Q1:
hour

Predicts: casual
Model: LinearRegression()
R2: 0.21451895598297788
MSE: 1498.4125076312766

Predicts: registered
Model: LinearRegression()
R2: 0.21647676130794066
MSE: 15797.208807015824

Predicts: cnt
Model: LinearRegression()
R2: 0.24248243957182863
MSE: 21303.391094605184

Predicts: casual
Model: KNeighborsRegressor()
R2: 0.522034739155139
MSE: 845.3736107246401

Predicts: registered
Model: KNeighborsRegressor()
R2: 0.36369059358267786
MSE: 9506.546750100255

Predicts: cnt
Model: KNeighborsRegressor()
R2: 0.4011517483657089
MSE: 13218.859712103713

day

Predicts: casual
Model: LinearRegression()
R2: 0.19960511434426753
MSE: 205017.51266836323

Predicts: registered
Model: LinearRegression()
R2: 0.39308643127205917
MSE: 573125.0011725498

Predicts: cnt
Model: LinearRegression()
R2: 0.14798117768654345
MSE: 1015257.9871236192

Predicts: casual
Model: KNeighborsRegressor()
R2: 0.22054436793692114
MSE: 225709.48848569565

Predicts: registered
Model: KNeighborsRegressor()

R2: 2.40694446915352
MSE: 2881763.670183208

Predicts: cnt
Model: KNeighborsRegressor()
R2: 2.3945796310020455
MSE: 4130593.09925226

The best model is KNeighborsRegressor for most of the prediction targets (casual, registered, cnt) in both the "hour" and "day" datasets. This can be seen from the higher R-squared (R2) values and lower Mean Squared Error (MSE) values for KNN compared to Linear Regression. I think this is because some features are non-linearity and it doesn't rely on the scale of the input features.

Q2:
Clusters predict:
Counter({3: 75, 2: 68, 1: 67})
Truth:
Counter({1: 70, 2: 70, 3: 70})
The results seems good as most of them are in the same group as it in truth data.