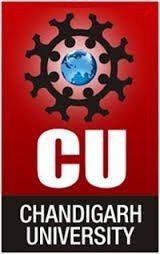
Tentative Payment Date Predictor

Submitted in partial fulfillment of the requirements for the award of degree of

### BACHELOR OF ENGINEERING IN

**COMPUTER SCIENCE & ENGINEERING**



**Submitted to: Submitted By:**

**Sugandhi Midha mam Shivam****(18bcs3054)**

**Mentor Signature**

**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING**

**Chandigarh University, Gharuan**

**Project Design**

**Module 1: Data collection Project Design**

In this machine learning project, we will develop Tentative Payment Date Predictor model with neural network to predict the near payment dates for professionalism in market and smooth business flow.

Learn how to develop a prediction model and how to build an interactive dashboard for tentative date analysis. We implemented Tentative Payment Date predictor using the LSTM model, OTOH, Plotly dash python framework for building dashboards.

Category: Machine Learning, Deep Learning Programming Language: Python

Tools & Libraries: Plotly Dash, LSTM

IDE: Jupyter Front End: Plotly Dash (for visualization) Back End: NA

Prerequisites: Python, Machine Learning, Deep Learning, Neural Network Intended Audience: Education, Developers, Data Engineers, Data Scientists

