Multilayer Perceptron Analysis of freight data for NRC the freight dataset from https://https://www150.statcan.gc.ca/n1/pub/50-503-x/50-503-x2018001-eng.htm) is used. This dataset contains information about more than 50,000 cases which were recorded from 2011 to 2016. The Canadian Freight Analysis Framework (CFAF) integrates data from several sources to create a comprehensive picture of freight flows across the country by geography, commodity and mode of transport. The framework database estimates tonnage, value, and tonnekilometers by origin and destination, by commodity type, and by mode. They recommmend that the database can be used in a variety of analyses such as assessing highway capacity and forecasting traffic, evaluating investments in infrastructure, examining trade flows, and analyzing policies including road pricing and multimodal freight programs. Information on the variables available in the database is provided. Using the information, the goal is to build a model to predict which case is profitable (for example, revenue is bigger that the specific value) and which case is not. **Data type Length Description Values** Year character 4 Reference year Mode character 2 The mode by which the shipment(s) moved. AR = Air RL = Rail TF = Truck (for-hire) SCTGGroup character 50 The type of commodity shipped. Groupings based on the Standard Classification of Transported Goods (SCTG) 2-digit level. OrigCMA character 3 In Canada, the sub-provincial area (census agglomeration (CA), census metropolitan area (CMA), rest of province (ROP)) or province/territory which is the origin of the shipment(s). Outside Canada, the United States and Mexico are grouped and all other countries are grouped. OrigProv character 2 In Canada, the province or territory which is the origin of the shipment(s). Outside Canada, the United States and Mexico are grouped and all other countries are grouped. OrigCtry character 2 The country which is the origin of the shipment(s). DestCMA character 3 In Canada, the sub-provincial area (census agglomeration (CA), census metropolitan area (CMA), rest of province (ROP)) or province/territory which is the destination of the shipment(s). Outside Canada, the United States and Mexico are grouped and all other countries are grouped. DestProv character 2 In Canada, the province or territory which is the destination of the shipment(s). Outside Canada, the United States and Mexico are grouped and all other countries are grouped. DestCtry character 2 The country which is the destination of the shipment(s). Shipments numeric 38 The aggregate number of shipments transported. For air and truck, a shipment represents the movement of a single commodity from an origin to a destination. For rail this represents the number of cars. Blank cells contain data which have been suppressed to meet the confidentiality requirements of the Statistics Act. Weight numeric 38 The aggregate weight of the shipments, in kilograms (kg). Blank cells contain data which have been suppressed to meet the confidentiality requirements of the Statistics Act. Revenue numeric 38 The aggregate revenue that the carrier earned from transporting the shipments, in dollars. Blank cells contain data which have been suppressed to meet the confidentiality requirements of the Statistics Act. Distance numeric 38 The aggregate distance that the shipments were transported, in kilometres (km). Blank cells contain data which have been suppressed to meet the confidentiality requirements of the Statistics Act. TonneKm numeric 38 The weight of each shipment multiplied by the distance transported and then aggregated. Blank cells contain data which have been suppressed to meet the confidentiality requirements of the Statistics Act. Value numeric 38 The