|  |  |  |
| --- | --- | --- |
| **پرسش ۱** | **نام و نام خانوادگی** | سیدرضا مسلمی |
| **شماره دانشجویی** | 810103326 |
| **پرسش ۲** | **نام و نام خانوادگی** | بی‌بی رقیه |
| **شماره دانشجویی** | 810102053 |
|  | **مهلت ارسال پاسخ** |  |

|  |  |  |
| --- | --- | --- |
|  | **به نام خدا**  **دانشگاه تهران**  **دانشکده‌ مهندسی برق و کامپیوتر** |  |
| **درس شبکه‌های عصبی و یادگیری عمیق**  **تمرین اول** | | |

**فهرست**

[**قوانین** 1](#_Toc185885162)

[**پرسش 1**. **عنوان پرسش اول به فارسی** 1](#_Toc185885163)

[۱-۱. مجموعه داده 1](#_Toc185885164)

[2-۱. پیش‌پردازش داده‌ها 2](#_Toc185885165)

[3-۱. نمایش ویژگی 4](#_Toc185885166)

[4-۱. ساخت مدل 7](#_Toc185885167)

[۵-۱. ارزیابی 13](#_Toc185885168)

[6-۱. امتیازی 15](#_Toc185885169)

[**پرسش ۲** **- پیشبینی ارزش نفت** 20](#_Toc185885170)

[۱-۲. **مقدمه** 20](#_Toc185885171)

[۲-۲. **مجموعه دادگان و آماده سازي** 20](#_Toc185885172)

[۳-۲. **پیاده سازي مدل ها** 22](#_Toc185885173)

[۴-۲. **ARIMA** 25](#_Toc185885174)

**شکل‌ها**

**No table of figures entries found.**

**جدول‌ها**

[جدول 1: جدول پرفورمنس LSTM, GRU, Bi-LSTM 22](#_Toc185885175)

# **قوانین**

قبل از پاسخ دادن به پرسش‌ها،‌ موارد زیر را با دقت مطالعه نمایید:

* از پاسخ‌های خود یک گزارش در قالبی که در صفحه‌ی درس در سامانه‌ی Elearn با نام ***REPORTS\_TEMPLATE.docx*** قرار داده شده تهیه نمایید.
* پیشنهاد می‌شود تمرین‌ها را در قالب گروه‌های دو نفره انجام دهید. (بیش از دو نفر مجاز نیست و تحویل تک نفره نیز نمره‌ی اضافی ندارد) توجه نمایید الزامی در یکسان ماندن اعضای گروه تا انتهای ترم وجود ندارد. (یعنی، می‌توانید تمرین اول را با شخص A و تمرین‌ دوم را با شخص B و ... انجام دهید)
* **کیفیت گزارش شما در فرآيند تصحيح از اهميت ويژه­اي برخوردار است**؛ بنابراین، لطفا تمامی نکات و فرض­هایی را كه در پیاده­سازی­ها و محاسبات خود در نظر مي­گيريد در گزارش ذکر کنید.
* در گزارش خود مطابق با آنچه در قالب نمونه قرار داده شده، برای شکل‌ها زیرنویس و برای جدول‌ها بالانویس در نظر بگیرید.
* الزامی به ارائه توضیح جزئیات کد در گزارش نیست، اما باید نتایج بدست آمده از آن را گزارش و تحلیل کنید.
* **تحلیل نتایج الزامی می‌باشد، حتی اگر در صورت پرسش اشاره‌ای به آن نشده باشد.**
* **دستیاران آموزشی ملزم به اجرا کردن کدهای شما نیستند**؛ بنابراین، هرگونه نتیجه و یا تحلیلی که در صورت پرسش از شما خواسته شده را به طور واضح و کامل در گزارش بیاورید. در صورت عدم رعایت این مورد، بدیهی است که از نمره تمرین کسر می­شود.
* **کدها حتما باید در قالب نوت‌بوک با پسوند .ipynb تهیه شوند، در پایان کار، تمامی کد اجرا شود و خروجی هر سلول حتما در این فایل ارسالی شما ذخیره شده باشد.** بنابراین برای مثال اگر خروجی سلولی یک نمودار است که در گزارش آورده‌اید، این نمودار باید هم در گزارش هم در نوت‌بوک کد‌ها وجود داشته باشد.
* **در صورت مشاهده‌ی تقلب امتیاز تمامی افراد شرکت­کننده در آن، 100- لحاظ می­شود.**
* تنها زبان برنامه نویسی مجاز **Python** است.
* **استفاده از کدهای آماده برای تمرین­ها به­ هیچ ­وجه مجاز نیست. در صورتی که دو گروه از یک منبع مشترک استفاده کنند و کدهای مشابه تحویل دهند، تقلب محسوب می‌شود.**
* نحوه محاسبه­ تاخیر به این شکل است: پس از پایان رسیدن مهلت ارسال گزارش، حداکثر تا یک هفته امکان ارسال با تاخیر وجود دارد، پس از این یک هفته نمره آن تکلیف برای شما صفر خواهد شد.
  + سه روز اول: بدون جریمه
  + روز چهارم: ۵ درصد
  + روز پنجم: ۱۰ درصد
  + روز ششم: ۱۵ درصد
  + روز هفتم: ۲۰ درصد
* حداکثر نمره‌ای که برای هر سوال می‌توان اخد کرد ۱۰۰ بوده و اگر مجموع بارم یک **سوال** بیشتر از ۱۰۰ باشد، در صورت اخد نمره بیشتر از ۱۰۰، اعمال نخواهد شد.
  + برای مثال: اگر نمره اخذ شده از سوال ۱ برابر ۱۰۵ و نمره سوال ۲ برابر ۹۵ باشد، نمره نهایی تمرین ۹۷.۵ خواهد بود و نه ۱۰۰.
* لطفا گزارش، کدها و سایر ضمایم را به در یک پوشه با نام زیر قرار داده و آن را فشرده سازید، سپس در سامانه‌ی Elearn بارگذاری نمایید:

