COMP9417 – Machine Learning

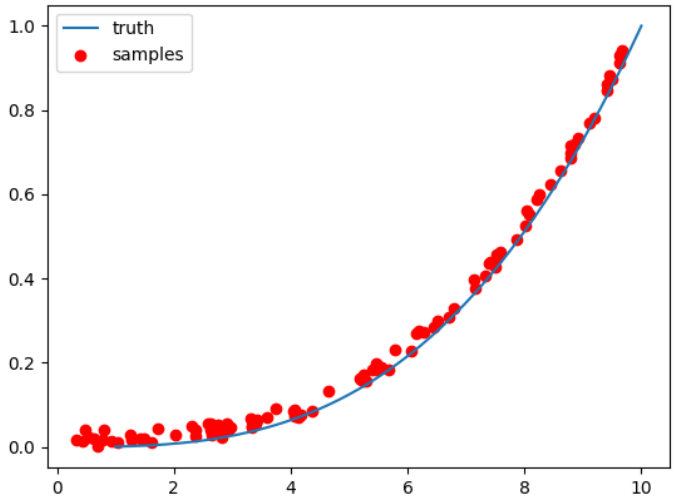
Tutorial: Ensemble Methods

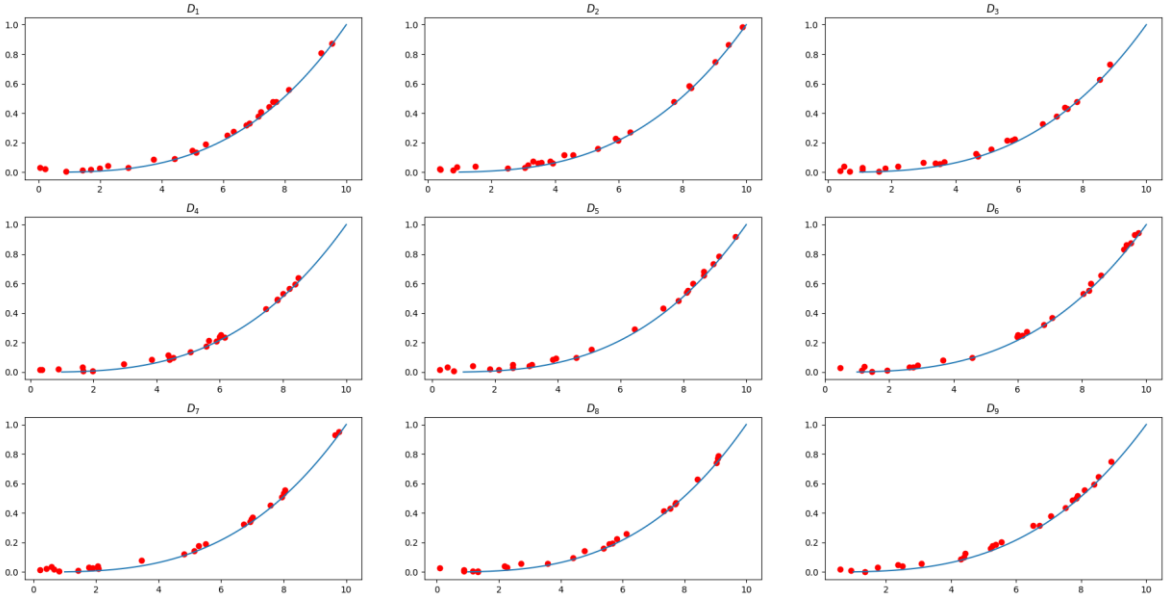
z5325987 - WanqingYang

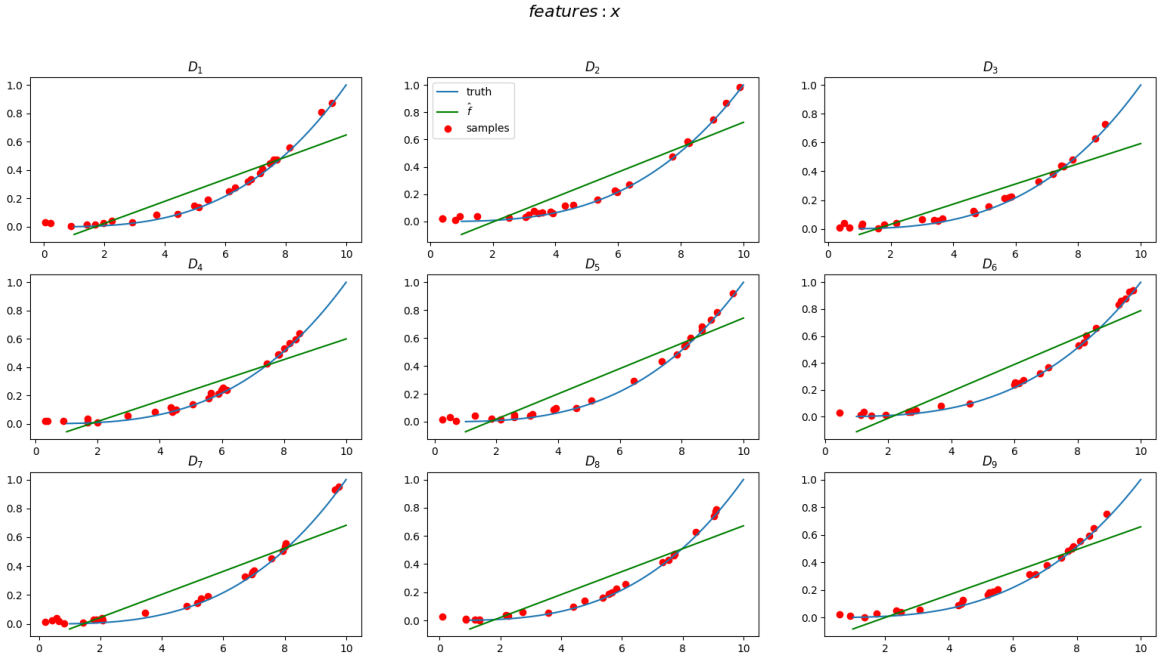
Exercise 1.

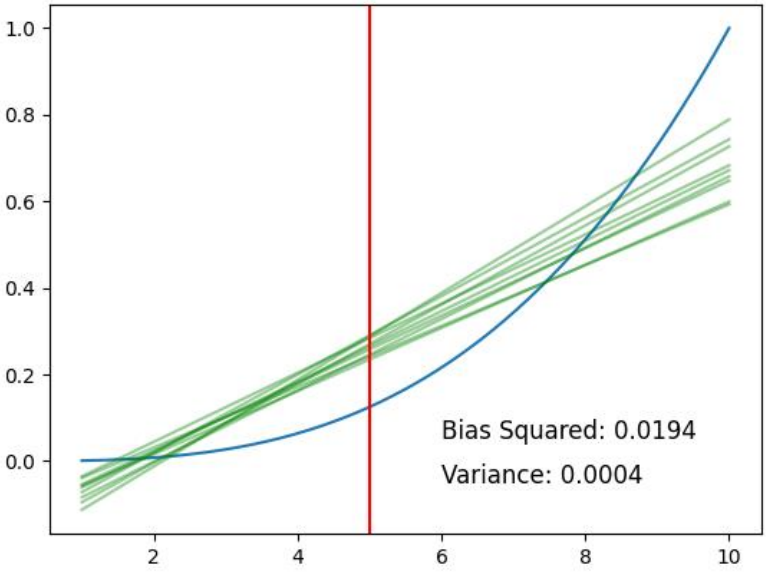
```
def f_sampler(f, n=100, sigma=0.1):
    # sample points from function f with Gaussian noise (0,sigma**2)
    xvals = np.random.uniform(0, 10, n) #
    epsilon = np.random.uniform(0, sigma * sigma, n)#
    yvals = f(xvals) + epsilon #

    # build dataset D
    D = np.zeros(shape=(n, 2))
    D[:, 0] = xvals;
    D[:, 1] = yvals;
```





