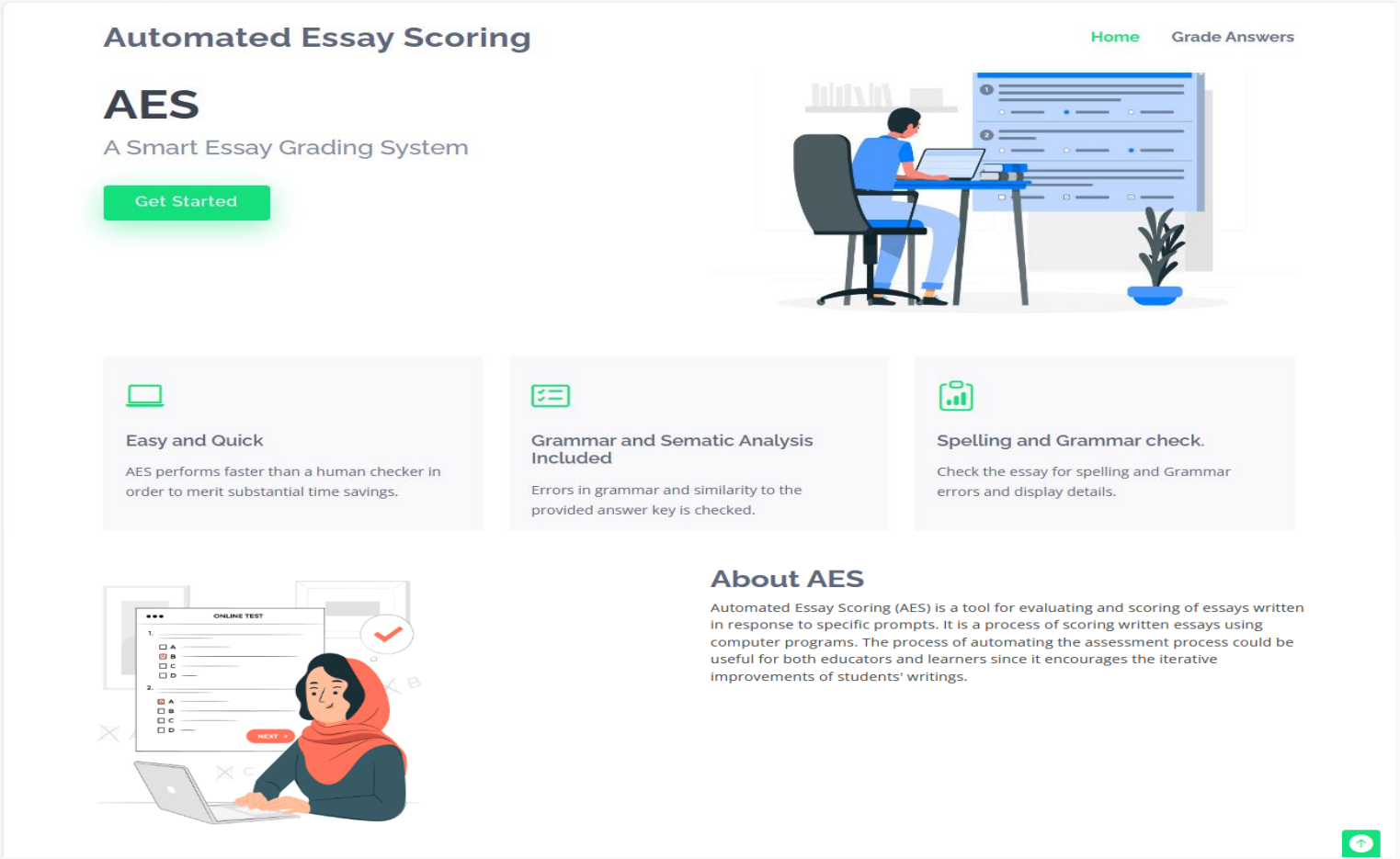
Result

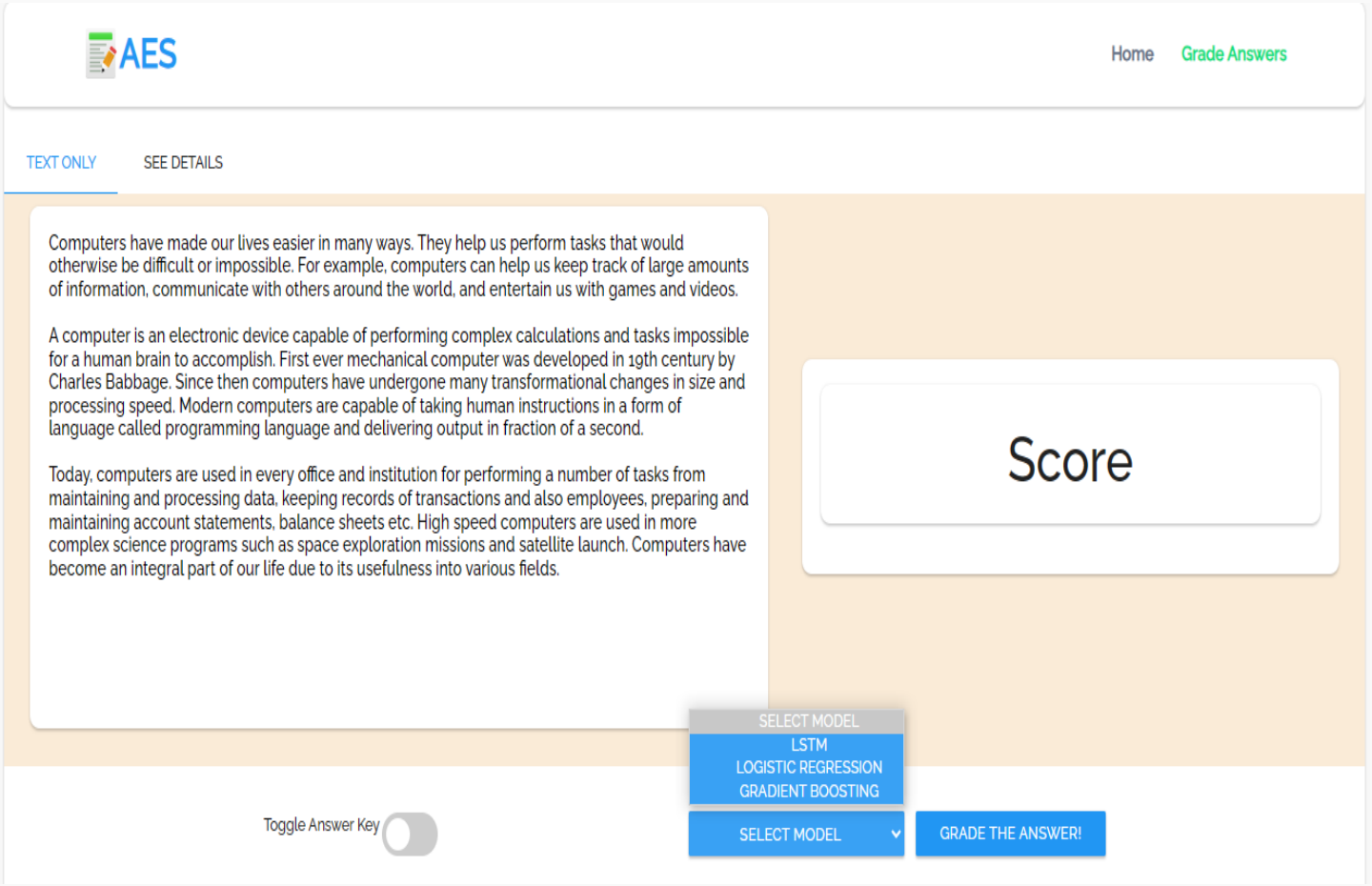
Quadratic Weighted Kappa (QWK) measures the agreement between two outcomes. We can interpret QWK as the amount of agreement between an algorithm's predictions and true labels.

- -1 : Complete disagreement
- 0 : Agreement by chance
- 0-0.2 ; Poor agreement
- 0.2-0.4 : Moderate agreement
- 0.4-0.6 : Good agreement
- 0.6-0.8 : Very good agreement
- 0.8-1 : Perfect agreement
- 1 : Complete agreement



Interface


Automated Essay Scoring
Home
Grade Answers
AES
A Smart Essay Grading System
Get Started
Easy and Quick
AES performs faster than a human checker in order to merit substantial time savings.
Grammar and Sematic Analysis Included
Errors in grammar and similarity to the provided answer key is checked.
Spelling and Grammar check.
Check the essay for spelling and Grammar errors and display details.
ONLINE TEST
NEXT
About AES
Automated Essay Scoring (AES) is a tool for evaluating and scoring of essays written in response to specific prompts. It is a process of scoring written essays using computer programs. The process of automating the assessment process could be useful for both educators and learners since it encourages the iterative improvements of students' writings.