aggregate value of the shipments, in dollars. Blank cells contain data which have been suppressed to meet the confidentiality requirements of the Statistics Act. **Read in Data** In [1]: import matplotlib.pyplot as plt import numpy as np import pandas as pd import seaborn as sns %matplotlib inline freight = pd.read\_csv('../../freight.csv') freight.head() Out[1]: Year Mode Commodity OrigCMA OrigProv OrigCtry DestCMA DestProv DestCtry Shipments Weight Revenue Miscellaneous **0** 2011 Calgary Alberta CANADA Edmonton Alberta CANADA NaN NaN NaN products Miscellaneous Nova CANADA **1** 2011 Calgary Alberta CANADA Halifax 311.0 22724.0 31504.10 products Scotia Miscellaneous **2** 2011 Calgary Alberta CANADA Hamilton Ontario CANADA 620.0 3560913.0 4846957.66 products Miscellaneous **3** 2011 Calgary Alberta CANADA Montreal Quebec CANADA 2086.0 244641.0 334898.57 products Miscellaneous New New CANADA **4** 2011 Air Calgary Alberta CANADA NaN NaN NaN Brunswick Brunswick products Clean continuous variables Fill missing values In [2]: freight.isnull().sum() Out[2]: Year 0 Mode 0 0 Commodity OrigCMA 0 0 OrigProv 0 OrigCtry 0 DestCMA DestProv 0 DestCtry 0 Shipments 1318 Weight 1318 Revenue 1318 Distance 1318 TonneKm 1318 Value 1318 dtype: int64 freight['Shipments'].fillna(freight['Shipments'].mean(), inplace=True) freight.head(10) Out[3]: Commodity OrigCMA OrigProv OrigCtry Year Mode DestCMA DestProv DestCtry Shipments Weight Miscellaneous **0** 2011 Calgary Alberta CANADA Edmonton Alberta CANADA 8479.342545 NaN products Miscellaneous **1** 2011 Calgary Alberta CANADA Halifax Nova Scotia CANADA 311.000000 22724.0 products products Miscellaneous **3** 2011 Alberta CANADA 244641.0 Calgary Montreal Quebec CANADA 2086.000000 products Miscellaneous New **4** 2011 Calgary Alberta CANADA CANADA 8479.342545 NaN products Brunswick Brunswick Newfoundland Newfoundland Miscellaneous CANADA 8479.342545 **5** 2011 Calgary Alberta CANADA and Labrador and Labrador products Miscellaneous Northwest Northwest CANADA 8479.342545 **6** 2011 Calgary Alberta CANADA Territories **Territories** products Other Miscellaneous Other Other **7** 2011 1799.000000 11335702.C Calgary Alberta CANADA international international products international Miscellaneous CANADA 8479.342545 **8** 2011 Calgary Alberta CANADA NaN Edward Island Edward Island products Miscellaneous **9** 2011 Quebec City CANADA 8479.342545 Calgary Alberta CANADA Quebec NaN products freight['Weight'].fillna(freight['Weight'].mean(), inplace=True) freight.head(10) Out[4]: Year Mode Commodity OrigCMA OrigProv OrigCtry **DestCMA** DestProv DestCtry **Shipments** Weig Miscellaneous **0** 2011 Air Alberta CANADA Alberta CANADA 8479.342545 1.219892e+ Calgary Edmonton products Miscellaneous **1** 2011 311.000000 2.272400e+ Air Alberta CANADA Halifax Nova Scotia CANADA Calgary products Miscellaneous **2** 2011 Air Alberta CANADA Hamilton CANADA 620.000000 3.560913e+ Calgary Ontario products Miscellaneous **3** 2011 Air Alberta CANADA Montreal Quebec CANADA 2086.000000 2.446410e+ Calgary products Miscellaneous New **4** 2011 Air Alberta CANADA CANADA 8479.342545 1.219892e+ Calgary products Brunswick Brunswick Miscellaneous Newfoundland Newfoundland **5** 2011 CANADA 8479.342545 1.219892e+ Alberta CANADA Calgary products and Labrador and Labrador Miscellaneous Northwest Northwest **6** 2011 Air Alberta CANADA CANADA 8479.342545 1.219892e+ Calgary products Territories Territories Other Miscellaneous Other Other Alberta CANADA **7** 2011 1799.000000 1.133570e+ Calgary international products international international Prince Miscellaneous Prince **8** 2011 Air Calgary Alberta CANADA CANADA 8479.342545 1.219892e+ Edward Island Edward Island products Miscellaneous 9 2011 Quebec City CANADA 8479.342545 