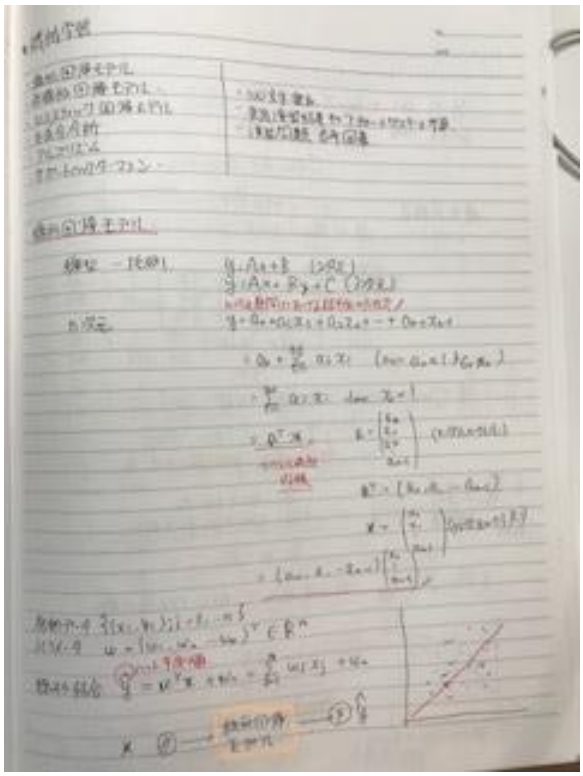
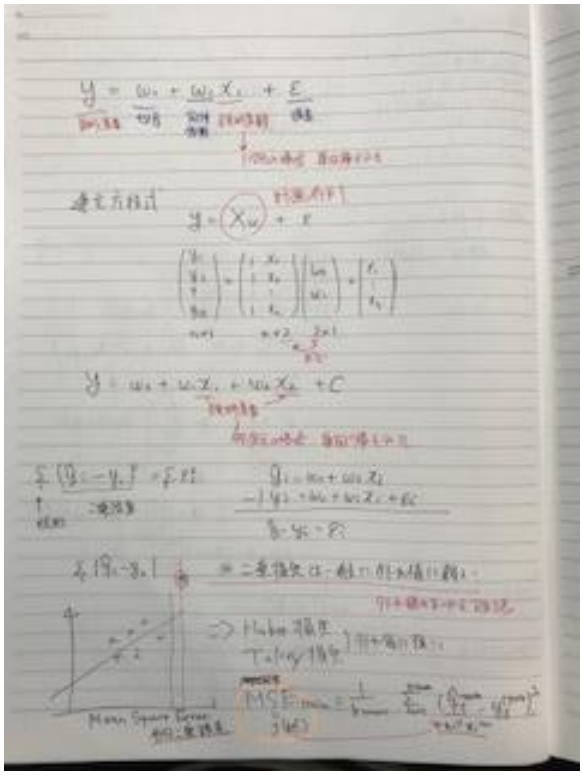
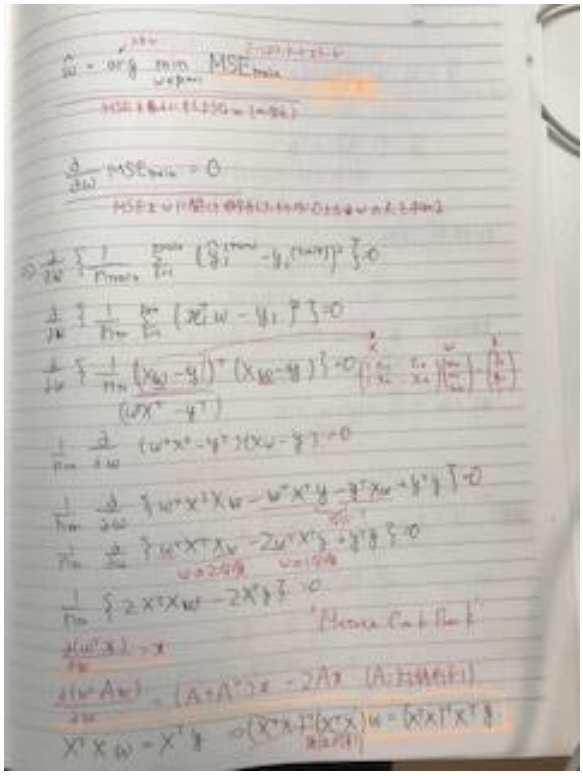