AES
Home
Grade Answers
TEXT ONLY
SEE DETAILS
Computers have made our lives easier in many ways. They help us perform tasks that would otherwise be difficult or impossible. For example, computers can help us keep track of large amounts of information, communicate with others around the world, and entertain us with games and videos.
A computer is an electronic device capable of performing complex calculations and tasks impossible for a human brain to accomplish. First ever mechanical computer was developed in 19th century by Charles Babbage. Since then computers have undergone many transformational changes in size and processing speed. Modern computers are capable of taking human instructions in a form of language called programming language and delivering output in fraction of a second.
Today, computers are used in every office and institution for performing a number of tasks from maintaining and processing data, keeping records of transactions and also employees, preparing and maintaining account statements, balance sheets etc. High speed computers are used in more complex science programs such as space exploration missions and satellite launch. Computers have become an integral part of our life due to its usefulness into various fields.
Score
SELECT MODEL
LSTM
LOGISTIC REGRESSION
GRADIENT BOOSTING
Toggle Answer Key
SELECT MODEL
GRADE THE ANSWER!

AES

Home Grade Answers

TEXT ONLY SEE DETAILS

Sentence Length Info:

Word Count: 222

Sentence Count: 12

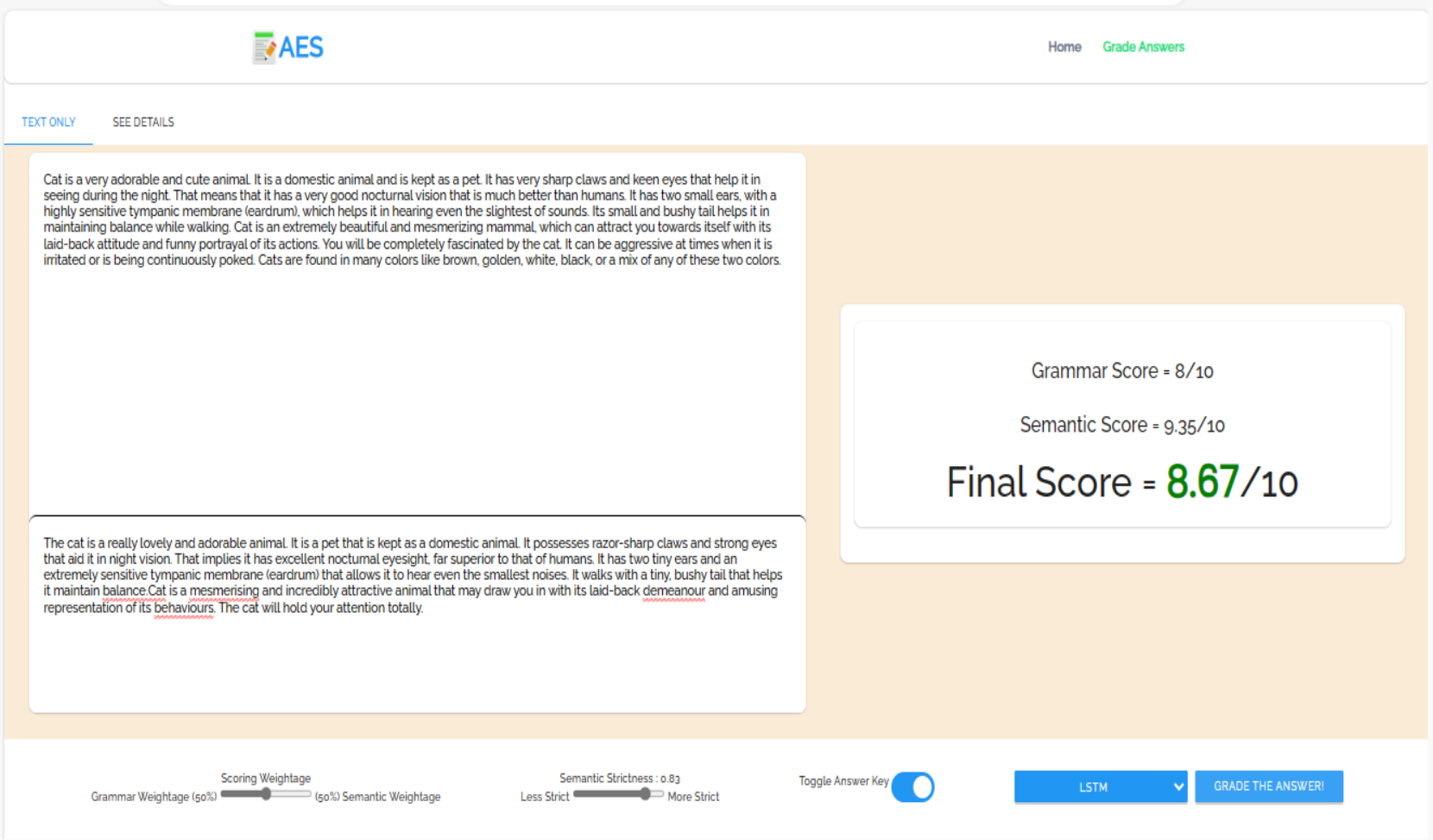
Paragraph Count: 1

Average Sentence Length: 18.5

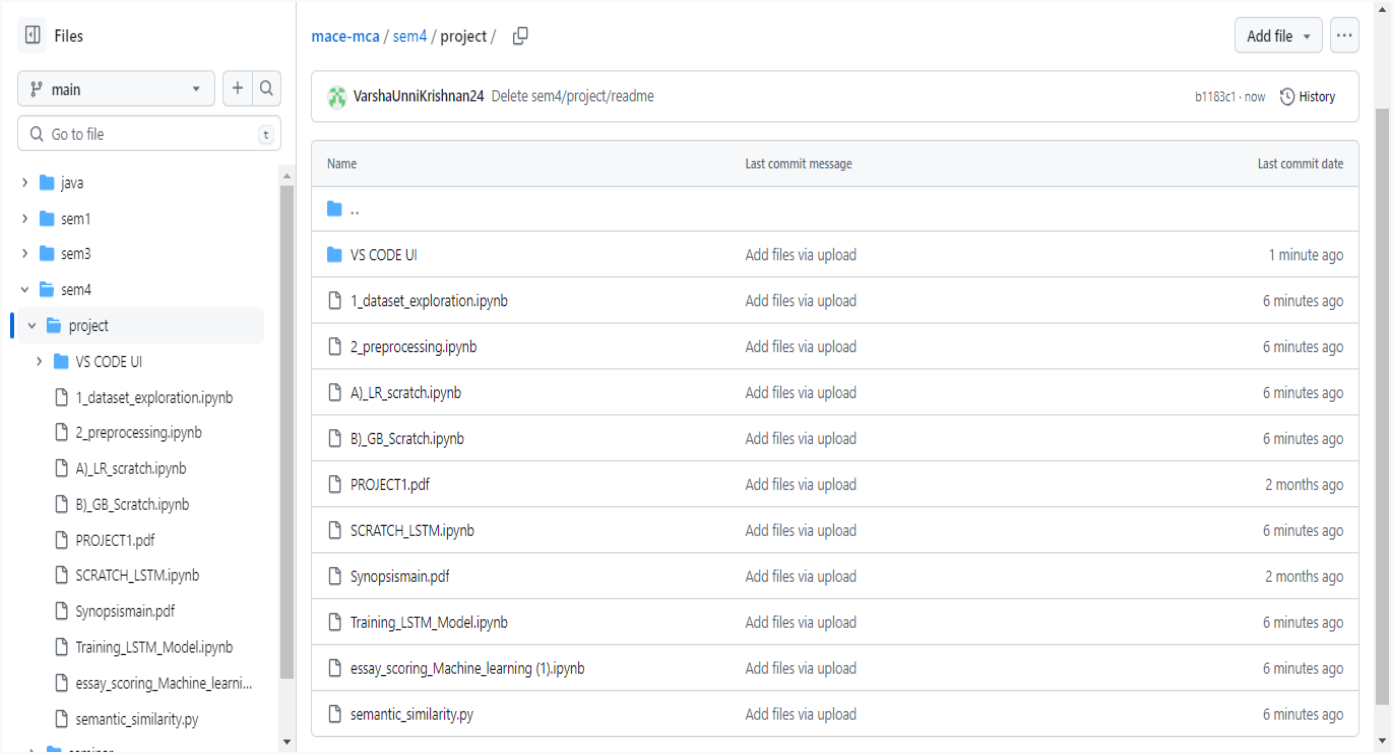
Grammar And Spell Check:

benfit, to, and, dosen't, rideing, biks, sur, awalk, cant, hase, bacik, thats, sur, ther, sur, sur, surton, aps, windos, microsoft, ect, dont, relize, payings, arnt, sars, lik, dont, ptonole, learnd, arnt, Coose, knolage

"Please note that while this web app strives for the best results, it may not be perfect.
Avoid testing the model with random inputs, as this could lead to inaccurate predictions.
Use genuine essays for proper grading outcomes.
Ensure that each essay contains a minimum of 5 points to provide sufficient material for analysis."