1.219892e+ Calgary Alberta CANADA Quebec products In [5]: | freight['Distance'].fillna(freight['Distance'].mean(), inplace=True) freight.head(10) Out[5]: Commodity OrigCMA OrigProv OrigCtry **DestCMA** Year Mode DestProv DestCtry Shipments Weig Miscellaneous **0** 2011 CANADA Alberta CANADA 8479.342545 1.219892e+ Calgary Alberta Edmonton products Miscellaneous **1** 2011 311.000000 2.272400e+ Calgary Alberta CANADA Halifax Nova Scotia CANADA products Miscellaneous **2** 2011 620.000000 3.560913e+ Calgary Alberta CANADA Hamilton CANADA Ontario products Miscellaneous **3** 2011 Calgary Alberta CANADA Quebec CANADA 2086.000000 2.446410e+ Montreal products Miscellaneous New **4** 2011 CANADA 8479.342545 1.219892e+ Calgary Alberta CANADA Brunswick Brunswick products Newfoundland Miscellaneous Newfoundland Alberta CANADA **5** 2011 CANADA 8479.342545 1.219892e+ Calgary and Labrador and Labrador products Miscellaneous Northwest Northwest CANADA 8479.342545 1.219892e+ **6** 2011 Calgary Alberta CANADA Territories **Territories** Miscellaneous Other Other Other 1799.000000 1.133570e+ **7** 2011 Calgary Alberta CANADA products international international international Miscellaneous Prince **8** 2011 CANADA 8479.342545 1.219892e+ Calgary Alberta CANADA Edward Island Edward Island products Miscellaneous **9** 2011 Air Alberta CANADA CANADA 8479.342545 1.219892e+ Calgary Quebec City Quebec products In [6]: freight['TonneKm'].fillna(freight['TonneKm'].mean(), inplace=True) freight.head(10) Out[6]: **DestCMA** Year Mode Commodity OrigCMA OrigProv OrigCtry DestProv DestCtry Shipments Weig Miscellaneous **0** 2011 8479.342545 1.219892e+ Calgary Alberta CANADA Edmonton Alberta CANADA Miscellaneous **1** 2011 Halifax 311.000000 2.272400e+ Calgary Alberta CANADA Nova Scotia CANADA products Miscellaneous **2** 2011 CANADA CANADA 620.000000 3.560913e+ Calgary Alberta Hamilton Ontario Miscellaneous **3** 2011 CANADA 2086.000000 2.446410e+ Calgary Alberta CANADA Montreal Quebec products Miscellaneous New New **4** 2011 CANADA 8479.342545 1.219892e+ Calgary Alberta CANADA Brunswick Brunswick Newfoundland Miscellaneous Newfoundland **5** 2011 CANADA CANADA 8479.342545 1.219892e+ Calgary Alberta products and Labrador and Labrador Miscellaneous Northwest Northwest **6** 2011 CANADA CANADA 8479.342545 1.219892e+ Calgary Alberta Territories **Territories** Other Other Miscellaneous **7** 2011 1799.000000 1.133570e+ Calgary Alberta CANADA products international international international Miscellaneous Prince Prince **8** 2011 CANADA 8479.342545 1.219892e+ Calgary Alberta CANADA Edward Island Edward Island Miscellaneous **9** 2011 Alberta CANADA CANADA 8479.342545 1.219892e+ Calgary Quebec City Quebec products In [7]: freight['Value'].fillna(freight['Value'].mean(), inplace=True) freight.head(10) Out[7]: Year Mode Commodity OrigCMA OrigProv OrigCtry **DestCMA** DestProv DestCtry Shipments Weig Miscellaneous 8479.342545 1.219892e+ **0** 2011 Alberta CANADA Alberta CANADA Calgary Edmonton products Miscellaneous **1** 2011 Air Alberta CANADA Halifax Nova Scotia CANADA 311.000000 2.272400e+ Calgary products Miscellaneous **2** 2011 Air Alberta CANADA Hamilton Ontario CANADA 620.000000 3.560913e+ Calgary products Miscellaneous **3** 2011 Air Alberta CANADA Montreal Quebec CANADA 2086.000000 2.446410e+ Calgary products New Miscellaneous New **4** 2011 Alberta CANADA CANADA 8479.342545 1.219892e+ Calgary products Brunswick Brunswick Miscellaneous Newfoundland Newfoundland CANADA 8479.342545 1.219892e+ **5** 2011 Air CANADA Calgary Alberta and Labrador products and Labrador Miscellaneous Northwest Northwest CANADA 8479.342545 1.219892e+ **6** 2011 Air Alberta CANADA Calgary products Territories Territories Miscellaneous Other Other 1799.000000 