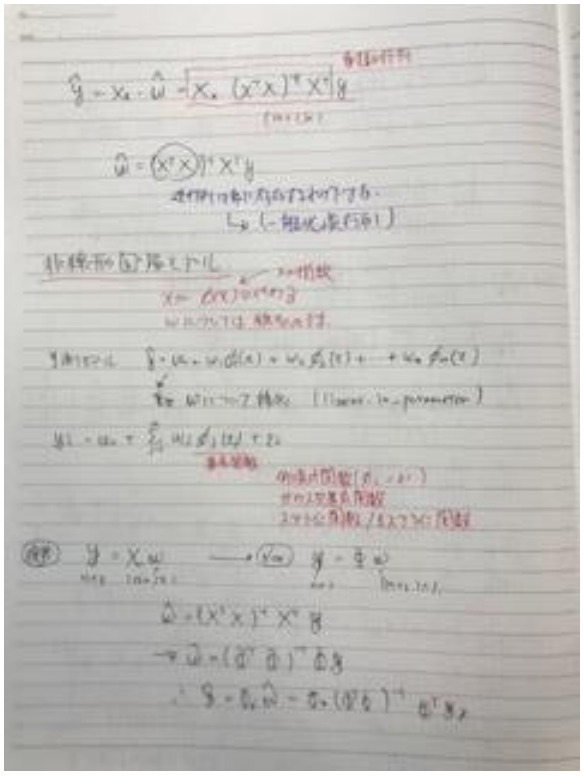
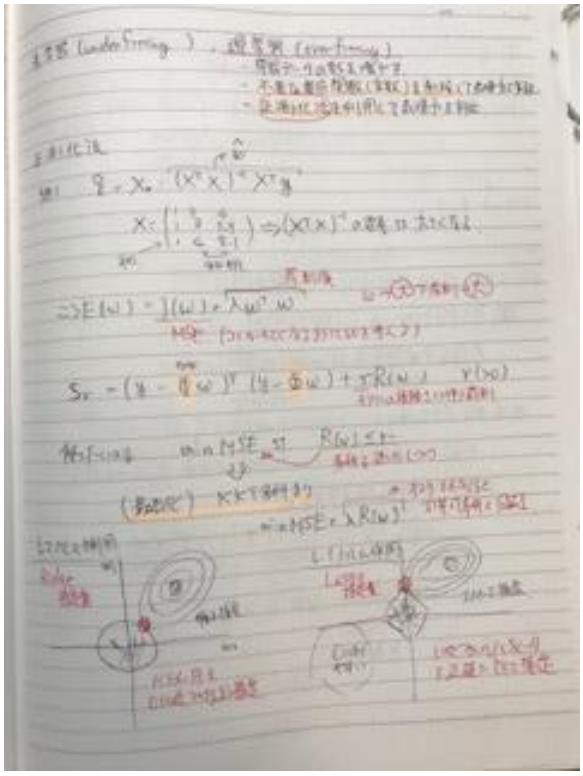
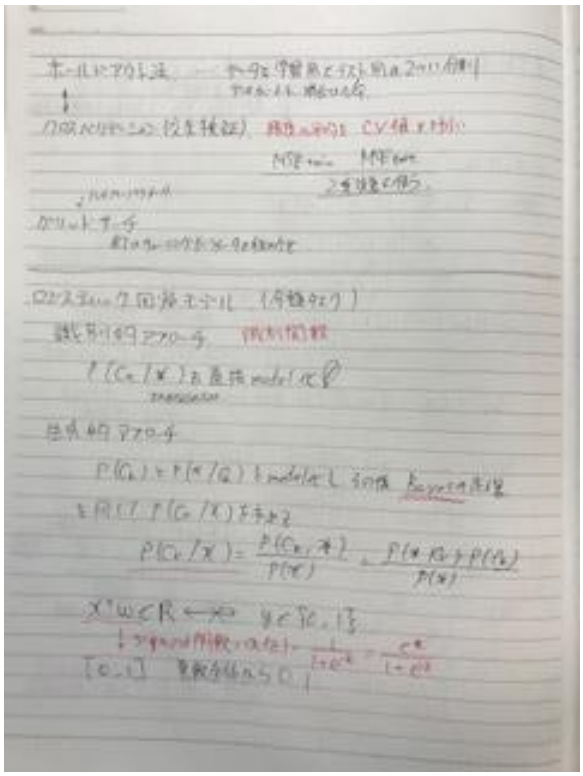


