**PHASE 2: INNOVATION**

* **CONSIDER INCORPORATING MACHINE LEARNING ALGORITHMS TO PREDICT FUTURE SALES TRENDS OR CUSTOMER BEHAVIOUR.**

1. **Data Collection and Quality: Ensure you have access to high-quality data. This includes historical sales data, customer data, and any relevant external data sources.**
2. **Data Preprocessing: Clean and preprocess your data to ensure it's in a suitable format for machine learning algorithms. This may involve handling missing values, normalizing data, and encoding categorical variables.**
3. **Feature Selection: Identify the most relevant features or variables that are likely to impact sales trends or customer behaviors. Feature selection is a critical step in building accurate models.**
4. **Model Selection: Choose appropriate machine learning models for your task. For sales predictions, time series forecasting models like ARIMA or machine learning models like Random Forests and Gradient Boosting are commonly used. For customer behavior predictions, classification and clustering algorithms can be valuable.**
5. **Training and Testing: Split your data into training and testing sets to evaluate the performance of your models. Cross-validation techniques can help in model selection and hyperparameter tuning.**
6. **Hyperparameter Tuning: Optimize the hyperparameters of your chosen machine learning models to improve their predictive accuracy.**
7. **Evaluation Metrics: Define appropriate evaluation metrics for your specific task. For sales predictions, metrics like Mean Absolute Error (MAE) or Root Mean Square Error (RMSE) are common. For customer behavior predictions, metrics such as accuracy, precision, recall, or F1-score may be used.**
8. **Data Pipelines: Create data pipelines that allow for the automated collection, preprocessing, and modeling of data. This can help ensure that your predictions stay up to date with new data.**
9. **Interpretability: Depending on the context, you may need to consider how to make your machine learning models interpretable, especially if you need to explain the predictions to stakeholders.**
10. **Monitoring and Maintenance: Continuously monitor the performance of your models and update them as new data becomes available. Machine learning models can become less accurate over time if not maintained properly.**
11. **Ethical Considerations: Be aware of potential biases in your data and ensure that your models do not perpetuate or exacerbate any existing biases.**
12. **Regulatory Compliance: Depending on your industry, there may be regulations and privacy considerations to address, especially when dealing with customer data.**
13. **Incorporating machine learning into your sales and customer behavior prediction processes can provide valuable insights and help in making data-driven decisions. However, it's important to approach this with a well-thought-out strategy and a focus on data quality and ethical considerations.**

**PYTHON CODE:**

**import pandas as pd**

**import numpy as np**

**import matplotlib.pyplot as plt**

**import seaborn as sns**

**default\_path = '../input/'**

**!ls ../input**

**train\_df = pd.read\_csv(default\_path+'sales\_train.csv')**

**items\_df = pd.read\_csv(default\_path+'items.csv')**

**test\_df = pd.read\_csv(default\_path+'test.csv')**

**train\_df['date'] = pd.to\_datetime(train\_df['date'], format='%d.%m.%Y')**

**dataset = train\_df.pivot\_table(index=['item\_id', 'shop\_id'],values=['item\_cnt\_day'], columns='date\_block\_num', fill\_value=0)**