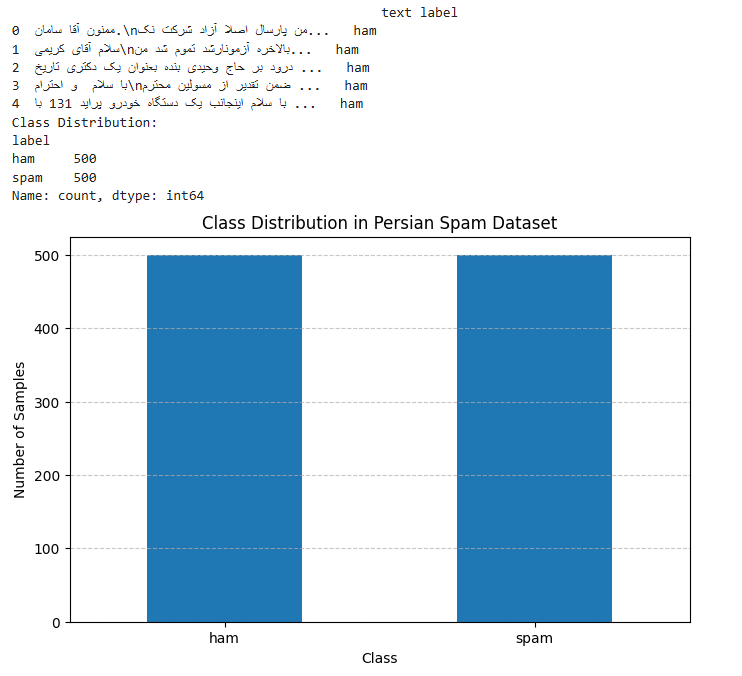
HW[Number] \_[Lastname]\_[StudentNumber]\_[Lastname]\_[StudentNumber].zip

(مثال: HW1\_Ahmadi\_810199101\_Bagheri\_810199102.zip)

* برای گروه‌های دو نفره، بارگذاری تمرین از جانب یکی از اعضا کافی است ولی پیشنهاد می‌شود هر دو نفر بارگذاری نمایند.

# **پرسش 1**. **عنوان پرسش اول به فارسی**

۱-۱. مجموعه داده



2-۱. پیش‌پردازش داده‌ها

Preprocessing text data is a critical step in Natural Language Processing (NLP) tasks. The goal of this exercise is to clean and normalize text data by removing unnecessary noise to ensure effective model performance. The following preprocessing steps were applied:

**Steps Taken:**

1. Normalization:

* The text was normalized using the Hazm library to standardize it for further processing.

1. Removing URLs:

* Any links or URLs (e.g., http://) were removed since they do not contribute to textual analysis.

1. Removing Email Addresses:

* Email addresses were eliminated to prevent bias from unrelated information.

1. Removing Phone Numbers:

* Long numbers and phone numbers were identified and excluded.

1. Reducing Repeated Characters:

* Consecutive repeated characters were replaced with a single instance (e.g., "عااااالی" → "عالی").

1. Tokenization:

* The text was broken down into individual tokens (words) using Hazm's word\_tokenize.

1. Stopword Removal:

* Stopwords (e.g., "و", "به") were filtered out using Hazm's stopword list.

1. Rejoining Tokens:

* The cleaned tokens were reassembled into a single text.



3-۱. نمایش ویژگی

The aim of this task is to convert preprocessed Persian text into numerical embeddings using ParsBERT, reduce the dimensionality of the embeddings, and prepare the data for further analysis.

**Steps Taken:**

1. Tokenization and Padding:

* The text was tokenized using the AutoTokenizer from the Hugging Face library.
* Padding was applied to ensure all tokenized sequences have the same length (e.g., 32 tokens).

1. Generating Embeddings:

* ParsBERT (HooshvareLab/bert-base-parsbert-uncased) was loaded, and its pre-trained model was used to compute embeddings.
* The embedding for each sentence was extracted using the CLS token representation, which serves as a sentence-level embedding.

1. Dimensionality Reduction:

* The embeddings (of size 768 per sentence) were reduced to 120 dimensions using Principal Component Analysis (PCA) for efficiency and compatibility with downstream tasks.

1. Integration:

* The reduced embeddings were added to the DataFrame, ensuring compatibility with machine learning models.



سوال‌ها:

1. What is the size of the embedding vector in ParsBERT?

* The default embedding size in ParsBERT is 768 dimensions.

1. What does reducing the size of the embedding vector achieve?

* It improves computational efficiency and reduces memory usage. It also helps mitigate overfitting by discarding less informative dimensions.

1. What is the concept of embeddings, and which words in the dataset might have similar embeddings?

* Embeddings are dense vector representations of text that capture semantic and syntactic meaning.
* Similar words (in terms of meaning or context) are mapped to vectors that are close in the embedding space. (like دانشگاه and مدرسه)

4-۱. ساخت مدل

The task involved creating and training CNN-LSTM, CNN, and LSTM models using hyperparameter optimization for batch sizes, learning rates, and optimizers to achieve the best possible accuracy for binary classification.

Process

1. Dataset Preparation

* The dataset was split into training, validation, and test sets using a 70-20-10 split.
* Features (X) were extracted from embedding columns, and labels (y) were encoded using LabelEncoder.
* For the CNN-LSTM model, the input data was reshaped to include an additional dimension.

2. Model Architectures

* CNN-LSTM:

Combines a convolutional neural network for feature extraction and an LSTM layer for sequential data analysis.

* CNN:

Uses convolutional layers followed by fully connected layers for classification.

* LSTM:

Relies on recurrent connections in LSTM layers to capture sequential patterns in the data.