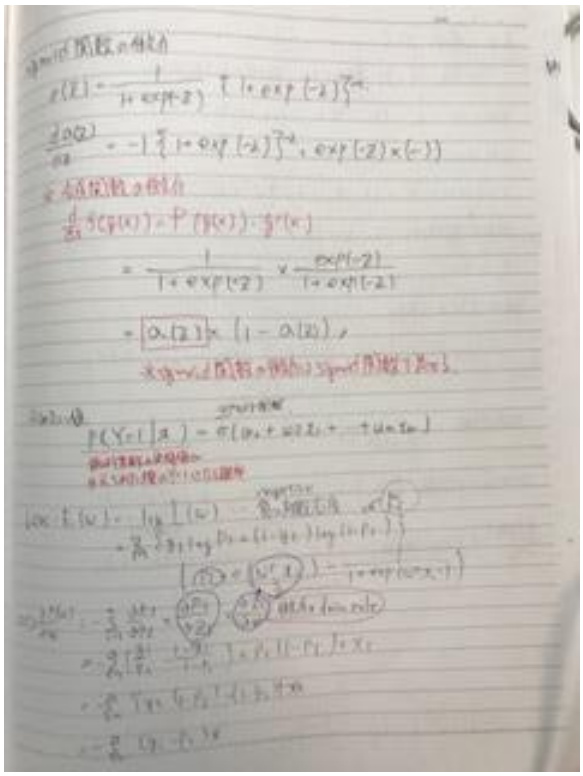


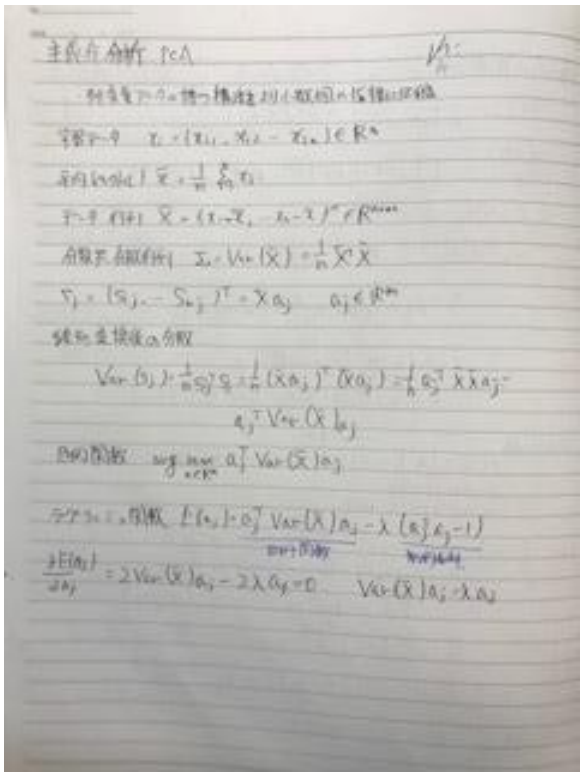


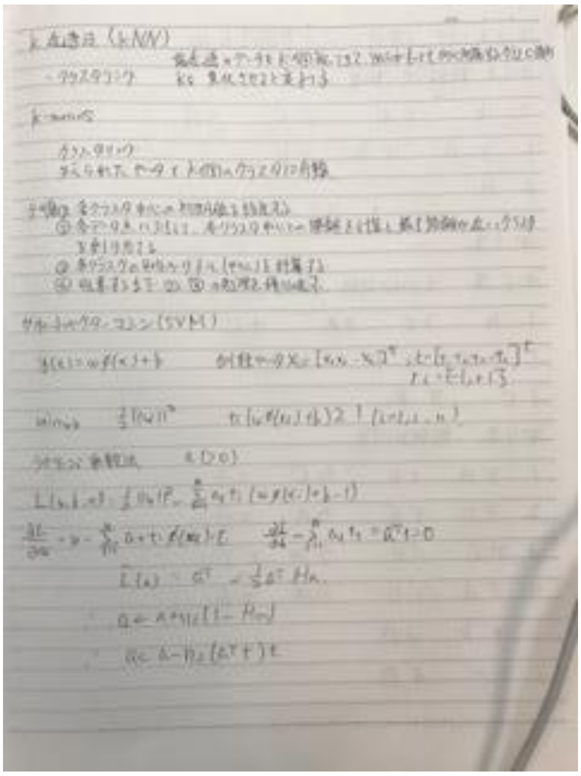


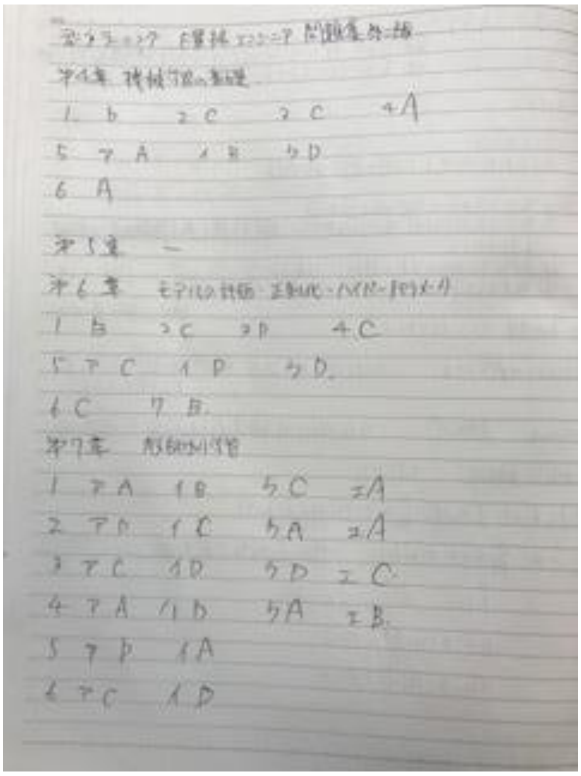


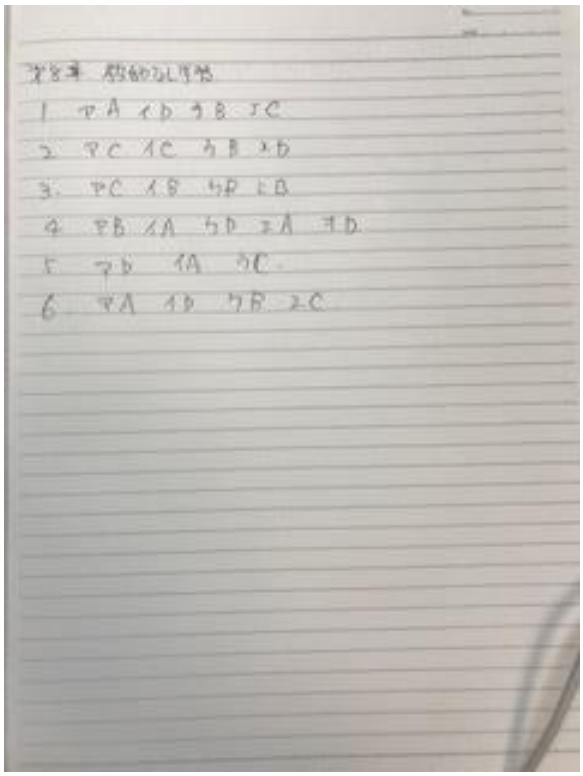


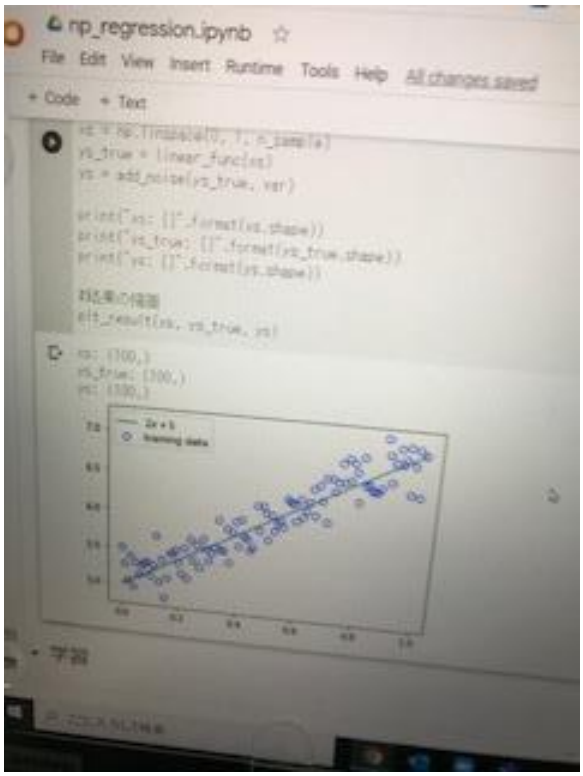


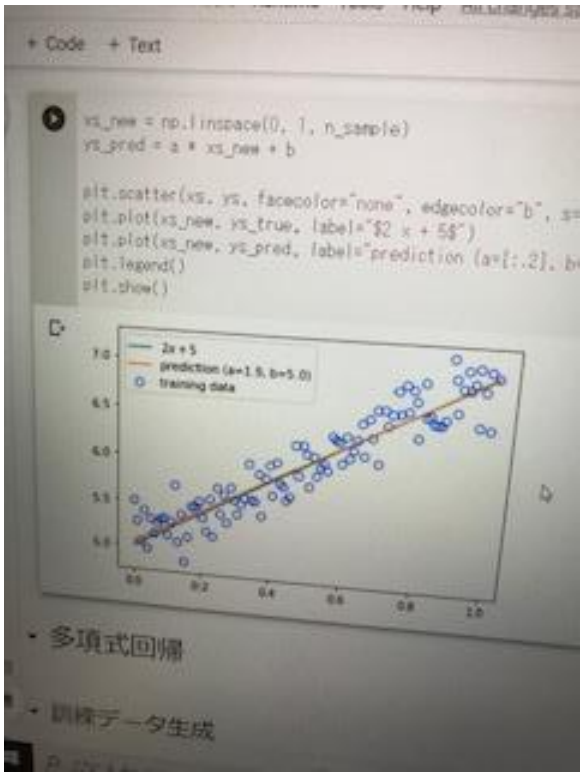