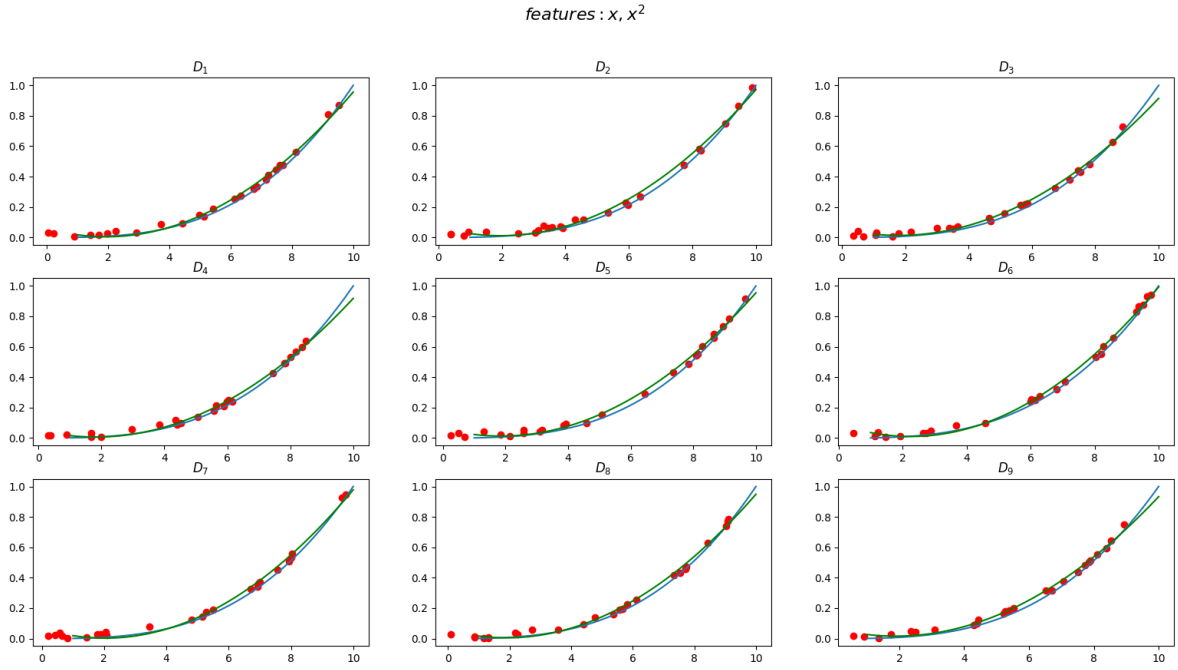

features : x
D_1
D_2
D_3
D_4
D_5
D_6
D_7
D_8
D_9
truth
$\hat{f}$
samples

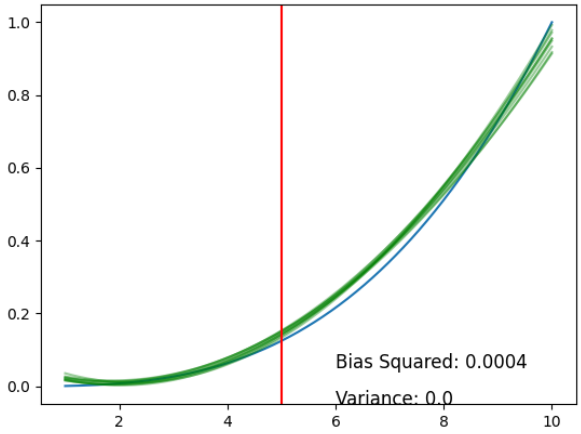

Bias Squared: 0.0194
Variance: 0.0004

Exercise 2.

```
mods = np.zeros((9, 3))  # store models
for i, ax in enumerate(ax.flat):
    ax.plot(xx, f(xx), label="truth")
    fsamples = f_sampler(f, 25, sigma=0.2)#
    ax.scatter(*fsamples.T, color="red", label="samples")
    ax.set_title(f"$D_{i + 1}$")

    # build model
    X = fsamples[:,0].reshape(-1, 1)#
    y = fsamples[:,1].reshape(-1, 1)#
    # your code here, Create dataset with extra features
    q_feature = PolynomialFeatures(degree=2)#
    X_train_q = q_feature.fit_transform(X)#
    mod = LinearRegression().fit(X_train_q, y)#
    lr = lambda x: mod.intercept_[0] + mod.coef_[0][1] * x + mod.coef_[0][2] * x ** 2
    mods[i][0] = mod.intercept_[0]
    for j in range(1, len(mod.coef_[0])):
        mods[i][j] = mod.coef_[0][j]
    ax.plot(xx, lr(xx), color="green", label="$\\hat{f}$")
fig.suptitle("$features: x, x^2$", fontsize=16)
plt.show()
```

