```
import os
os.getcwd()
os.chdir(r"C:\Users\Avirup Gupta Roy\Desktop")
import pandas as pd
import numpy as np
df=pd.read_csv(r"accident.csv")
df['Monthly income of riders']=df['Monthly income of riders'].str.replace('$','')
df['Monthly income of riders']=df['Monthly income of riders'].str.replace(',','')
df['Price per week']=df['Price per week'].str.replace('$','')
df['Average parking rates per month']=df['Average parking rates per
month'].str.replace('$','')
df['Number of weekly riders']=df['Number of weekly riders'].str.replace(',','')
df['Population of city']=df['Population of city'].str.replace(',','')
df['Monthly income of riders']=df['Monthly income of riders'].astype(int)
df['Price per week']=df['Price per week'].astype(int)
df['Number of weekly riders']=df['Number of weekly riders'].astype(int)
df['Average parking rates per month']=df['Average parking rates per
month'].astype(int)
df['Population of city']=df['Population of city'].astype(int)
dataset=df.values
Y=dataset[:,1]
X=dataset[:,2:5]
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.6,
                                                    random_state=1)
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge
from sklearn.linear_model import Lasso
from sklearn.linear_model import ElasticNet
from sklearn.ensemble import BaggingRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.ensemble import ExtraTreesRegressor
from sklearn.tree import DecisionTreeRegressor
from sklearn.neighbors import KNeighborsRegressor
from sklearn.svm import SVR
from sklearn.metrics import explained_variance_score
from sklearn.metrics import mean_absolute_error
num_instances = len(X)

models = []
models.append(('LiR', LinearRegression()))
models.append(('Ridge', Ridge()))
models.append(('Lasso', Lasso()))
models.append(('ElasticNet', ElasticNet()))
models.append(('Bag_Re', BaggingRegressor()))
models.append(('RandomForest', RandomForestRegressor()))
models.append(('ExtraTreesRegressor', ExtraTreesRegressor()))
models.append(('KNN', KNeighborsRegressor()))
models.append(('CART', DecisionTreeRegressor()))
models.append(('SVM', SVR()))

results = []
names = []
scoring = []
for name, model in models:
    # Fit the model
```

```
    model.fit(X_train, y_train)
    predictions = model.predict(X_test)
     # Evaluate the model
    score = explained_variance_score(y_test, predictions)

    names.append(name)

    msg = "%s: %f " % (name, score)
    print(msg)
#subset selection
print("STEPWISE REGRESSION USING CROSS VALIDATION")
def step_cross_val(df):
    from sklearn.linear_model import LinearRegression
    from sklearn.cross_validation import cross_val_score
    from sklearn import preprocessing
    data_y=df['Number of weekly riders']
    features=df.columns.values
    consider=[]
    numerics=['City ', 'Price per week',
       'Population of city', 'Monthly income of riders',
       'Average parking rates per month']
    i=1
    for col in features:

        print("Including "+str(i)+": "+str(col))
        i+=1
        consider.append(col)
        if col not in numerics:

            le = preprocessing.LabelEncoder()
            df[col] = df[col].fillna("Missing")
            df[col] = le.fit_transform(df[col])

        else:
            df[col] = df[col].fillna(0)

        data = df[consider]
        X = data.values
        Y = data_y.values
        from sklearn.model_selection import train_test_split

        X_train, X_test, y_train, y_test = train_test_split(X, Y,
test_size=0.6,random_state=1)

        log_regressor = LinearRegression()
        log_regressor.fit(X_train, y_train)
        predictions = log_regressor.predict(X_test)
        score2 = cross_val_score(log_regressor,X_test,y_test.astype(float),cv=3)
        print ("lin_reg: mean="+str(np.mean(score2))+" std="+str(np.std(score2)))
step_cross_val(df)
```

```
LiR: 0.865303
Ridge: 0.865063
Lasso: 0.865295
ElasticNet: 0.864105
```

```
Bag_Re: 0.697878
RandomForest: 0.656451
ExtraTreesRegressor: 0.731687
KNN: 0.552123
CART: 0.722741
SVM: 0.000000
STEPWISE REGRESSION USING CROSS VALIDATION
Including 1: City
lin_reg: mean=0.8105332791535037 std=0.15018986324501793
Including 2: Number of weekly riders
lin_reg: mean=0.9495139426880659 std=0.0174763174806128
Including 3: Price per week
lin_reg: mean=0.947166113919017 std=0.020374778288352633
Including 4: Population of city
lin_reg: mean=0.9016503283097008 std=0.03807740470946127
Including 5: Monthly income of riders
lin_reg: mean=0.8887689979589486 std=0.04528626089539973
Including 6: Average parking rates per month
lin_reg: mean=0.8061851789404214 std=0.17061088809588135
```