3. Hyperparameter Optimization

A greedy search algorithm was used to optimize the following hyperparameters:

* Batch Sizes: [8, 64]
* Learning Rates: [0.001, 0.0001]
* Optimizers: [Adam, SGD]

Models were trained using these combinations, and validation accuracy was used to select the best parameters.

4. Final Results

* Best Batch Size: 8 (Accuracy: 0.5571)
* Best Learning Rate: 0.001 (Accuracy: 0.5214)
* Best Optimizer: Adam (Accuracy: 0.5571)

1)



Greedy Search for Optimize Parameters:



2)

****

3) Training of the CNN and LSTM models with optimal hyperparameters:

* CNN:



* LSTM:



سوال‌ها:

1. What are the strengths and weaknesses of CNN and LSTM models?

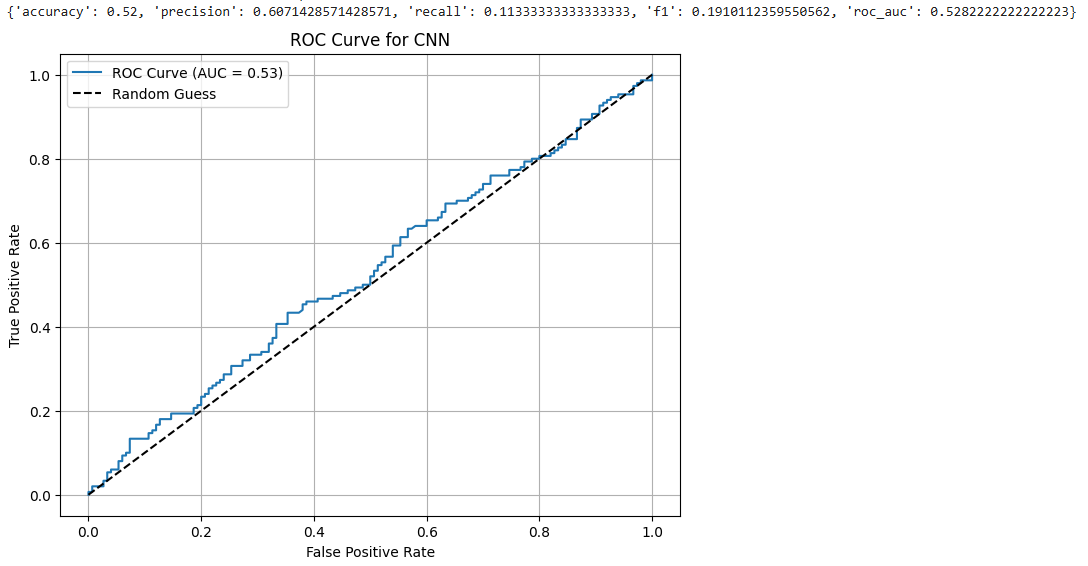
|  |  |  |
| --- | --- | --- |
| **Model** | **Strengths** | **Weaknesses** |
| **CNN** | Effective for spatial/feature extraction tasks (e.g., image, text embeddings). | Struggles with sequential dependencies or long-term patterns. |
| **LSTM** | Excellent at capturing temporal/sequential relationships. | Computationally intensive, sensitive to long sequences, and slower training. |

2. What is the purpose of combining CNN and LSTM in the model?

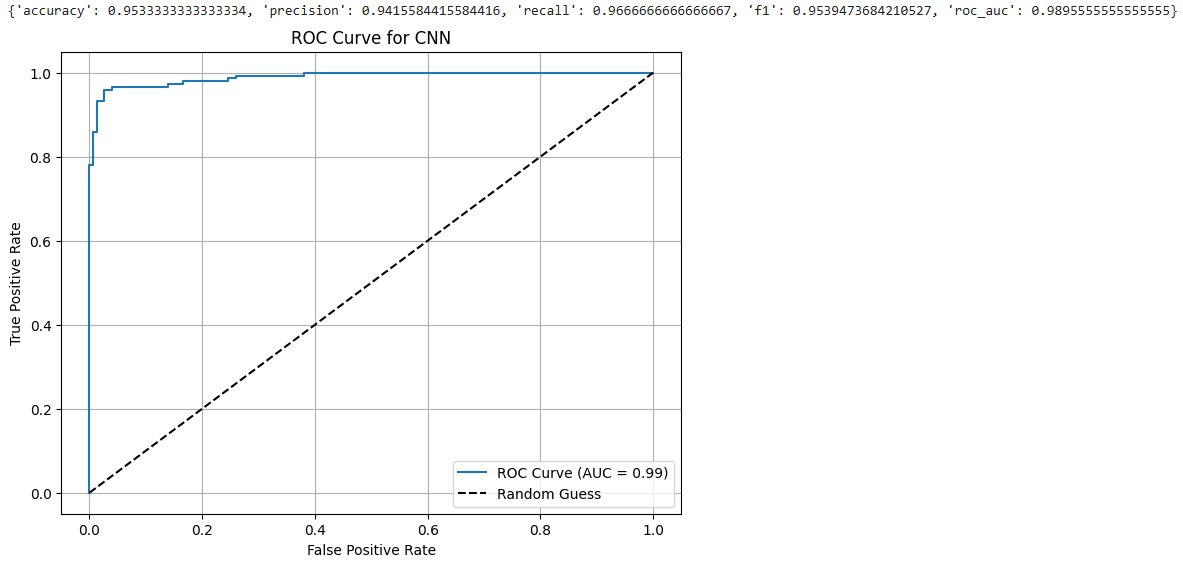
* Purpose:
  + CNN extracts key features from the input (e.g., spatial patterns in embeddings).
  + LSTM captures sequential dependencies from these extracted features.
* Outcome:
  + This combination leverages the strengths of both architectures, making it suitable for tasks where spatial and temporal relationships are important.