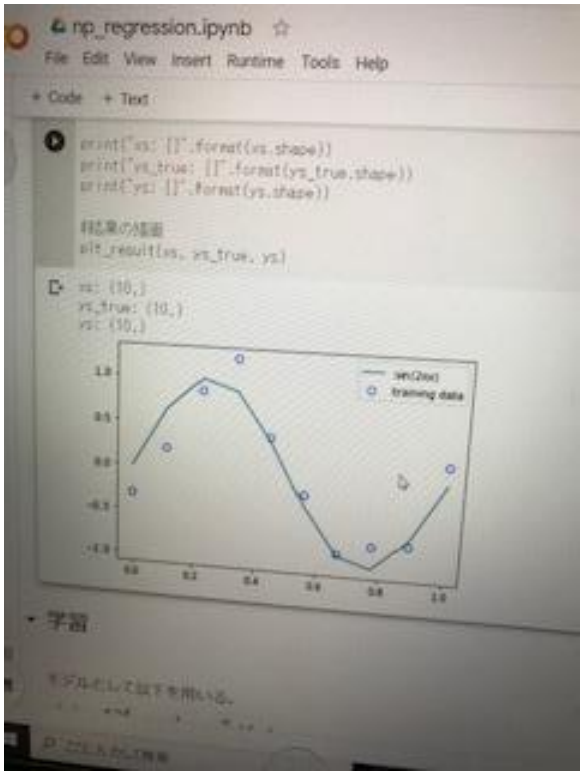

k近傍法 (kNN)
k-means
クラスタリング
サポートベクターマシン (SVM)

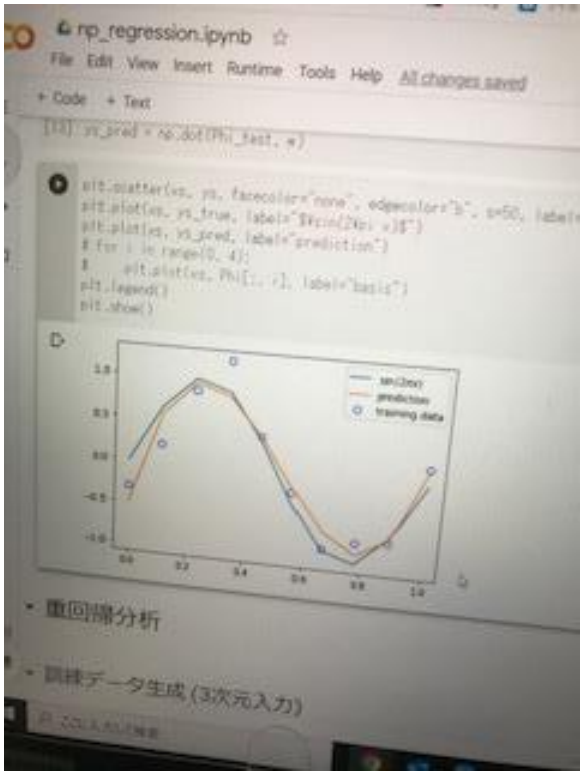

第4章 機械学習の基礎
1 b 2 C 3 C 4 A
5 ア A イ B ウ D
6 A
第5章 ―
第6章 モデルの評価・正則化・ハイパーパラメータ
1 B 2 C 3 D 4 C
5 ア C イ D ウ D
6 C 7 B
第7章
1 ア A イ B ウ C エ A
2 ア D イ C ウ A エ A
3 ア C イ D ウ D エ C
4 ア A イ D ウ A エ B
5 ア D イ A
6 ア C イ D