1.133570e+ **7** 2011 Air Alberta CANADA Calgary products international international Prince Miscellaneous Prince **8** 2011 Alberta CANADA CANADA 8479.342545 1.219892e+ Calgary Edward Island Edward Island products Miscellaneous CANADA 8479.342545 1.219892e+ **9** 2011 Air Alberta CANADA Calgary Quebec City Quebec products Combine **Drop unnnecessary variables** In [8]: freight.drop(['OrigCMA', 'OrigCtry', 'DestCMA', 'DestCtry'], axis=1, inplace=True) In [9]: | freight.head() Out[9]: Commodity OrigProv DestProv Year Mode Shipments Weight Revenue Distance TonneKm Miscellaneous **0** 2011 Alberta 8479.342545 1.219892e+08 NaN 6.497016e+06 9.254620e+07 2.848 products Miscellaneous Nova **1** 2011 Alberta 311.000000 2.272400e+04 31504.10 1.164384e+06 8.507866e+04 3.913 products Scotia Miscellaneous **2** 2011 Alberta 620.000000 3.560913e+06 4846957.66 1.672140e+06 9.603782e+06 6.132 products Miscellaneous Clean categorical variables Fill in missing & create indicator for `Revenue' In [10]: freight.isnull().sum() Out[10]: Year 0 Mode 0 Commodity 0 OrigProv 0 DestProv 0 Shipments 0 0 Weight 1318 Revenue Distance TonneKm 0 Value 0 dtype: int64 **Convert Mode to numeric** mode\_num = {'Air': 1, 'Rail': 2, 'Truck for-hire':3} freight['Mode'] = freight['Mode'].map(mode\_num) freight.head() Out[12]: Year Mode Commodity OrigProv DestProv Shipments Weight Distance **TonneKm** Revenue Miscellaneous 8479.342545 1.219892e+08 6.665525e+06 6.497016e+06 9.254620e+07 2.8 **0** 2011 Alberta products Miscellaneous Nova 311.000000 2.272400e+04 3.150410e+04 1.164384e+06 8.507866e+04 3.9 **1** 2011 Alberta products Scotia Miscellaneous **2** 2011 Alberta 620.000000 3.560913e+06 4.846958e+06 1.672140e+06 9.603782e+06 6.1 products Miscellaneous **3** 2011 Alberta Quebec 2086.000000 2.446410e+05 3.348986e+05 6.270484e+06 7.351513e+05 4.2 products Miscellaneous **4** 2011 8479.342545 1.219892e+08 6.665525e+06 6.497016e+06 9.254620e+07 2.8 Brunswick products origprov\_num = {'Alberta': 1, 'British Columbia': 2, 'Manitoba':3, 'New Brunswick':4, 'Newfo undland and Labrador':5, 'Northwest Territories':6, 'Nova Scotia':7, 'Nunavut':8, 'Ontario': 9, 'Other international':10, 'Prince Edward Island':11, 'Quebec':12, 'Saskatchewan':13, 'Unit ed States and Mexico':14, 'Yukon':15} freight['<mark>OrigProv</mark>'] = freight['<mark>OrigProv</mark>'].map(origprov\_num) Out[13]: Commodity OrigProv DestProv Shipments Year Mode Weight Revenue Distance TonneKm Miscellaneous **0** 2011 Alberta 8479.342545 1.219892e+08 6.665525e+06 6.497016e+06 9.254620e+07 2.8 products Miscellaneous Nova 311.000000 2.272400e+04 3.150410e+04 1.164384e+06 8.507866e+04 3.9 **1** 2011 Scotia Miscellaneous **2** 2011 Ontario 620.000000 3.560913e+06 4.846958e+06 1.672140e+06 9.603782e+06 6.1 Miscellaneous **3** 2011 Quebec 2086.000000 2.446410e+05 3.348986e+05 6.270484e+06 7.351513e+05 4.2 Miscellaneous **4** 2011 8479.342545 1.219892e+08 6.665525e+06 6.497016e+06 9.254620e+07 2.8 destprov\_num = {'Alberta': 1, 'British Columbia': 2, 'Manitoba':3, 'New Brunswick':4, 'Newfo undland and Labrador':5, 'Northwest Territories':6, 'Nova Scotia':7, 'Nunavut':8, 'Ontario': 9, 'Other international':10, 'Prince Edward Island':11, 'Quebec':12, 'Saskatchewan':13, 'Unit ed States and Mexico':14, 'Yukon':15} freight['DestProv'] = freight['DestProv'].map(destprov\_num) Out[14]: Year Mode Commodity OrigProv DestProv Shipments Weight Revenue **Distance** TonneKm Miscellaneous **0** 2011 1 8479.342545 1.219892e+08 6.665525e+06 6.497016e+06 9.254620e+07 2.8 Miscellaneous **1** 2011 311.000000 2.272400e+04 3.150410e+04 1.164384e+06 8.507866e+04 3.93 Miscellaneous **2** 2011 620.000000 3.560913e+06 4.846958e+06 1.672140e+06 9.603782e+06 6.13 Miscellaneous **3** 2011 12 2086.00000 2.446410e+05 3.348986e+05 