۵-۱. ارزیابی

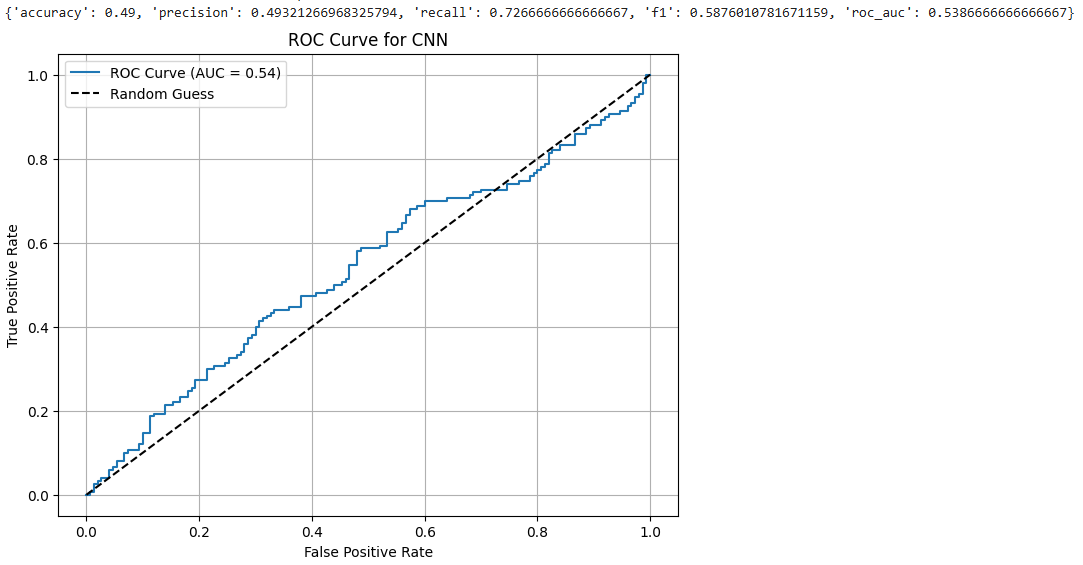
LSTM-CNN Results:



CNN Results:



LSTM Results:



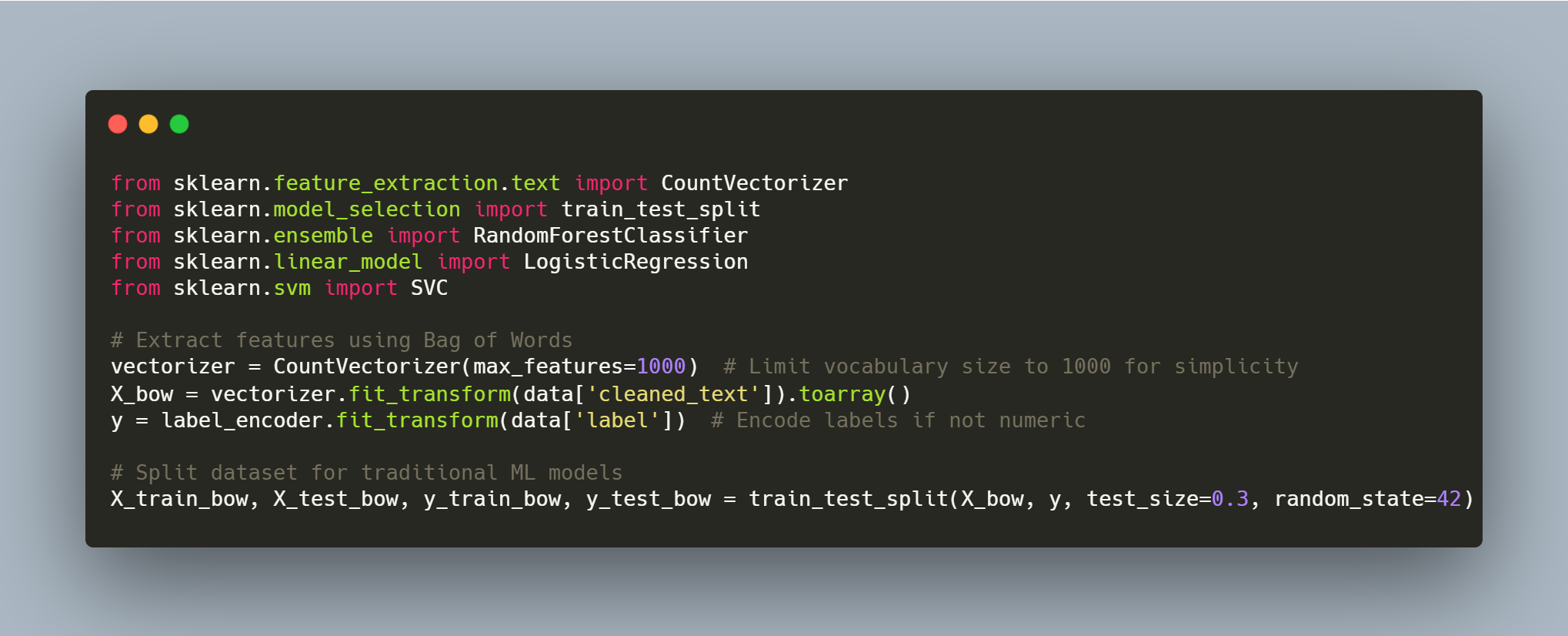
جدول1: مقایسه CNN، LSTM و CNN-LSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **ROC-AUC** |
| **CNN-LSTM** | 52.00% | 60.71% | 11.33% | 19.10% | 52.82% |
| **CNN** | 98.95% | 95.39% | 96.66% | 95.39% | 98.95% |
| **LSTM** | 49.00% | 49.32% | 72.67% | 58.76% | 53.87% |

6-۱. امتیازی

This report evaluates and compares the performance of traditional Machine Learning (ML) models and deep learning models (CNN, LSTM, CNN-LSTM) using a binary classification task. Features were extracted using the Bag of Words (BoW) method for ML models, while neural networks used embeddings and end-to-end training.