1 ヤA イb ウB エC
2 ヤC イC ウB オb
3 ヤC イB ウA エB
4 ヤB イA ウb オA カb
5 ヤb イA ウC
6 ヤA イb ウB エC

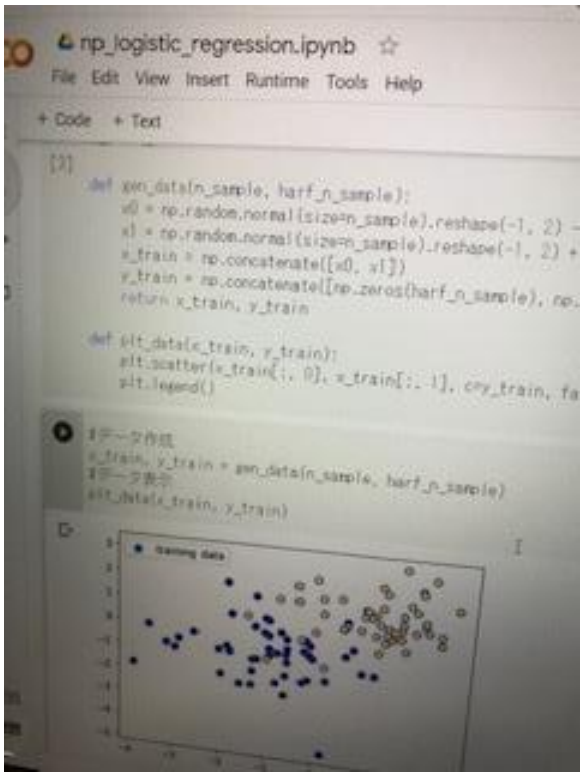

np_regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
xs: (100,)
ys_true: (100,)
ys: (100,)
2x + 5
training data

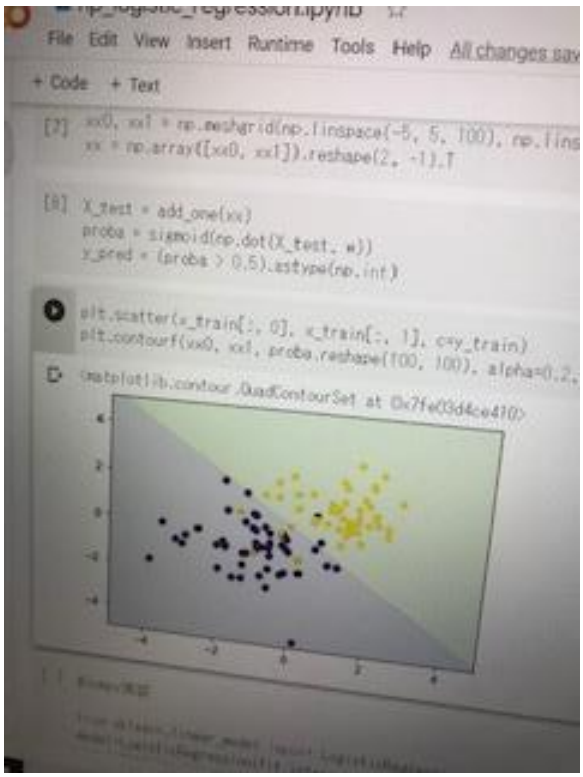

+ Code + Text
xs_new = np.linspace(0, 1, n_sample)
ys_pred = a * xs_new + b
plt.scatter(xs, ys, facecolor="none", edgecolor="b",
plt.plot(xs_new, ys_true, label="$2 x + 5$")
plt.plot(xs_new, ys_pred, label="prediction (a=[:.2],
plt.legend()
plt.show()
2x + 5
prediction (a=1.9, b=5.0)
training data
多項式回帰
訓練データ生成

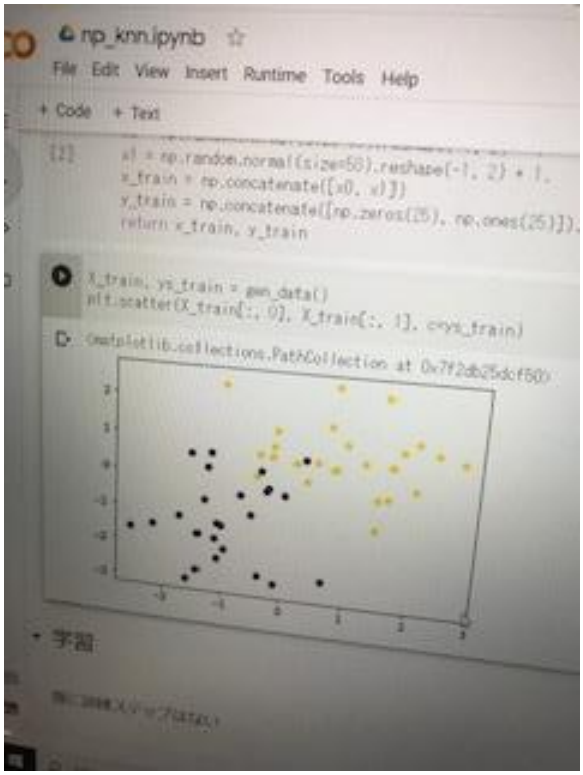

np_regression.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
print("xs: {}".format(xs.shape))
print("ys_true: {}".format(ys_true.shape))
print("ys: {}".format(ys.shape))
結果の描画
plot_result(xs, ys_true, ys)
xs: (10,)
ys_true: (10,)
ys: (10,)
sin(2πx)
training data
学習

