Model Validation Report

[DOCUMENT SUBTITLE]

**Document Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Description of change | Author | date |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**Model Info**

|  |  |
| --- | --- |
| Model | Autoencoder |
| Model Version | Version 3 (??) |
| Date | 11/08/2022 |
| Primary Validator | Navid Kalantari |
| Model Registry link |  |

**Scope**

The following encompasses the scope of the MVR:

* An independent assessment of the appropriateness & performance of a model to meet its design objectives
* Observations of technical model risks and/or limitations, with associated mitigation steps taken
* Business risks associated with the model usage and the mitigations to those risks

**Model Performance/Appropriateness**

Provide text and/or evidence that the model is meeting its performance and appropriateness including its design and business objectives.

**Summary:**

The Autoencoder model is an unsupervised learning model. The goal of the model is to reduce the number of false positives that are resulting from the SHAPE model. As such, the performance of the autoencoder shall be validated both in terms of how well it's able to reconstruct the inputs (as an autoencoder), and how well it can remove positives from the output of the SHAPE model.

The validation result verifies that the model is able to reduce the number of false positives that result from the SHAPE model. In terms of reconstruction error, the performance of the model is also acceptable but could be improved.

**Verification:**

As mentioned in the summary, the validation of the autoencoder consists of validating its performance as an autoencoder model and how well it can reconstruct the input data, as well as the capability to reduce the number of false positives. The performance of the model in terms of the reconstruction loss will be discussed in detail later when discussing the training performance.

This section will focus more on the business side and the ability of the model to reduce the number of false positives. To verify that the model is able to reduce the number of false positives we have joined three databases together: the outputs of the shape model, the results of the autoencoder, and the rules-based labels for the data on Jan 20, 2022. This analysis could have been done better with more data points over a longer time period, but at the time of this analysis, due to the creation of time windows for the autoencoder, this would have been very time-consuming.

The rules-based model identified 9 positive cases for that particular day. Table 1 shows the confusion matrix from the SHAPE model. As shown in this table the SHAPE model was able to identify three (3) of the positive cases out of the nine (9) total positive cases. The number of false positive cases from the shape model was 9,900. The result of the model after the application of the autoencoder is shown in Table 2. As could be seen in this table the number of false positive cases is reduced to 7,292 cases. As such, the autoencoder was able to identify %26 of the false positives. Due to a large number of negative cases, the accuracy and recall of the model are not changed significantly but the precision of the model is increased from 0.000303 to 0.000411.

The retrained model performs better than the previous version of the model (shown in Table 3). The previous version of the model removed 2,444 false positive cases, while the retrained model identified 2,608. The performance of the retrained model is approximately %7 better than the previous version of the model.

The six cases that were miss classified by the Shape model are shown in Table 4.

Does the model meet the identified design objectives? 🞏

Does the model meet the identified business purposes? 🞏

**Data Assurance**

Provide text and/or evidence that support the following was followed as per the Model Card:

* appropriate data quality checks were performed
* feature selection process
* usage of data proxy, if applicable, was appropriate
* training/evaluation split methodology

**Summary**

Overall the data quality checking was performed in the training pipeline. Feature selection was not needed for the autoencoder as was not germane in this case. %75 of the data is kept for training and the rest (%25) was used for validation. The validation set was taken from the end of the dataset. The model training procedure used all the implied volatility data to train the autoencoder. Perhaps the use of only true positives could have potentially improved the performance of the autoencoder in this application. Other than this, no serious issue was identified regarding thea assurance.

**Verification:**

The dataset used covers a single-day (Jan 20, 2022) option trade data. This date is a Thursday. The data is converted into a Volatility Index (VI) for a 120-second time window. Then the VI is min-max scaled to have a min of zero and a max value of one. As will be seen later, the data has a very narrow range. This suggests that data normalization, instead of min-max scaling might have also been a viable potential option.

The dataset has a total of 1,286,217 observations. The test and validation split procedure assign the first 75% of the data to the training set and the remaining to the validation set. As such, the validation set contains the data that is related to the end of the day. This could potentially be an issue, although there is no evidence of this posing any real problems in the training and validation accuracy of the model.

One potential issue is the use of both positive and negative samples in the training of the autoencoder. The suggestion would have been to train the model using positive observations, with the hope that the autoencoder will learn how to reproduce positives only. This could have resulted in the inability of the autoencoder to reproduce negative samples.

The dataset has been controlled for duplicate and nan values and no instance of either one of them has been observed. The description of the data is shown in Table ….

The distribution of the training and validation set is shown in Figure … and Figure …. Figure … shows how the two distributions lay on each other and Figure … shows their joint distribution. Both data sets seem to be having a somewhat similar distribution and no serious issues could be observed.

The time window for some positive and negative cases are also shown in Figures …. To ….

Was the data approach from the Model Card Followed? 🞏

**Model Selection & Training**

Provide text and/or evidence that the Model Section Process was as correct including the following:

* hyperparameter tuning
* Training of the model
* feature selection process
* appropriateness of the model chosen

**Summary**

The model has not done any hyperparameter tuning or exploration of other model structures. There are some critisims to the model structure used in the autoencoder, which could potentially improve the model results. As mentioned previously feature selection is not applicable to this model. This section will focus more on the training of the autoencoder and its reconstruction ability.

