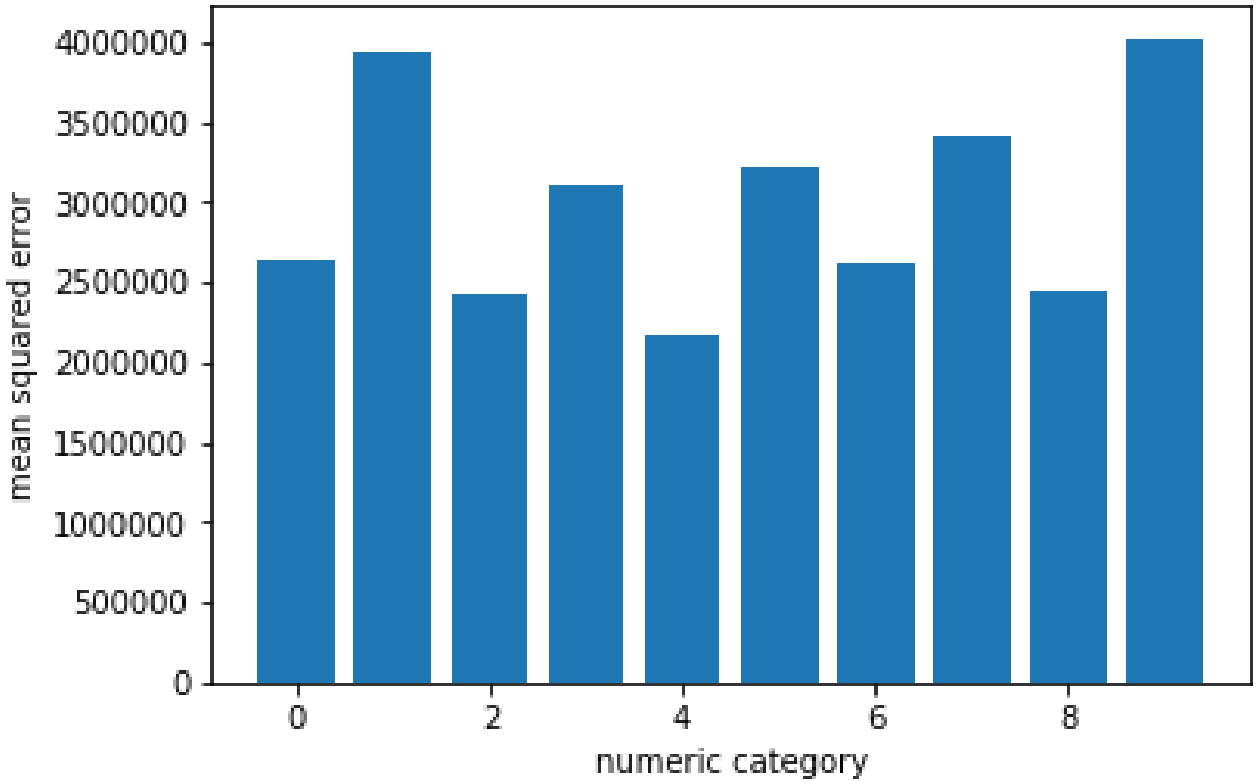
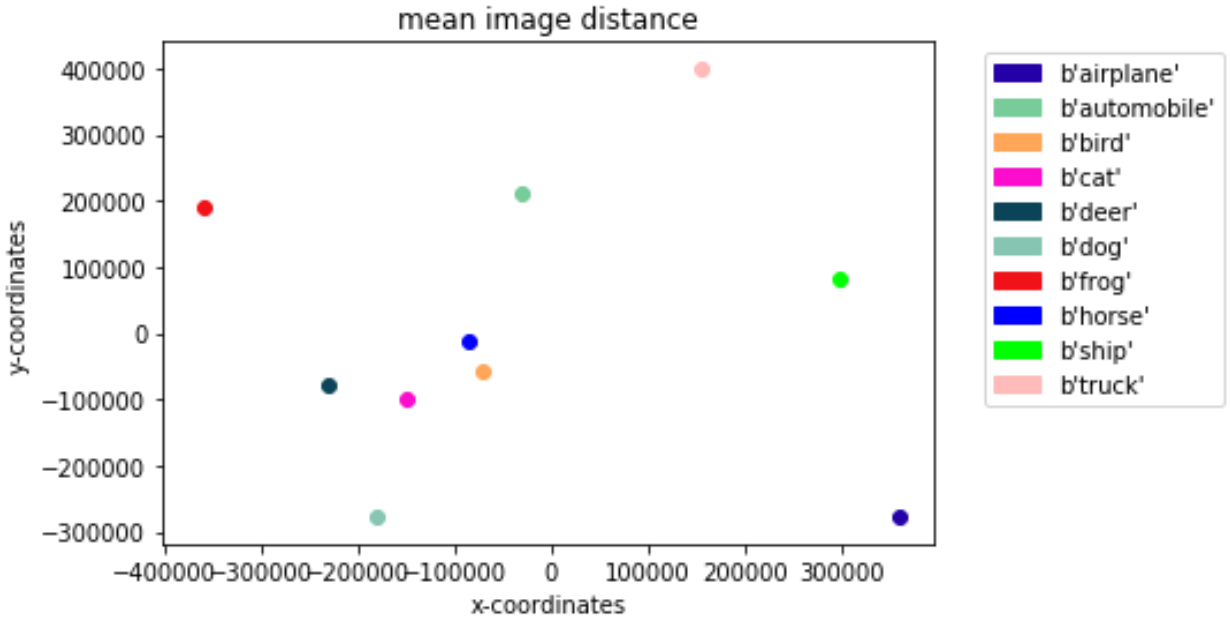


Each category errors

```
2629500.1004819553
3946291.224011452
2432114.02164956
3106143.4820486614
2181917.3378302944
3230258.326091132
2623435.5478147804
3424450.537289103
2438663.506102403
4024359.831349784
```




```
   x-coord      y-coord
[[ 0.51215621 -0.42662628]
 [-0.04443739  0.32894098]
 [-0.10092729 -0.0883715 ]
 [-0.21263145 -0.1514903 ]
 [-0.32848289 -0.11950559]
 [-0.25576397 -0.42702635]
 [-0.5117351   0.2940566 ]
 [-0.1200817  -0.02058604]
 [ 0.42573751  0.12823048]
 [ 0.22240012  0.61567833]]
```

```
#(a)
#label 0
pca_data = PCA(n_components=20)
pca_train_0 = pca_data.fit_transform(train_0)
train_0_mean = pca_data.mean_
pca_train0_recons = np.dot(pca_train_0, pca_data.components_) + pca_data.mean_
result_train_0 = mse(pca_train0_recons, train_0)
print(result_train_0)
result_q1 = []
result_q1.append(result_train_0)


#label 1
pca_data = PCA(n_components=20)
pca_train_1 = pca_data.fit_transform(train_1)
train_1_mean = pca_data.mean_