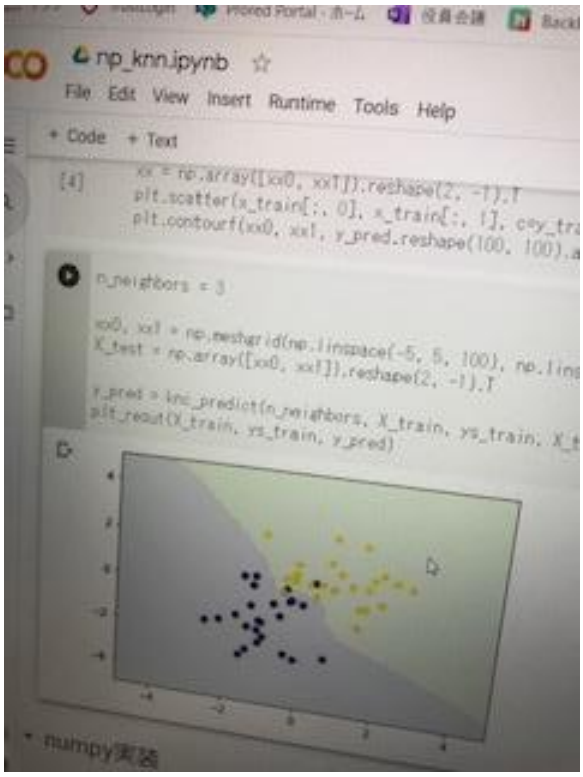

np_regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
training data
重回帰分析
訓練データ生成 (3次元入力)

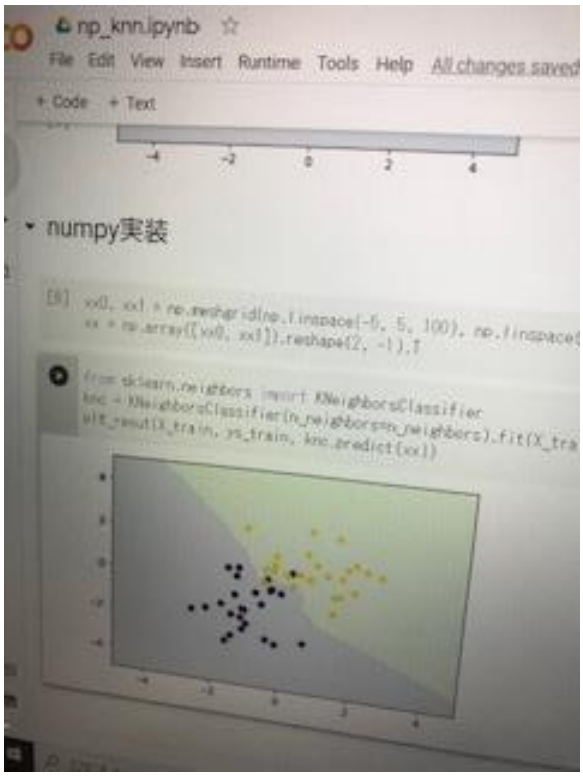

np_regression.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
予測
1 cell hidden
パラメータ推定結果

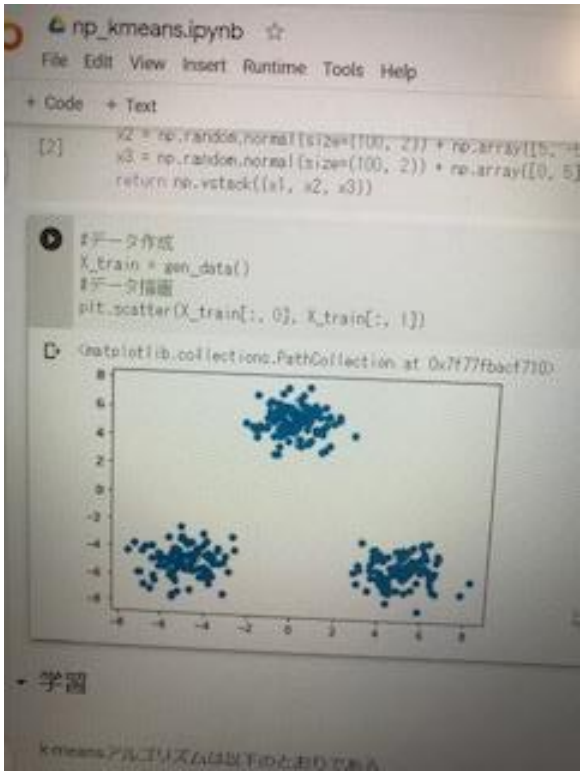

np_logistic_regression.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
def gen_data(n_sample, harf_n_sample):
x_train, y_train = gen_data(n_sample, harf_n_sample)
fit_data(x_train, y_train)
training data