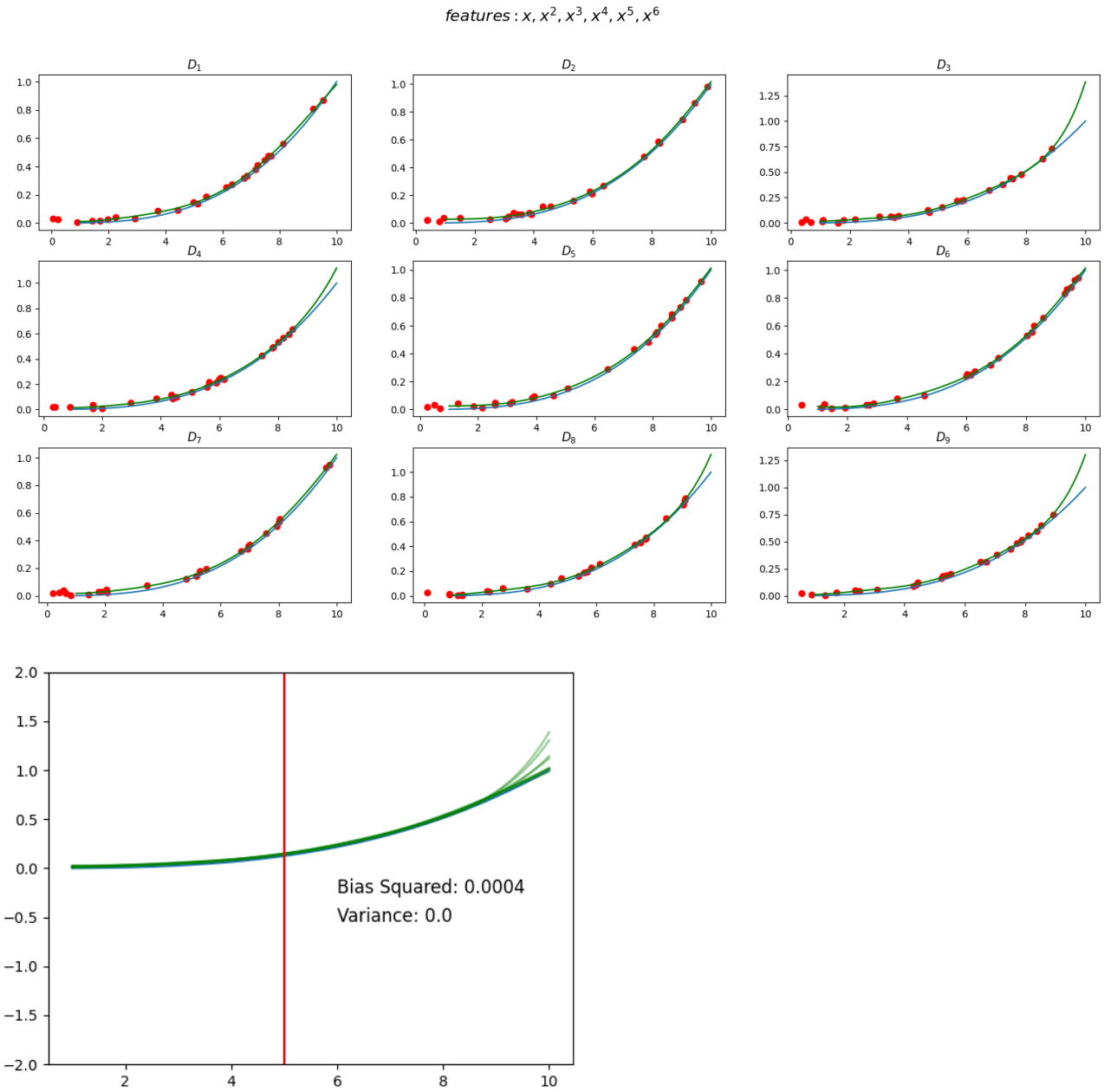




Exercise 3.

```
mods = np.zeros((9, 7))  # store models
for i, ax in enumerate(ax.flat):
    ax.plot(xx, f(xx), label="truth")
    fsamples = f_sampler(f, 25, sigma=0.2)
    ax.scatter(*fsamples.T, color="red", label="samples")
    ax.set_title(f"$D_{i + 1}$")

    # build model
    X = fsamples[:, 0].reshape(-1, 1)  #
    y = fsamples[:, 1].reshape(-1, 1)  #
    # your code here, Create dataset with extra features
    q_feature = PolynomialFeatures(degree=6)  #
    X_train_q = q_feature.fit_transform(X)  #
    mod = LinearRegression().fit(X_train_q, y)  #
    lr = lambda x: mod.intercept_[0] + mod.coef_[0][1] * x + mod.coef_[0][2] * x ** 2 + mod.coef_[0][3] * (x ** 3) + \
                   mod.coef_[0][4] * (x ** 4) + mod.coef_[0][5] * (x ** 5) + mod.coef_[0][6] * (x ** 6)
    mods[i][0] = mod.intercept_[0]
    for j in range(1, len(mod.coef_[0])):
        mods[i][j] = mod.coef_[0][j]
    # your code here
    ax.plot(xx, lr(xx), color="green", label="$\\hat{f}$")
```


features : $x, x^2, x^3, x^4, x^5, x^6$
D_1
D_2
D_3
D_4
D_5
D_6
D_7
D_8
D_9

Bias Squared: 0.0004
Variance: 0.0