Use of Bag of Words for feature representation:

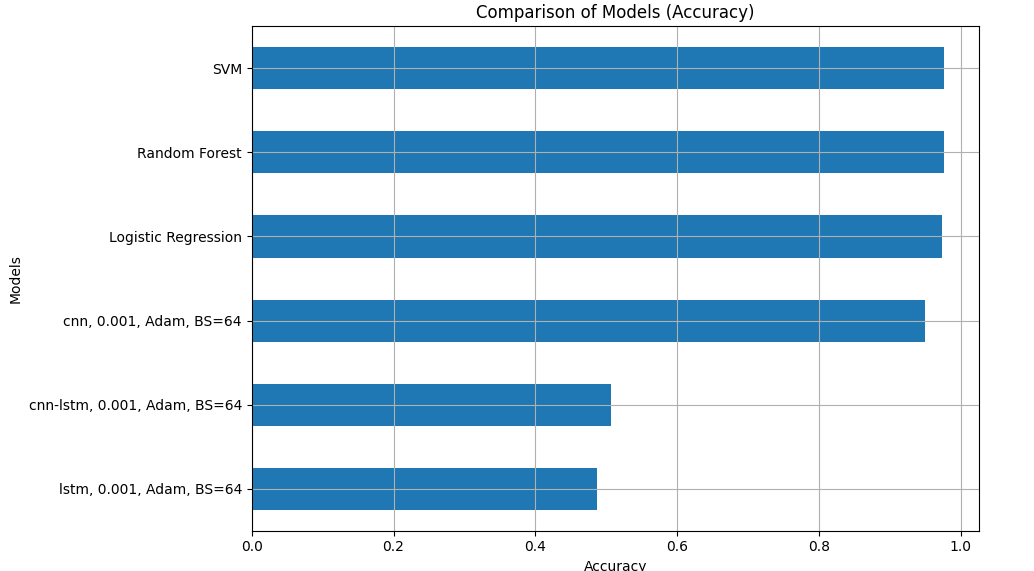


Results for 3 ML models:



جدول2: مقایسه CNN، LSTM و CNN-LSTM و سه مدل ML

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Accuracy** | **Precision** | **Recall** | **F1 Score** | **ROC-AUC** |
| **CNN-LSTM** | 52.00% | 60.71% | 11.33% | 19.10% | 52.82% |
| **CNN** | 98.95% | 95.39% | 96.66% | 95.39% | 98.95% |
| **LSTM** | 49.00% | 49.32% | 72.67% | 58.76% | 53.87% |
| **Logistic Regression** | 97.33% | 98.63% | 96.00% | 97.29% | 99.77% |
| **Random Forest** | 97.67% | 96.12% | 99.33% | 97.70% | 99.84% |
| **SVM** | 97.67% | 97.98% | 97.33% | 97.65% | 99.75% |



A graph with blue and white stripes

Description automatically generated

A graph with blue and white bars

Description automatically generated

A graph with blue and white stripes

Description automatically generated

A graph with blue and white bars

Description automatically generated

**Analysis**

1. Neural Network Models

* CNN-LSTM:
  + Achieved the lowest accuracy (52%) and recall (11.33%), indicating poor generalization and ineffective feature extraction for this task.
* CNN:
  + Outperformed other neural network models with a very high accuracy (98.95%) and balanced precision-recall performance.
* LSTM:
  + Demonstrated high recall (72.67%) but poor overall accuracy (49%), indicating that while it identifies positive cases well, it struggles with classification balance.

2. Traditional Machine Learning Models

* Logistic Regression:
  + Achieved competitive results with 97.33% accuracy and high ROC-AUC (99.77%), making it effective for linear separable data.
* Random Forest:
  + Delivered robust performance with 97.67% accuracy and the highest ROC-AUC (99.84%), showing its ability to model complex patterns.
* SVM:
  + Matched Random Forest in accuracy (97.67%) and nearly matched in ROC-AUC (99.75%), highlighting its effectiveness for text-based data.

**Conclusion**

1. Best Model:
   * Among neural networks, CNN provided the best performance, demonstrating its strength in feature extraction.
   * Among traditional models, Random Forest performed the best overall, slightly edging out SVM in ROC-AUC.
2. Key Takeaways:
   * Traditional ML models like Random Forest and SVM are highly effective for tasks using BoW features, offering competitive results with simpler architectures.
   * Neural networks require better tuning or different architectures (e.g., Transformers) for competitive performance in text classification tasks.

# **پرسش ۲** **- پیشبینی ارزش نفت**

## ۱-۲. **مقدمه**

## ۲-۲. **مجموعه دادگان و آماده سازي**

To analyze and prepare Crude Oil Futures data (CL=F) from Yahoo Finance for further modeling and analysis. Key tasks include handling missing values, normalizing data, and splitting the dataset into training and testing sets.

**Steps and Results**

1. Data Collection

* Source: Yahoo Finance
* Date Range: January 1, 2010, to December 31, 2023.
* Selected Column: Adj Close, which represents the adjusted closing price of crude oil.
* Initial Data Shape: (3,373, 1) (3,373 daily observations).

2. Handling Missing Values

* Introduced 10% missing values at random to simulate real-world scenarios.
* Missing values were filled using the mean imputation method, replacing NaN values with the column mean.

3. Normalization

* The MinMaxScaler was used to scale the Adj Close column to the range [0, 1], ensuring the values are normalized for effective modeling.