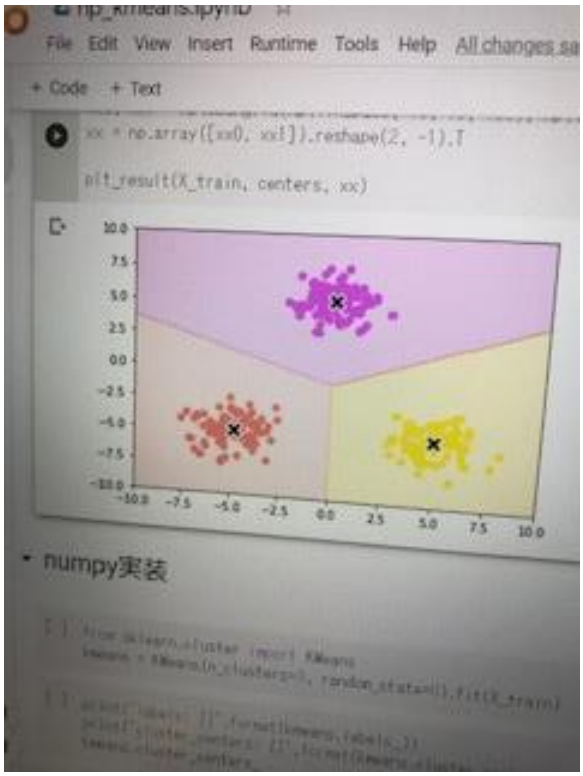

File Edit View Insert Runtime Tools Help All changes sav
+ Code + Text
X_test = add_one(xx)
proba = sigmoid(np.dot(X_test, w))
y_pred = (proba > 0.5).astype(np.int)
plt.scatter(x_train[:, 0], x_train[:, 1], c=y_train)
plt.contourf(xx0, xx1, proba.reshape(100, 100), alpha=0.2,
<matplotlib.contour.QuadContourSet at 0x7fe03d4ce410>

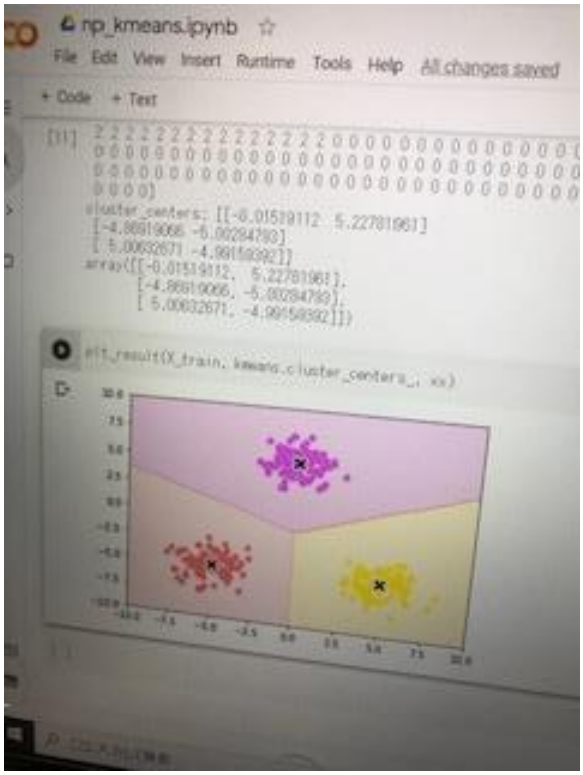

np_knn.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
[2]
x1 = np.random.normal(size=50).reshape(-1, 2) + 1.
x_train = np.concatenate([x0, x1])
y_train = np.concatenate([np.zeros(25), np.ones(25)])
return x_train, y_train
X_train, ys_train = gen_data()
plt.scatter(X_train[:, 0], X_train[:, 1], c=ys_train)
<matplotlib.collections.PathCollection at 0x7f2db25dcf50>
学習

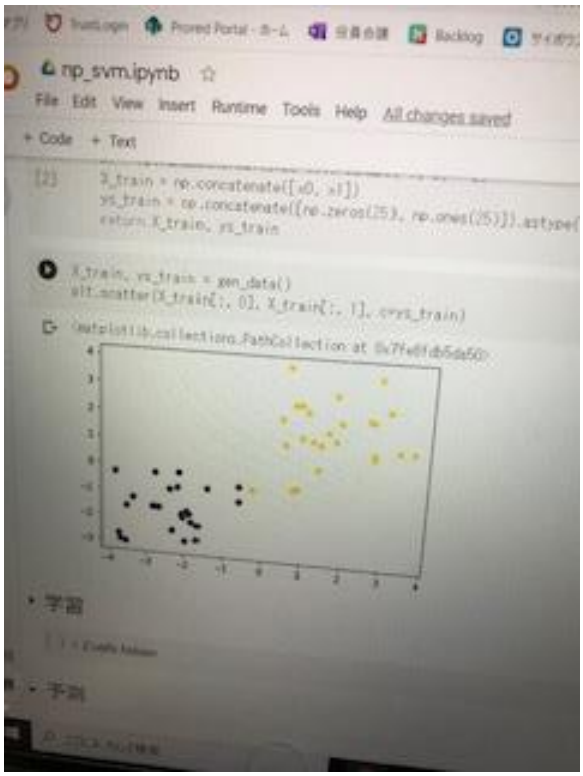

np_knn.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
[4]
xx = np.array([xx0, xx1]).reshape(2, -1).T
plt.scatter(x_train[:, 0], x_train[:, 1], c=y_tra
plt.contourf(xx0, xx1, y_pred.reshape(100, 100).a
n_neighbors = 3
xx0, xx1 = np.meshgrid(np.linspace(-5, 5, 100), np.lins
X_test = np.array([xx0, xx1]).reshape(2, -1).T
y_pred = knc.predict(n_neighbors, X_train, ys_train, X_t
plt_result(X_train, ys_train, y_pred)
numpy実装