**dataset = dataset.reset\_index()**

**dataset = pd.merge(test\_df, dataset, on=['item\_id', 'shop\_id'], how='left')**

**dataset = dataset.fillna(0)**

**dataset = dataset.drop(['shop\_id', 'item\_id', 'ID'], axis=1)**

**X\_train = np.expand\_dims(dataset.values[:, :-1], axis=2)**

**y\_train = dataset.values[:, -1:]**

**X\_test = np.expand\_dims(dataset.values[:, 1:], axis=2)**

**y\_test = dataset.values[:, :1]**

**from keras.models import Sequential**

**from keras.layers import LSTM, Dense, Dropout**

**model = Sequential()**

**model.add(LSTM(units=64, input\_shape=(33, 1)))**

**model.add(Dropout(0.3))**

**model.add(Dense(1))**

**model.compile(loss='mse',**

**optimizer='adam',**

**metrics=['mean\_squared\_error'])**

**history = model.fit(X\_train, y\_train, batch\_size=4096, epochs=10)**

**plt.plot(history.history['loss'], label= 'loss(mse)')**

**plt.plot(np.sqrt(history.history['mean\_squared\_error']), label= 'rmse')**

**plt.legend(loc=1)**

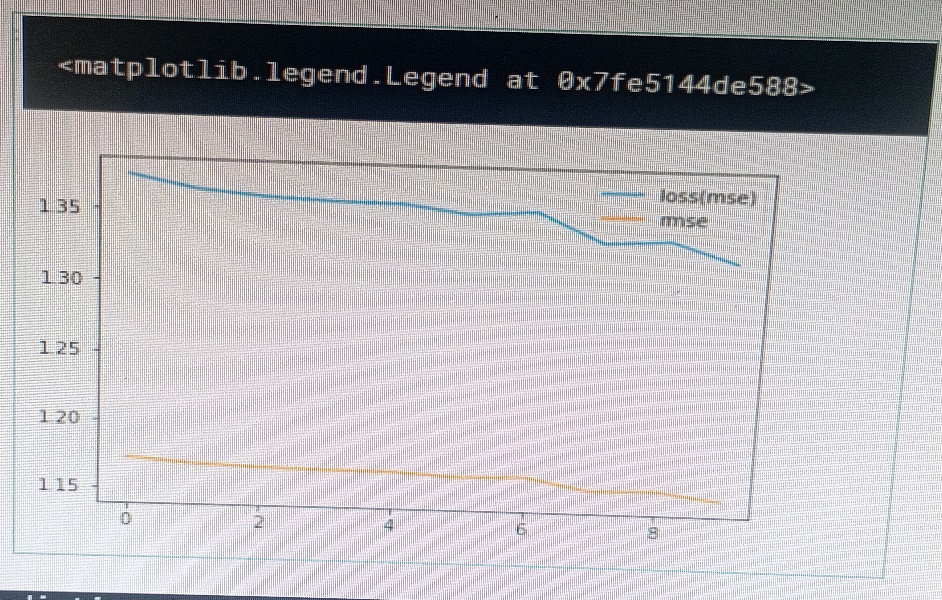
**LSTM\_prediction = model.predict(X\_test)**

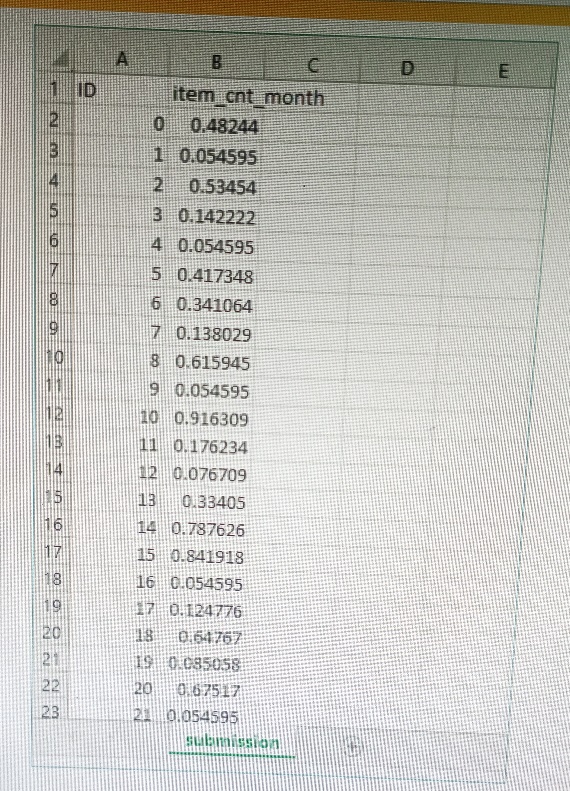
**LSTM\_prediction = LSTM\_prediction.clip(0, 20)**

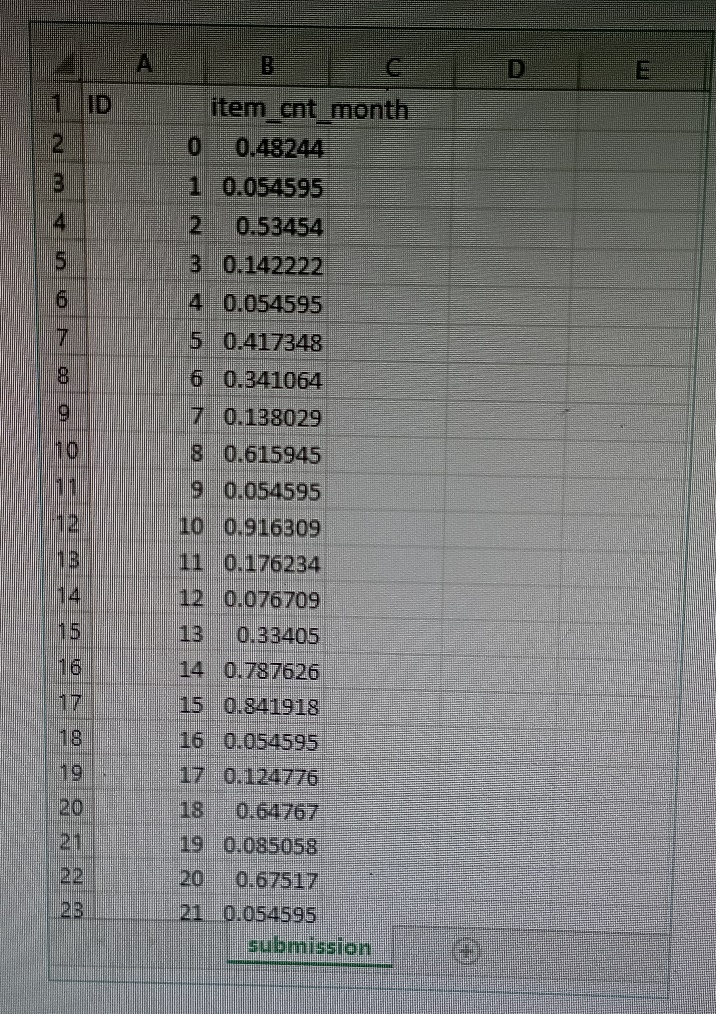
**submission = pd.DataFrame({'ID': test\_df['ID'], 'item\_cnt\_month': LSTM\_prediction.ravel()})**

**submission.to\_csv('submission.csv',index=False)**

**OUTPUT:**

**(1).**

****

****