```
import sys
import pandas as pd
import numpy as np
import random
import re
import joblib
import json
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score

def load_data(csvfile):
    """
    Read in CSV file data
    *This function assumes held-out data has the same format as training data

    """

    df = pd.read_csv(csvfile)

    ids = np.array(df["complaint_id"])
    product_groups = np.array(df["product_group"])
    text_list = np.array(df["text"])

    return ids, product_groups, text_list

def vectorize_documents(document_list, vectorizer):
    """
    Function to vectorize input text documents

    """
    X = vectorizer.transform(document_list)

    return X

if __name__ == "__main__":

    input_data_file = sys.argv[1]

    print("\nLoading Data...")
    ids, product_groups, text_list = load_data(input_data_file)

    print("\nLoading Group Labels...")
    with open("group_mapping.json", "r") as file:
        group_to_label = json.load(file) # Get mapping for model numbers to group labels

    print("\nLoading Vectorizer...")
    vectorizer = joblib.load("NWCaseStudyvectorizer.joblib")

    print("\nLoading Model...")
    clf = joblib.load("NWCaseStudyNBModel.joblib")

    print("\nVectorizing Input Data...")
    X = vectorize_documents(text_list, vectorizer)
    y = np.array([group_to_label[group] for group in product_groups])

    print("\nMaking Predictions...")
    y_pred = clf.predict(X)

    acc = accuracy_score(y_pred, y)
    print("Model Accuracy: %f" %acc)
```