## Algorithms Used:

1. Support Vector Machines (SVM) and
2. Artificial Neural Networks (ANN) **Modules and libraries:** Numpy

Pandas Dash Plotly.graph Keras Sklearn

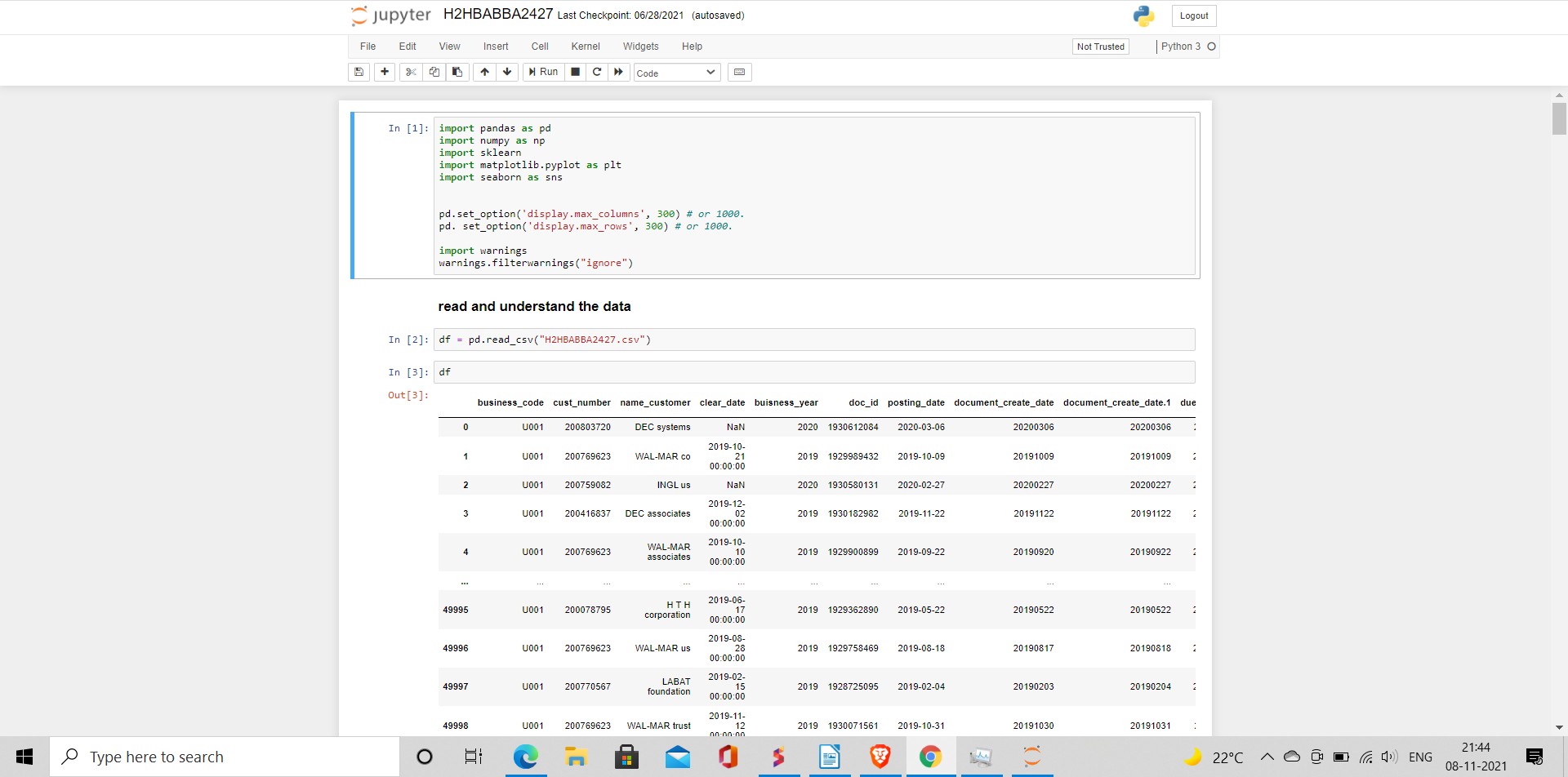
# Innovation in model/design/solution

### Approaches:

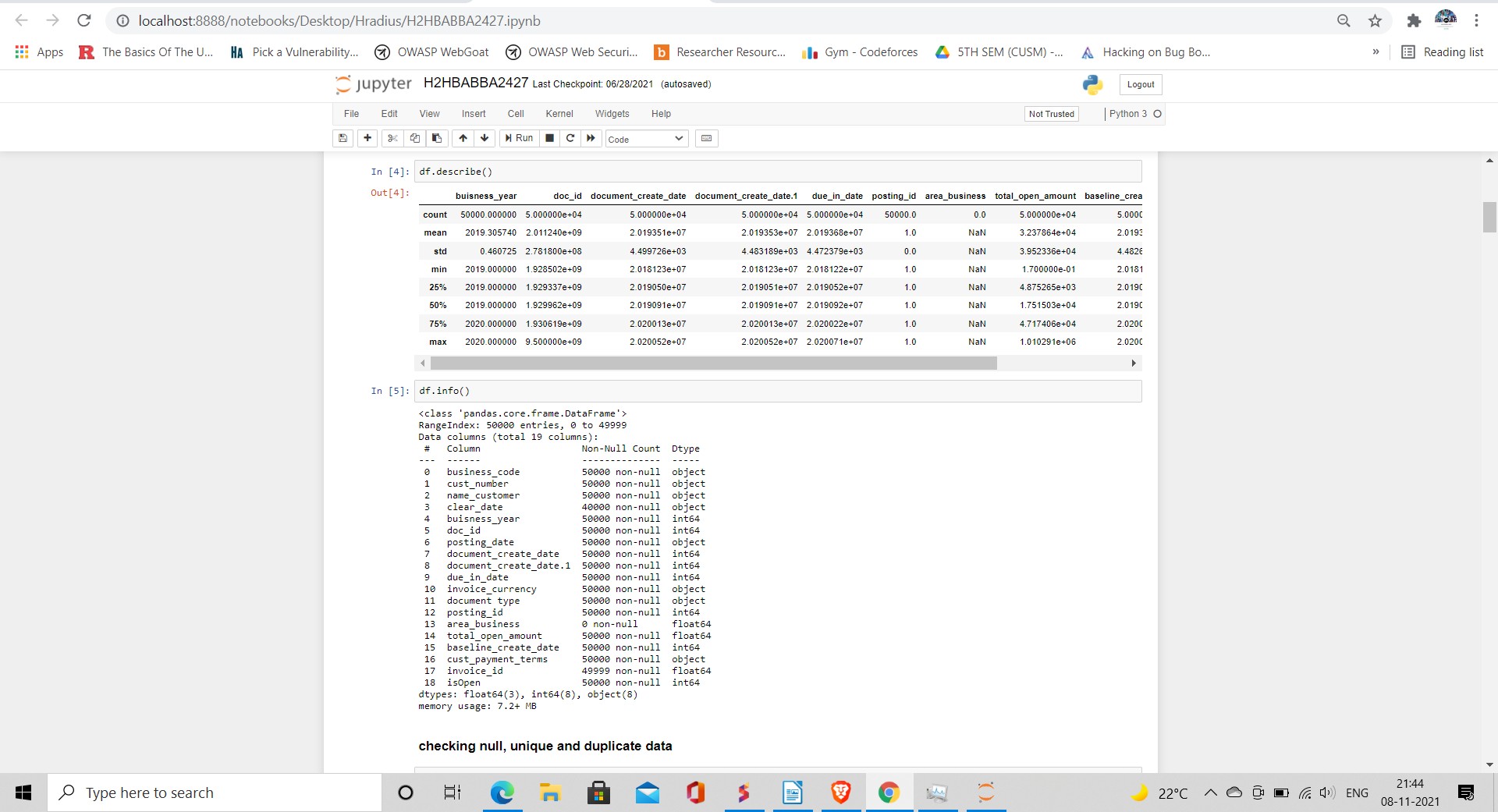
1. Read the dataset:
2. Analyze the closing prices from dataframe:
3. Sort the dataset on date time and filter “Date” and “Close” columns:
4. Normalize the new filtered dataset:
5. Build and train the LSTM model:
6. Take a sample of a dataset to make Payment Date predictions using the LSTM model:
7. Save the LSTM model:
8. Visualize the predicted payment dates with actual payment dates.