pca_train1_recons = np.dot(pca_train_1, pca_data.components_) + pca_data.mean_
result_train_1 = mse(pca_train1_recons, train_1)
print(result_train_1)
result_q1.append(result_train_1)
```

```
def _E_matrix(distance_matrix):
    return np.transpose(distance_matrix) * distance_matrix/2
```

```
#pcoa
E_distance = _E_matrix(distance_matrix)
A = np.zeros((10,10))
for i in range(10):
    A[i][i] = 1
A = A - 1/10
```

```
W = A*distance_matrix*np.transpose(distance_matrix)
eigvals, eigvecs = np.linalg.eigh(W)
```

```
idxs_descending = eigvals.argsort()[::-1]
new_eigvals = eigvals[idxs_descending]
new_eigvecs = eigvecs[:, idxs_descending]
new_eigvecs = new_eigvecs[:,0:2]
new_eigvals = new_eigvals[0:2]
```

```
u, s, vh = np.linalg.svd(W, full_matrices=False)
idxs_descending = s.argsort()[::-1]
s = s[idxs_descending]
for i in range(2,10):
    s[i] = 0
s = np.diag(s)
s = s[0:2,0:2]
result = np.matmul(new_eigvecs, s)
print(result)
```

```
distance_matrix = np.zeros((10,10))
for i in range(10):
    for j in range(10):
        distance_matrix[i][j] = distanceM(train_matrix[i], train_matrix[j])
```