**Dependencies and ML Description**

**Dependencies Used:**

from serpapi import GoogleSearch

import pandas as pd

import numpy as np

from datetime import datetime

from openpyxl import load\_workbook

from sqlalchemy import create\_engine

import math

import matplotlib.pyplot as plt

import seaborn as sn

from scipy.stats import shapiro

from scipy.stats import lognorm

from scipy.stats.stats import pearsonr

from sklearn import tree

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import StandardScaler

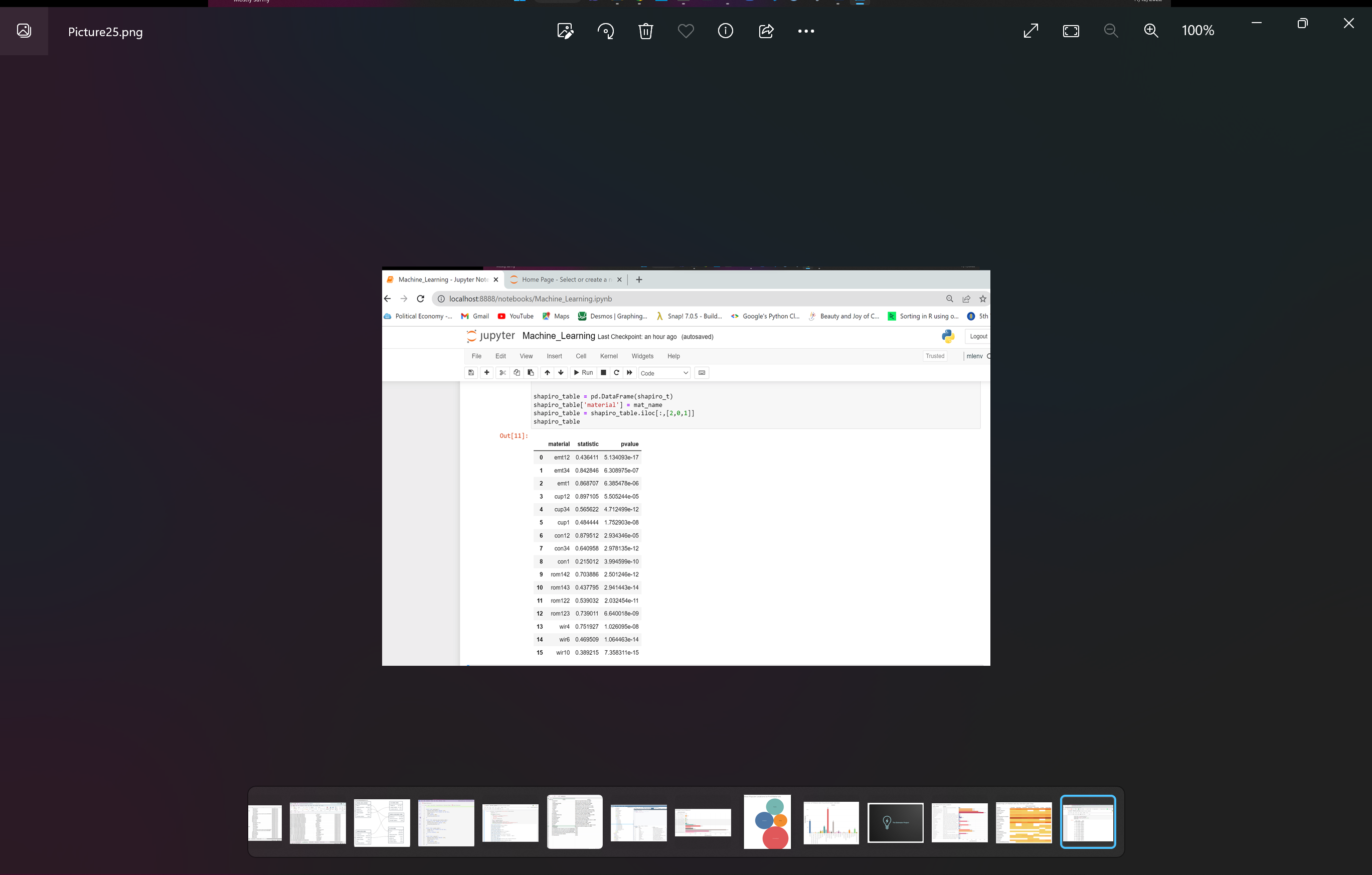
from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import confusion\_matrix, accuracy\_score, classification\_report

from sklearn.datasets import load\_iris

from sklearn.naive\_bayes import GaussianNB

**ML Description**

1. I ran the data through a Shapiro-Wilk test to determine if our material prices are normally distributed. As you can see from the table below, all p-values are less than .05 which means that our material prices are normally distributed. (see “Picture25.png”). 
2. I checked to see if there was a correlation between:
   1. Conduit and coupling (fitting)
   2. Conduit and connector (fitting)
   3. #4 wire and #6 wire
   4. #6 wire and #10 wire
   5. #10 wire and # 4 wire

Results: All the items I tested show correlation. All were less than a p-value of 0.05 (see “Picture26.png”). Graphical user interface, text

Description automatically generated

There is likely a correlation between the price of finished materials and the raw materials needed to make them (copper to wire and steel to conduit). So, I added the following features:

* Price of copper per pound (10/20-11/19)
* Price of steel per pound (10/20-11/19)

1. Next, I label encoded: material description, vendor, date, day of week, region, and, target price but kept the price column as is to create a new data frame. I ran a correlation matrix and saw some connections (see Picture27.png). Graphical user interface, text, application

   Description automatically generated
2. With the data encoded, I dropped “about average” to define a target of either “low price” or “high price”. With this done, I was able to separate my data into features (X) and target (y).
3. Finally, I split the data: test train split (Training data = 75% and Test data = 25%), scaled the data, created a **Decision Tree** model, fit the model and made predictions: Based on material name, price, vendor, date, day of week and region could I predict high or low prices (see “Picture28.png”)

Graphical user interface, text

Description automatically generated

1. Last, I created a confusion matrix and printed my results (“Picture29.png”). Graphical user interface, text, application

   Description automatically generated