**Verification** The structure of the model used is as given in …. The model consists of a 64-unit LSTM followed by a 20% dropout on the encoder side. The input to the LSTM is a 120-unit one-dimensional array and the encoder shrinks it to 64 units. The dropout randomly sets 20% of the outputs to zero. Similarly, on the decoder side, there is an LSTM layer that up-samples the 64-unit data to the original 120-unit data. This model structure is similar to the one used in the previous version of the autoencoder model.

The training is done using the Adam optimization method and is trained for 10 epochs. The loss function used in the training is Mean Squared Error (MSE). The final MSE on the training set varies for each run as the seed has not been set but it's approximately 0.009. The history of the model training and validation loss is given in Figure ….

Looking at the MSE would not provide a clear view of the model's accuracy in this case. The range of the input data is between zero and one and is very consolidated in the 0.9 to 1 range. As the data is less than one, and as a result, the errors are also expected to be less than one, squaring the errors will result in small numbers; that is the reason why the MSE is very low. Alternatively, it would be better to look at the Mean Absolute Error and preferably the Mean Absolute Percentage Error (MAPE). Table … shows the goodness of fit measures for the training and validation set.

Other than the aggregate error, a more detailed error analysis is also required. The error analysis should be performed in both space and time. Error is space is the amount of reproduction error per each of the 120 items. Figure … shows the level of error per each of the 120 data points in each time window. As could be seen in this figure the magnitude of error is relatively higher for the first few items and then gets relatively similar after that. That is very typical in sequence models where the effect of the first items is lost in time. There are some remedial measures for this issue such as using bidirectional LSTMs, etc.

From a time perspective, each of the points in the 120-unit time window is a five-second observation. So the training data spans some time during the day. To validate the consistency of the errors over time, we sliced both the training and validation data into ten (10) time buckets and looked at the changes in the error across each bucket. As shown in Figure … the amount of error seems to be consistent in both training and validation and within all the ten buckets.

The model doesn’t seem to show serious signs of overfitting or underfitting. Although the fact that the error of the validation set is consistently lower than the training set is an interesting point to consider. We have looked into the consistency between the training and validation set in the previous chapter in detail. The validation and training do not seem to have any significant difference in terms of distribution. The fact that the validation loss is consistently lower could be a result of using dropout in the model training process.

The model is compared with three different alternatives during the validation process.

* Using the overall mean of the data and using that single number as an autoencoder will only slightly deteriorate the performance of the model. The MAPE of the reconstruction error will change from 4% to 6%.
* Use the variance of each time window, and use a cutoff value for the variance. As the model …
* Changing the details of the model could result in better reconstruction errors. Reducing the dimension of the LSTM from 64 to 20 and removing the dropout shows an improvement in the reconstruction loss compared to the original model used. This is shown in Figure …

Was the Model Selection approach from the Model Card Followed? 🞏

**Model Monitoring Plan**

Provide text and/or evidence that the Model Monitoring Pion is satisfactory. Include how it ensure that the model continues to perform as expected in accordance with its design objectives and business purpose.

**Summary**

Describe the high-level Model Monitoring Pion

Click or tap here to enter text.

**Verification:**

Click or tap here to enter text.

Is the Proposed Model Monitoring Plan satisfactory? 🞏

**Assurance Goals**

Provide test and or evidence that the Assurance goals have been met including the following:

* evidence of the execution of the Assurance Plan
* built and tested as per plan
* Provide results of the model
* Quantitative Analysis {Perf over total data set and subsets of data)
* Model Monitoring Plan is sufficient

**Summary:**

Describing high-level what Assurance Goals were met and what Assurance Plan was followed

Click or tap here to enter text.

**Verification:**

Click or tap here to enter text.

Was the type and level of Assurance activities performed sufficient? 🞏

**Risks**

Provide any risks with the Release that need to be considered as part of this release. Be sure to consider Business and Technical risks.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk Type (Business/Technical/Other)** | **Risk Description** | **Severity** | **Mitigation** | **Potential Impact** |
| Business | (Autoencoder)  Different model structures with potentially fewer parameters could have been explored. The existing training pipeline doesn’t do any hyperparameter tuning and doesn’t explore other potential options to reduce the number of false positives. | Medium | More detailed analysis and use of hyperparameter tuning in the model training pipeline. | Potential loss of model performance. |
| Business | (Autoencoder)  The model is trained using all the data (both positive and negative). Training the model only on positive data could have resulted in the model learning the behavior of positive data and better identification of the negative cases. | Medium | Train the autoencoder on positive data only and then apply it to the entire dataset. | Potential loss of model performance. |
| Technical | (Autoencoder)  The model training pipeline was not easy to run and there were some issues in the pipeline. The addition of a read-me file could also facilitate the model run. | Low | More rigorous code review and addition of a read-me file to the repository. | Increase the effort of retraining. |
| Technical | (Autoencoder)  The seed of the random number and the training process was not set. As a result, it was difficult to reproduce the results shown in the model card. | Low | The random seeds should be set of all the required processes. | Increase the effort of retraining. |

**Sign Off**

Provide a formal sign off on the release of the model under test.

Click or tap here to enter text.