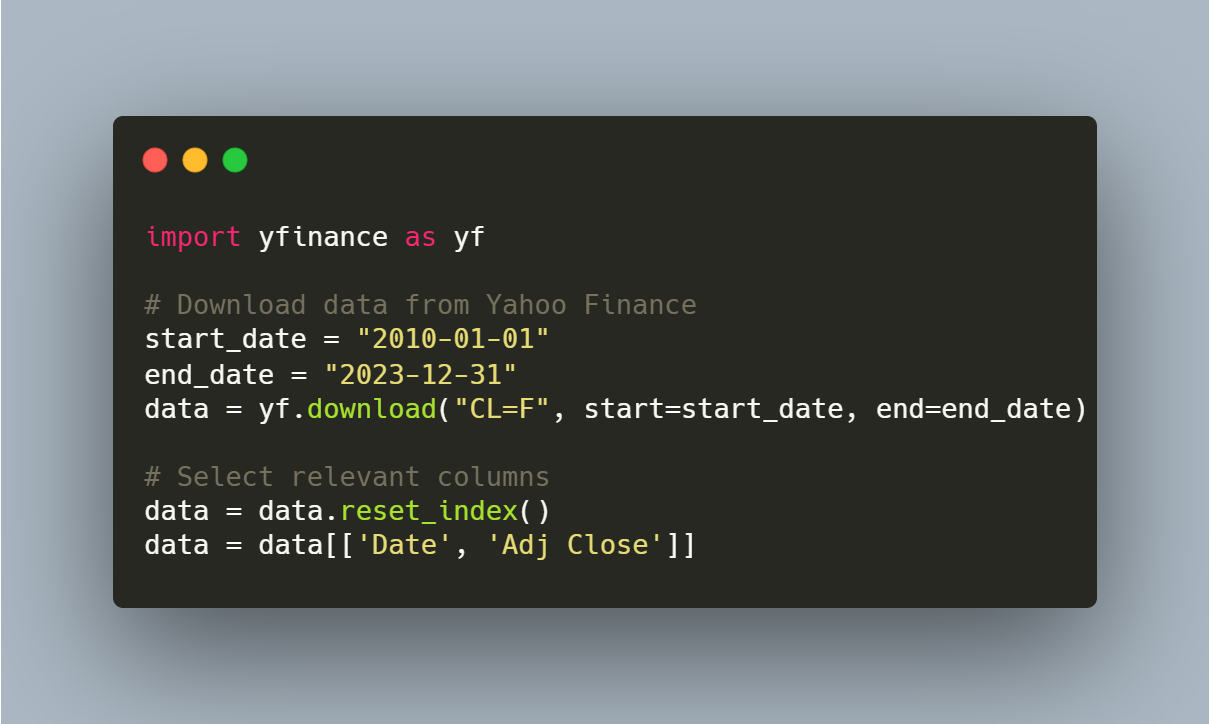
4. Dataset Splitting

* The normalized data was split into:
  + Training Set: 70% of the data.
  + Testing Set: 30% of the data.

5. Visualization

* A histogram was plotted to compare the distributions of the training and testing datasets.

Data Collection:



Handling Missing Values

A screen shot of a computer program

Description automatically generated

Normalization

A screenshot of a computer program

Description automatically generated

Dataset Splitting

A screenshot of a computer program

Description automatically generated

**Visualization**

A screen shot of a computer program

Description automatically generated

## ۳-۲. **پیاده سازي مدل ها**

The goal is to implement and evaluate three deep learning models — LSTM, GRU, and Bi-LSTM — for time-series forecasting of normalized crude oil prices. Models are trained using the Mean Squared Error (MSE) loss, and performance is evaluated using metrics like R-squared, MAE, RMSE, and MAPE.

**Results**

**Performance Metrics**

جدول 1: جدول پرفورمنس LSTM, GRU, Bi-LSTM

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **MSE** | **R-Squared** | **MAE** | **RMSE** | **MAPE** |
| **LSTM** | 0.0019 | 0.7055 | 0.0353 | 0.0440 | 4.7025% |
| **GRU** | 0.0013 | 0.8030 | 0.0224 | 0.0360 | 3.0241% |
| **Bi-LSTM** | 0.0015 | 0.7691 | 0.0253 | 0.0389 | 3.4771% |

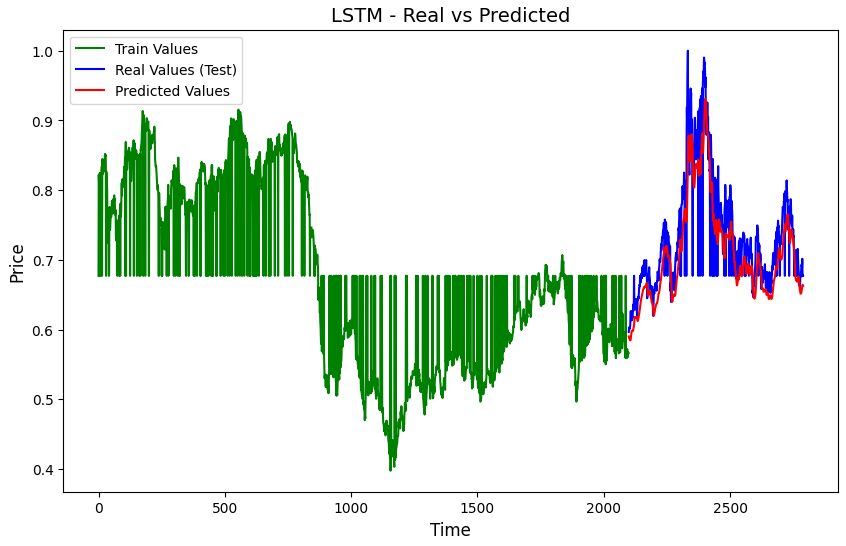
**Key Observations**

* GRU achieved the best overall performance with the lowest MSE (0.0013) and MAPE (3.0241%), as well as the highest R-squared (0.8030).
* LSTM showed weaker performance compared to GRU and Bi-LSTM, especially in MAE and MAPE, indicating less accurate predictions.
* Bi-LSTM performed better than LSTM but slightly underperformed compared to GRU.