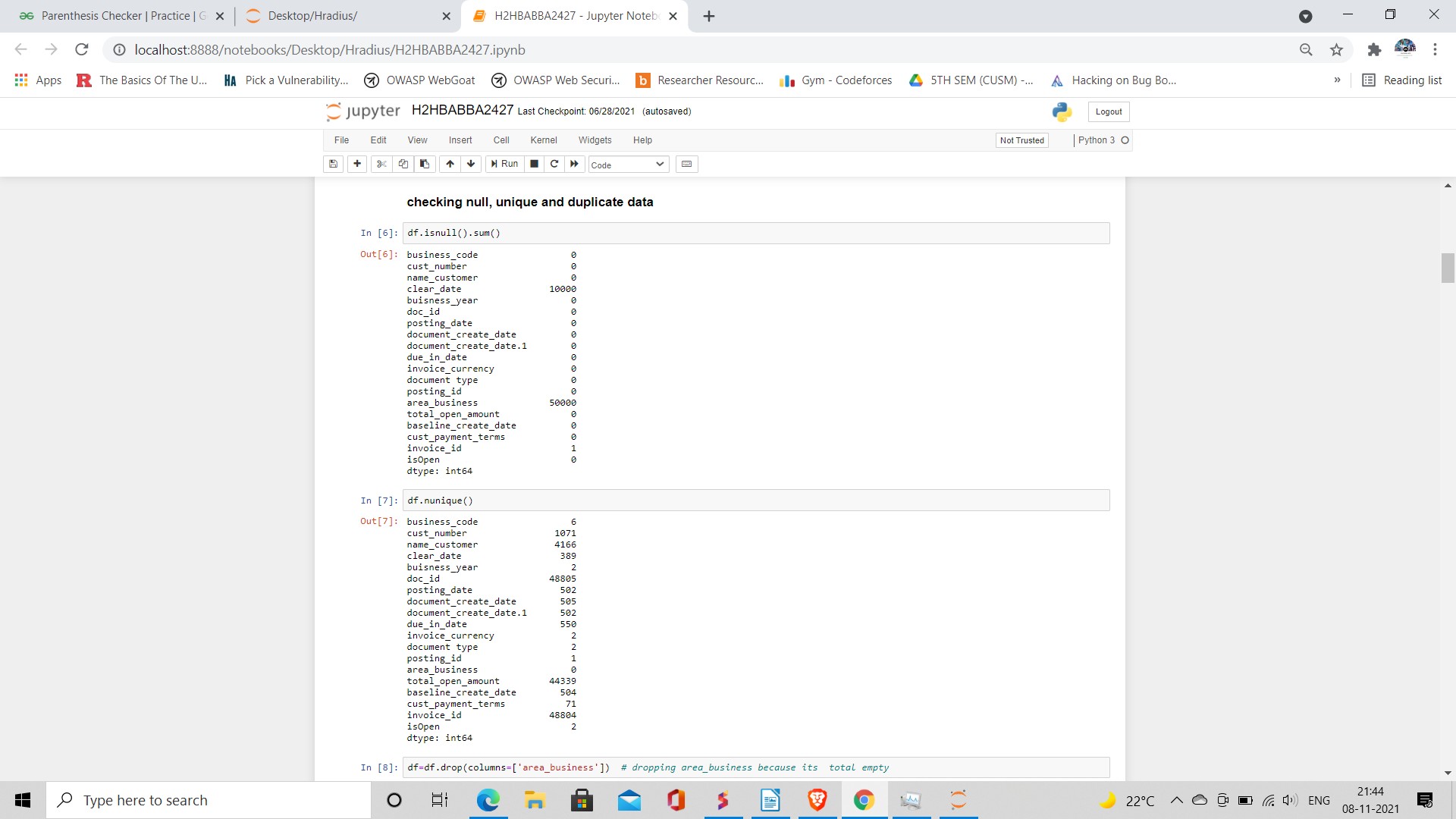
**Implementation (at least 40%)**

1. **Import libraries and read data**



1. **Checking Null, unique and Duplicate data**





1. **Checking Correlation and data interdependency**