6.270484e+06 7.351513e+05 4.2: products Miscellaneous **4** 2011 4 8479.342545 1.219892e+08 6.665525e+06 6.497016e+06 9.254620e+07 2.84 commodity\_num = {'Agricultural products': 1, 'Automobiles and other Transportation Equipmen t': 2, 'Base metals and Articles of Base metals':3, 'Coal':4, 'Food':5, 'Forest products':6, 'Fuel Oils and crude petroleum':7, 'Minerals':8, 'Miscellaneous products':9, 'Other Manufact ured goods':10,'Plastic and Chemical products':11, 'Waste and Scrap':12} freight['Commodity'] = freight['Commodity'].map(commodity\_num) Out[15]: Year Mode Commodity OrigProv DestProv Shipments Weight Revenue Distance TonneKm **0** 2011 1 8479.342545 1.219892e+08 6.665525e+06 6.497016e+06 9.254620e+07 2.848 **1** 2011 9 1 311.000000 2.272400e+04 3.150410e+04 1.164384e+06 8.507866e+04 3.913 **2** 2011 620.000000 3.560913e+06 4.846958e+06 1.672140e+06 9.603782e+06 6.132 **3** 2011 9 1 12 2086.00000 2.446410e+05 3.348986e+05 6.270484e+06 7.351513e+05 4.212 **4** 2011 4 8479.342545 1.219892e+08 6.665525e+06 6.497016e+06 9.254620e+07 2.848 **Drop unnecessary variables** freight.drop(['Revenue'], axis=1, inplace=True) Write out cleaned data In [17]: freight.to\_csv('../../freight\_cleaned.csv', index=False) In [18]: import pandas as pd from sklearn.model\_selection import train\_test\_split freight = pd.read\_csv('../../freight\_cleaned.csv') freight.head() Out[18]: Year Mode Commodity OrigProv DestProv Shipments **TonneKm** Value Rever Weight Distance **0** 2011 1 8479.342545 1.219892e+08 6.497016e+06 9.254620e+07 2.848707e+08 **1** 2011 311.000000 2.272400e+04 1.164384e+06 8.507866e+04 3.913263e+06 1 9 1 620.000000 3.560913e+06 1.672140e+06 9.603782e+06 6.132191e+08 **2** 2011 12 2086.000000 2.446410e+05 6.270484e+06 7.351513e+05 4.212923e+07 **3** 2011 9 1 **4** 2011 4 8479.342545 1.219892e+08 6.497016e+06 9.254620e+07 2.848707e+08 In [19]: | ffeatures = freight.drop('Revenue\_ind', axis=1) flabels = freight['Revenue\_ind'] X\_train, X\_test, y\_train, y\_test = train\_test\_split(ffeatures, flabels, test\_size=0.4, rando m\_state=42) In [20]: for dataset in [y\_train, y\_val, y\_test]: print(round(len(dataset) / len(flabels), 2)) 0.6 0.2 0.2 In [21]: | X\_train.to\_csv('../../train\_ffeatures.csv', index=False) X\_val.to\_csv('.../.../val\_ffeatures.csv', index=False) X\_test.to\_csv('../../test\_ffeatures.csv', index=False) y\_train.to\_csv('../../train\_flabels.csv', index=False) y\_val.to\_csv('.../.../val\_flabels.csv', index=False) y\_test.to\_csv('../../test\_flabels.csv', index=False) In [22]: import joblib import pandas as pd from sklearn.model\_selection import GridSearchCV from sklearn.neural\_network import MLPClassifier import warnings warnings.filterwarnings('ignore', category=FutureWarning) warnings.filterwarnings('ignore', category=DeprecationWarning) tr\_features = pd.read\_csv('.../.../train\_ffeatures.csv') tr\_labels = pd.read\_csv('../../train\_flabels.csv') In [23]: def print\_results(results): print('BEST PARAMS: {}\n'.format(results.best\_params\_)) means = results.cv\_results\_['mean\_test\_score'] stds = results.cv\_results\_['std\_test\_score'] for mean, std, params in zip(means, stds, results.cv\_results\_['params']):  $print('\{\} (+/-\{\}) for \{\}'.format(round(mean, 3), round(std * 2, 3), params))$ In [24]: mlp = MLPClassifier() parameters = { 'hidden\_layer\_sizes': [(10,), (50,), (100,)], 'activation': ['relu', 'tanh', 'logistic'], 'learning\_rate': ['constant', 'invscaling', 'adaptive'] cv = GridSearchCV(mlp, parameters, cv=5) cv.fit(tr\_features, tr\_labels.values.ravel()) print\_results(cv) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi zation hasn't converged yet. % self.max\_iter, ConvergenceWarning) C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5

