import pandas as pd

from sklearn.preprocessing import MinMaxScaler

from sklearn.feature\_selection import RFE

from sklearn.linear\_model import LinearRegression

import statsmodels.api as sm

def binary\_map(x):

return x.map({'yes': 1, "no": 0})

def prepare\_dataset(filename):

housing = pd.read\_csv(filename)

varlist = ['mainroad', 'guestroom', 'basement', 'hotwaterheating', 'airconditioning', 'prefarea']

housing[varlist] = housing[varlist].apply(binary\_map)

# Let's drop the first column from status df using 'drop\_first = True'

status = pd.get\_dummies(housing['furnishingstatus'], drop\_first = True)

# Add the results to the original housing dataframe

housing = pd.concat([housing, status], axis = 1)

housing.drop(['furnishingstatus'], axis = 1, inplace = True)

scaler = MinMaxScaler()

num\_vars = ['area', 'bedrooms', 'bathrooms', 'stories', 'parking']

housing[num\_vars] = scaler.fit\_transform(housing[num\_vars])

return housing

def learn(filename):

df\_train = prepare\_dataset(filename)

y\_train = df\_train.pop('price')

names = df\_train.pop('houseID')

X\_train = df\_train

lm = LinearRegression()

lm.fit(X\_train, y\_train)

rfe = RFE(lm, n\_features\_to\_select=10) # running RFE

rfe = rfe.fit(X\_train, y\_train)

col = X\_train.columns[rfe.support\_]

X\_train\_rfe = X\_train[col]

X\_train\_rfe = sm.add\_constant(X\_train\_rfe)

lm = sm.OLS(y\_train,X\_train\_rfe).fit()

X\_train\_rfe = X\_train\_rfe.drop(['const'], axis=1)

return lm, X\_train\_rfe.columns

def predict(model, filename, columns):

dataset = prepare\_dataset(filename)

X\_test\_new = dataset[columns]

X\_test\_new = sm.add\_constant(X\_test\_new)

preds = model.predict(X\_test\_new)

with\_predictions = dataset.merge(preds.to\_frame(), left\_index=True, right\_index=True)

with\_predictions = with\_predictions.rename(columns={0:'predicted'})

return with\_predictions

model, columns = learn('housing\_train.csv')

predictions = predict(model, 'housing\_test.csv', columns)

predictions = predictions[['houseID', 'predicted']]

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