**Model Predictions and Visualization**

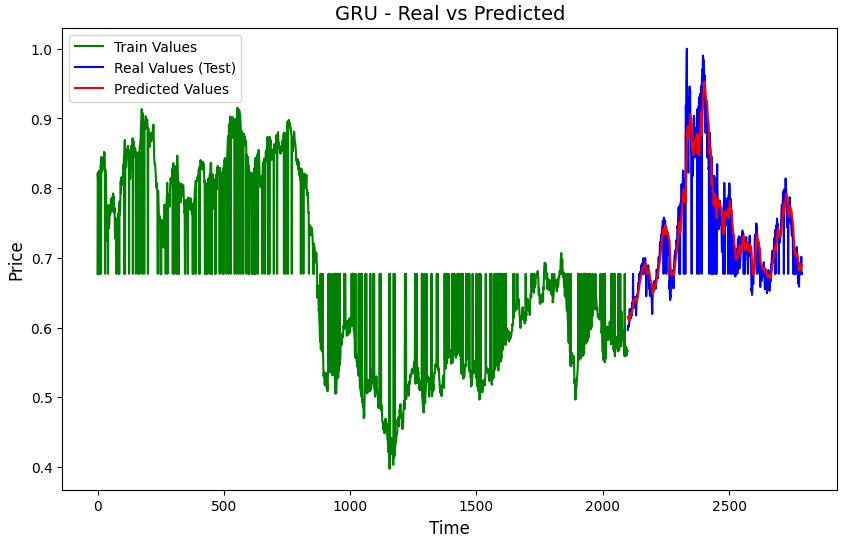
1. LSTM Predictions

* Trend: Captured general patterns but struggled with fine-grained details.
* Visualization: Predicted values closely align with real values but occasionally exhibit deviations.



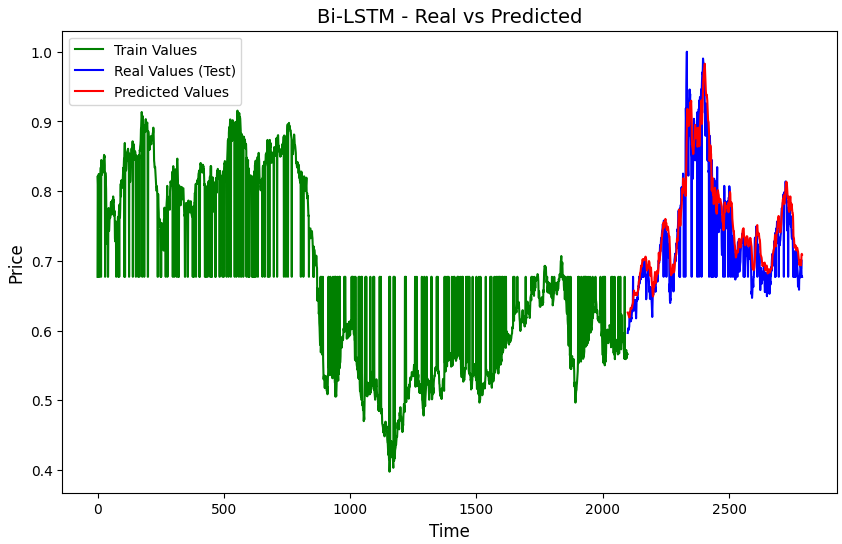
2. GRU Predictions

* Trend: GRU consistently followed the actual price trends, demonstrating superior adaptability and precision.
* Visualization: Predictions overlapped significantly with real values, indicating high accuracy.



3. Bi-LSTM Predictions

* Trend: Balanced performance, capturing trends well but with minor overfitting to training data.
* Visualization: Predictions were close to real values, with slightly larger deviations than GRU.



**MAE (Mean Absolute Error)**:

* Measures the average absolute difference between actual and predicted values.
* Lower values indicate better performance.

**RMSE (Root Mean Squared Error)**:

* Measures the square root of the average squared differences between actual and predicted values.
* More sensitive to large errors compared to MAE.

**MAPE (Mean Absolute Percentage Error)**:

* Measures the average percentage error relative to actual values.
* Useful for understanding prediction errors in percentage terms.

**R-Squared**:

* Explains the proportion of variance in the actual values captured by the model.
* Higher values (closer to 1) indicate better predictive power.

## ۴-۲. **ARIMA**

**Difference Between ARIMA and SARIMA Models:**

1. ARIMA (AutoRegressive Integrated Moving Average):
   * ARIMA is a model used for analyzing and forecasting time series data that consists of three main components:
     + AR (AutoRegressive): This component models the relationship between the current observation and previous observations.
     + I (Integrated): This component makes the time series stationary by differencing the data to eliminate trends.
     + MA (Moving Average): This component models the relationship between current values and the residual errors from previous periods.

Key Differences:

* + ARIMA is suitable for time series data that do not have seasonality or trends.
  + SARIMA, on the other hand, is an extension of ARIMA that can model seasonality (i.e., periodic patterns).

1. SARIMA (Seasonal ARIMA):
   * SARIMA is an extension of the ARIMA model that is designed for time series data that exhibits seasonality.
   * In addition to the AR, I, and MA components, SARIMA includes additional parameters to model seasonal effects.
   * SARIMA uses five additional parameters for seasonal modeling:
     + P (seasonal autoregression).
     + D (seasonal differencing).
     + Q (seasonal moving average).
     + S (the period of seasonality).

Key Differences:

* + ARIMA works best for data without seasonality, while SARIMA is specifically designed for data with seasonal patterns, such as sales data that fluctuates with seasons or monthly patterns in economic indicators.

Advantages and Limitations of the ARIMA Model:

Advantages:

1. Simplicity: ARIMA is simple to understand and implement, making it accessible for a wide range of users.
2. Forecasting Capability: It is effective in forecasting stationary time series data with no seasonal patterns.
3. Wide Application: ARIMA is widely used in financial, economic, and product demand forecasting.
4. Nonlinear Modeling: It can model some nonlinear relationships with appropriate parameter settings.