71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5
71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5 71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

BEST PARAMS: {'activation': 'relu', 'hidden\_layer\_sizes': (100,), 'learning\_rate': 'invscalin

0.892 (+/-0.113) for {'activation': 'relu', 'hidden\_layer\_sizes': (10,), 'learning\_rate': 'co

0.916 (+/-0.08) for {'activation': 'relu', 'hidden\_layer\_sizes': (10,), 'learning\_rate': 'inv

0.911 (+/-0.08) for {'activation': 'relu', 'hidden\_layer\_sizes': (10,), 'learning\_rate': 'ada

0.929 (+/-0.023) for {'activation': 'relu', 'hidden\_layer\_sizes': (50,), 'learning\_rate': 'co

0.936 (+/-0.018) for {'activation': 'relu', 'hidden\_layer\_sizes': (50,), 'learning\_rate': 'in

0.936 (+/-0.029) for {'activation': 'relu', 'hidden\_layer\_sizes': (50,), 'learning\_rate': 'ad

0.911 (+/-0.045) for {'activation': 'relu', 'hidden\_layer\_sizes': (100,), 'learning\_rate': 'c

0.942 (+/-0.02) for {'activation': 'relu', 'hidden\_layer\_sizes': (100,), 'learning\_rate': 'in

0.933 (+/-0.032) for {'activation': 'relu', 'hidden\_layer\_sizes': (100,), 'learning\_rate': 'a

0.755 (+/-0.012) for {'activation': 'tanh', 'hidden\_layer\_sizes': (10,), 'learning\_rate': 'co

0.763 (+/-0.021) for {'activation': 'tanh', 'hidden\_layer\_sizes': (10,), 'learning\_rate': 'in

0.757 (+/-0.028) for {'activation': 'tanh', 'hidden\_layer\_sizes': (10,), 'learning\_rate': 'ad

0.771 (+/-0.005) for {'activation': 'tanh', 'hidden\_layer\_sizes': (50,), 'learning\_rate': 'co

C:\Users\purit\anaconda3\lib\site-packages\sklearn\neural\_network\\_multilayer\_perceptron.py:5
71: ConvergenceWarning: Stochastic Optimizer: Maximum iterations (200) reached and the optimi

zation hasn't converged yet.

g'}

scaling'}

ptive'}

nstant'}

aptive'}

vscaling'}

daptive'}

nstant'}

aptive'}

nstant'}

In [25]: freight.isnull().sum()

Out[25]: Year

Mode

Commodity

OrigProv

DestProv

Distance

Revenue\_ind dtype: int64

In [26]: cv.best\_estimator\_

Out[27]: ['../../MLP\_model.pkl']

TonneKm

Value

Weight

Shipments

zation hasn't converged yet.

0

0

0

0

0

0

0

0

o o

warm\_start=False)

In [27]: joblib.dump(cv.best\_estimator\_, '../../MLP\_model.pkl')

Out[26]: MLPClassifier(activation='relu', alpha=0.0001, batch\_size='auto', beta\_1=0.9,

beta\_2=0.999, early\_stopping=False, epsilon=1e-08, hidden\_layer\_sizes=(100,), learning\_rate='invscaling', learning\_rate\_init=0.001, max\_fun=15000, max\_iter=200,

tol=0.0001, validation\_fraction=0.1, verbose=False,

momentum=0.9, n\_iter\_no\_change=10, nesterovs\_momentum=True,
power\_t=0.5, random\_state=None, shuffle=True, solver='adam',

I have identified the best best estimator as above, which gives me 94.2%

Check if there's any null value.

vscaling'}

% self.max\_iter, ConvergenceWarning)

% self.max\_iter, ConvergenceWarning)