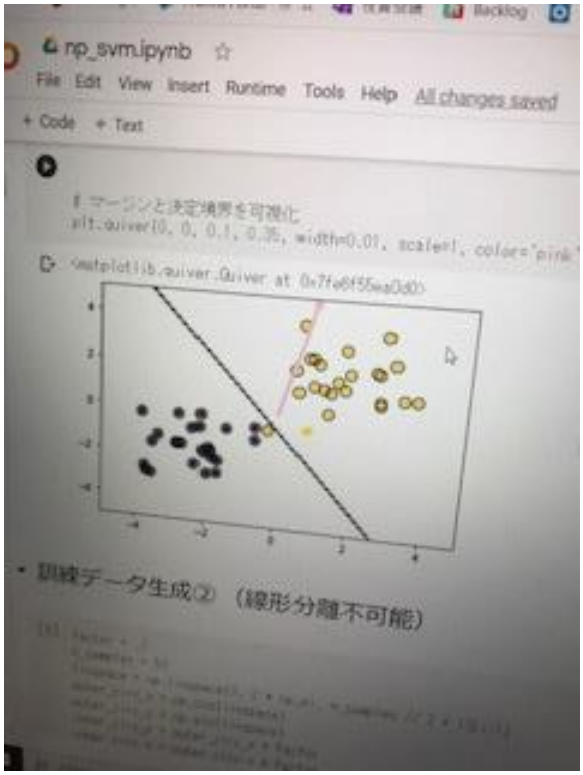

np_knn.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
numpy実装

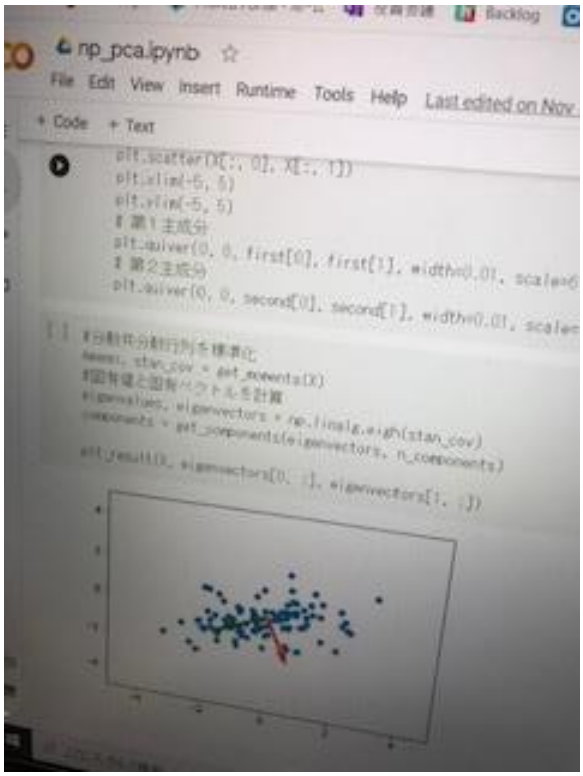

np_kmeans.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
学習

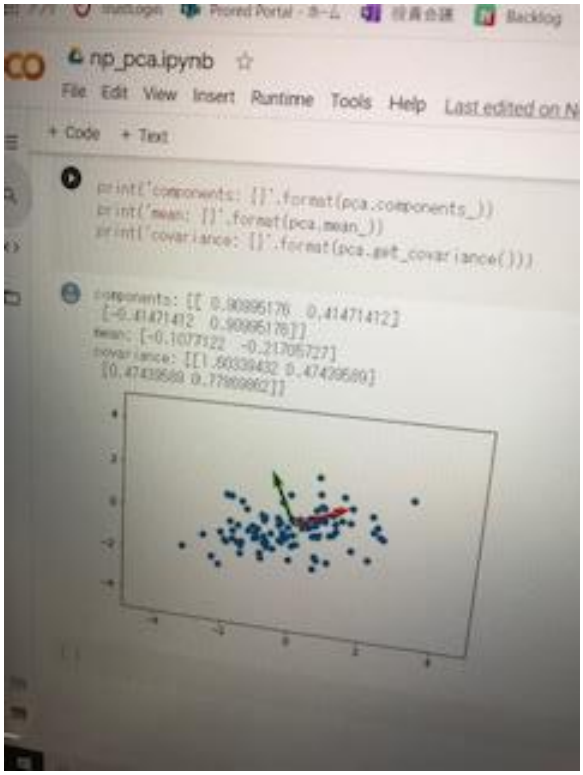

File Edit View Insert Runtime Tools Help
+ Code + Text
plt_result(X_train, centers, xx)
numpy実装


np_kmeans.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
cluster_centers: [[-0.01519112 5.22781961]
[-4.86919066 -5.00284793]
[5.00632671 -4.99158992]]
plt_result(X_train, kmeans.cluster_centers_, xx)


np_svm.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
X_train, ys_train = gen_data()
plt.scatter(X_train[:, 0], X_train[:, 1], c=ys_train)
・学習
・予測


np_svm.ipynb
File Edit View Insert Runtime Tools Help All changes saved
+ Code + Text
# マージンと決定境界を可視化
plt.quiver(0, 0, 0.1, 0.35, width=0.01, scale=1, color='pink'
・訓練データ生成② （線形分離不可能）


np_pca.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text


np_pca.ipynb
File Edit View Insert Runtime Tools Help
+ Code + Text
print('components: {}'.format(pca.components_))
print('mean: {}'.format(pca.mean_))
print('covariance: {}'.format(pca.get_covariance()))