Limitations:

1. Stationarity Requirement: ARIMA requires that the data be stationary, meaning that its statistical properties (mean, variance) do not change over time. For non-stationary data, differencing must be applied, which may lead to loss of information.
2. No Seasonal Handling: ARIMA cannot handle seasonality in data unless modified (e.g., using SARIMA).
3. Complex Patterns: For more complex time series with non-linearities or intricate patterns, ARIMA may not provide accurate predictions.
4. Parameter Tuning: Finding the optimal parameters (p, d, q) can be time-consuming and requires trial and error.

Conclusion: ARIMA is suitable for stationary, non-seasonal data, but for data with seasonality or non-stationary trends, models like SARIMA or more advanced approaches should be used.

The ARIMA model (AutoRegressive Integrated Moving Average) is commonly used for forecasting and analyzing time series data. It consists of a combination of three main components: AutoRegressive (AR), Integrated (I), and Moving Average (MA).

**Mathematical Concept of ARIMA Model:**

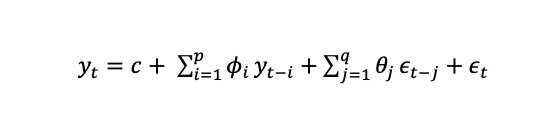
The ARIMA model has three main parameters, denoted as ppp, ddd, and qqq:

1. **AutoRegressive (AR)** 
   * This parameter represents the number of lag periods in the model, which are used to model the relationship between the current value and its past values.



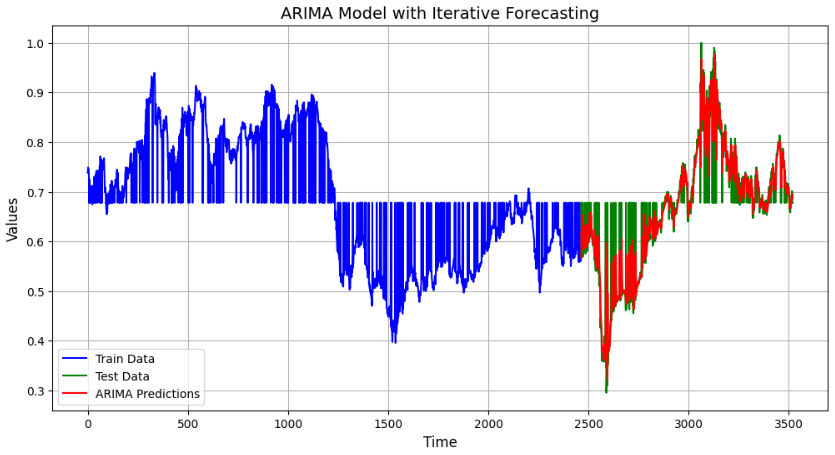
1. **Moving Average (MA)**
   * This parameter represents the number of lag periods used to model the relationship between the current value and the past error terms.
   * 

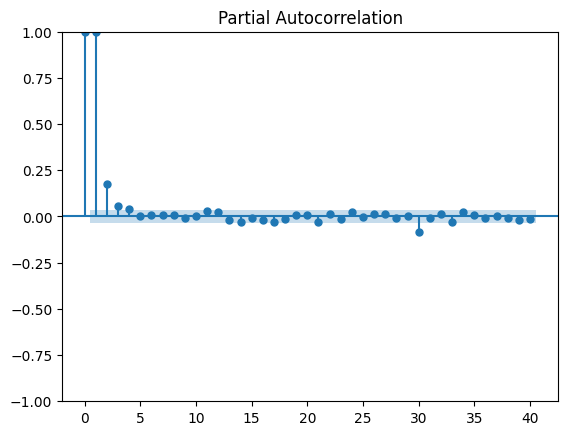
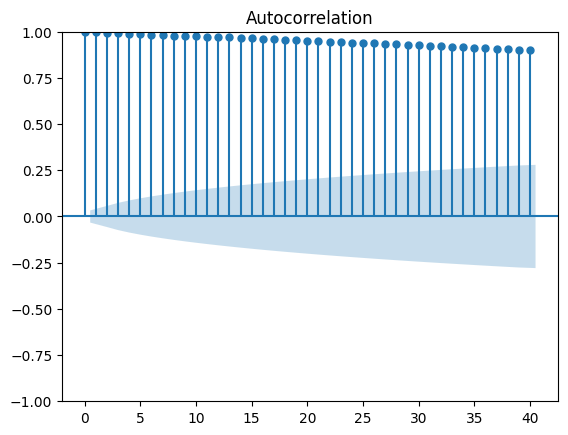
**General ARIMA Model Formula:**

****

**Optimization of Parameters**:

* Auto-ARIMA was used to find the best parameters (p, d, q).
* Best Order: (0, 1, 1).





**Visualization**

1. Prediction Plot:
   * ARIMA predictions closely followed the test data's trend, but with some deviations during high variability periods.
   * The plot indicates a reasonably accurate model for iterative forecasting.
2. Autocorrelation and Partial Autocorrelation:
   * ACF (Autocorrelation Function): Shows strong lag correlation, indicating the need for differencing.
   * PACF (Partial ACF): Indicates the significance of lagged terms in ARIMA modeling.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Model** | **MSE** | **R-Squared** | **MAE** | **RMSE** | **MAPE** |
| **LSTM** | 0.0019 | 0.7055 | 0.0353 | 0.0440 | 4.7025% |
| **GRU** | 0.0013 | 0.8030 | 0.0224 | 0.0360 | 3.0241% |
| **Bi-LSTM** | 0.0015 | 0.7691 | 0.0253 | 0.0389 | 3.4771% |
| **ARIMA** | 0.0025 | 0.8466 | 0.1391 | 0.0499 | 2273% |