AES
Home
Grade Answers
TEXT ONLY
SEE DETAILS
Cat is a very adorable and cute animal. It is a domestic animal and is kept as a pet. It has very sharp claws and keen eyes that help it in seeing during the night. That means that it has a very good nocturnal vision that is much better than humans. It has two small ears, with a highly sensitive tympanic membrane (eardrum), which helps it in hearing even the slightest of sounds. Its small and bushy tail helps it in maintaining balance while walking. Cat is an extremely beautiful and mesmerizing mammal, which can attract you towards itself with its laid-back attitude and funny portrayal of its actions. You will be completely fascinated by the cat. It can be aggressive at times when it is irritated or is being continuously poked. Cats are found in many colors like brown, golden, white, black, or a mix of any of these two colors.
The cat is a really lovely and adorable animal. It is a pet that is kept as a domestic animal. It possesses razor-sharp claws and strong eyes that aid it in night vision. That implies it has excellent nocturnal eyesight, far superior to that of humans. It has two tiny ears and an extremely sensitive tympanic membrane (eardrum) that allows it to hear even the smallest noises. It walks with a tiny, bushy tail that helps it maintain balance.Cat is a mesmerising and incredibly attractive animal that may draw you in with its laid-back demeanour and amusing representation of its behaviours. The cat will hold your attention totally.
Grammar Score = 8/10
Semantic Score = 9.35/10
Final Score = 8.67/10
Scoring Weightage
Grammar Weightage (50%)
(50%) Semantic Weightage
Semantic Strictness : 0.83
Less Strict
More Strict
Toggle Answer Key
LSTM
GRADE THE ANSWER!

Git history


Files
main
Go to file
java
sem1
sem3
sem4
project
VS CODE UI
1_dataset_exploration.ipynb
2_preprocessing.ipynb
A)_LR_scratch.ipynb
B)_GB_Scratch.ipynb
PROJECT1.pdf
SCRATCH_LSTM.ipynb
Synopsismain.pdf
Training_LSTM_Model.ipynb
essay_scoring_Machine_learni...
semantic_similarity.py
mace-mca / sem4 / project /
Add file
VarshaUnniKrishnan24 Delete sem4/project/readme
b1183c1 · now
History
Name
Last commit message
Last commit date
..
VS CODE UI
Add files via upload
1 minute ago
1_dataset_exploration.ipynb
Add files via upload
6 minutes ago
2_preprocessing.ipynb
Add files via upload
6 minutes ago
A)_LR_scratch.ipynb
Add files via upload
6 minutes ago
B)_GB_Scratch.ipynb
Add files via upload
6 minutes ago
PROJECT1.pdf
Add files via upload
2 months ago
SCRATCH_LSTM.ipynb
Add files via upload
6 minutes ago
Synopsismain.pdf
Add files via upload
2 months ago
Training_LSTM_Model.ipynb
Add files via upload
6 minutes ago
essay_scoring_Machine_learning (1).ipynb
Add files via upload
6 minutes ago
semantic_similarity.py
Add files via upload
6 minutes ago

Project implementation plan:

- System Study : 18/01/2024
- Literature Review: 20/01/2024
- Dataset Preprocessing: 29/01/2024
- Dataset Exploration: 22/01/2024
- Detailed Study of Algorithm and Architecture: 04/02/2024
- Build 3 Pre defined Architectures: 15/02/2024
- Initial Presentation Verification:19/02/2024
- Initial Presentation:20/02/2024
- Detailed Study of Suggestions: 23/02/2024
- Implementation of scratch code: 11/03/2024
- GUI Implementation: 25/03/2024
- Documenation : 15/04/2024

Resources

- https://www.kaggle.com/c/asap-aes/data
- https://ieeexplore.ieee.org/abstract/document/9363782/
- https://link.springer.com/chapter/10.1007/978-981-16-6616-2_4
- https://link.springer.com/chapter/10.1007/978-981-15-7907-3_26

Thank you