תרגיל בית 4 – למידה חישובית 1
096411

<u>שמות המגישים</u>:
רוני פרידמן, ת.ז: 205517097
מור לוינבוק, ת.ז: 205451123

שאלה 1:

א. $S^{(i)}$ – קבוצת כל הדוגמאות מסט האימון, כאשר החלפנו את הדוגמא ה(x_i, y_i) בדוגמא (x', y').

$A(S)$ – ההיפותזה (כלל ההחלטה) שהאלגוריתם A החזיר לנו על ידי מציאת המינימום של $f_s(w)$ עבור סט האימון S הנתון.

ב.

$$f_s(v) = L_s(v) + \lambda\|v\|_2^2 = \frac{1}{m}\sum_{i=1}^{m} l(v, x_i, y_i) + \lambda\|v\|_2^2 =$$

$$= \frac{1}{m}\left(\left(\sum_{i=1}^{i-1} l(v, x_i, y_i)\right) + l(v, x', y') + \left(\sum_{i=i+1}^{m} l(v, x_i, y_i)\right) - l(v, x', y') + l(v, x_i, y_i)\right) + \lambda\|v\|_2^2 =$$

$$= \frac{1}{m}\left(L_{S^{(i)}}(v) - l(v, x', y') + l(v, x_i, y_i)\right) + \lambda\|v\|_2^2 = \frac{1}{m}L_{S^{(i)}}(v) + \lambda\|v\|_2^2 + \frac{l(v, x_i, y_i) - l(v, x', y')}{m}$$

לכן מתקיים:

$$f_s(v) - f_s(u) = \frac{1}{m}L_{S^{(i)}}(v) + \lambda\|v\|_2^2 + \frac{l(v, x_i, y_i) - l(v, x', y')}{m} - \left(\frac{1}{m}L_{S^{(i)}}(u) + \lambda\|u\|_2^2 + \frac{l(u, x_i, y_i) - l(u, x', y')}{m}\right) =$$

$$= \left(\frac{1}{m}L_{S^{(i)}}(v) + \lambda\|v\|_2^2\right) - \left(\frac{1}{m}L_{S^{(i)}}(u) + \lambda\|u\|_2^2\right) + \frac{l(v, x_i, y_i) - l(u, x_i, y_i)}{m} + \frac{l(u, x', y') - l(v, x', y')}{m}$$

כנדרש.

ג. מסעיף ב' מתקיים:

$$f_s\left(A\left(S^{(i)}\right)\right) - f_s\left(A(S)\right) = \frac{1}{m}L_{S^{(i)}}\left(A\left(S^{(i)}\right)\right) - \frac{1}{m}L_{S^{(i)}}\left(A(S)\right) + \frac{l\left(A\left(S^{(i)}\right), x_i, y_i\right) - l\left(A(S), x_i, y_i\right)}{m} + \frac{l\left(A(S), x', y'\right) - l\left(A\left(S^{(i)}\right), x', y'\right)}{m}$$

אנו יודעים כי:

$$\frac{1}{m}L_{S^{(i)}}\left(A\left(S^{(i)}\right)\right) \le \frac{1}{m}L_{S^{(i)}}\left(A(S)\right)$$

וזאת כי A מחזיר לנו את המינימום על ממוצע ה*loss* של סט האימון הנבדק. כלומר ההיפותזה $A\left(S^{(i)}\right)$ מביאה לנו את המינימום על $L_{S^{(i)}}$ ולכן ממוצע ה*loss* שמתקבל על ידי ההיפותזה $A(S)$ יכול רק לגדול.

אם כך, מתקיים:

$$f_s\left(A\left(S^{(i)}\right)\right) - f_s\left(A(S)\right) \le \frac{l\left(A\left(S^{(i)}\right), x_i, y_i\right) - l\left(A(S), x_i, y_i\right)}{m} + \frac{l\left(A(S), x', y'\right) - l\left(A\left(S^{(i)}\right), x', y'\right)}{m}$$

ד. אנו יודעים כי הפונקציה $g(w)=\lambda\|w\|^2$ היא קמורה חזק בדרגה 2λ.
בנוסף אנו יודעים כי הפונקציה $h(S,w)=\frac{1}{m}L_S(w)$ היא קמורה.
ממשפט, אנו יודעים כי הפונקציה $f(S,w)=h(S,w)+g(w)$ גם היא קמורה חזק בדרגה 2λ.

לפי משפט, אנו יודעים כי בהנתן S,ובגלל שf קמורה חזק בדרגה 2λ,עבור u שממזער את f מתקיים:

$$f(S,w)-f(S,u)\geq\frac{2\lambda}{2}\|w-u\|^2=\lambda\|w-u\|^2$$

לכל w.
נציב $u=A(S),w=A\left(S^{(i)}\right)$ ונקבל:

$$\lambda\left\|A\left(S^{(i)}\right)-A(S)\right\|^2=\lambda\left\|A(S)-A\left(S^{(i)}\right)\right\|^2\leq f_S\left(A\left(S^{(i)}\right)\right)-f_S\big(A(S)\big)$$

היות ו$A(S)$ הממזער של הפונק' f_S.
טענה זו בשילוב עם סעיף ג' נותנת לנו:

$$\lambda\left\|A\left(S^{(i)}\right)-A(S)\right\|^2\leq\frac{l\left(A\left(S^{(i)}\right),x_i,y_i\right)-l\left(A(S),x_i,y_i\right)}{m}+\frac{l\left(A(S),x',y'\right)-l\left(A\left(S^{(i)}\right),x',y'\right)}{m}$$

ה. נניח כי פונק' ה$loss$ היא ρ ליפשיצית. מעובדה זו, ומחוקי הערך המוחלט, בהנתן (x,y):

$$l\left(A\left(S^{(i)}\right),x_i,y_i\right)-l\left(A(S),x_i,y_i\right)\leq\left|l\left(A\left(S^{(i)}\right),x_i,y_i\right)-l\left(A(S),x_i,y_i\right)\right|\leq\rho\left\|A\left(S^{(i)}\right)-A(S)\right\|$$

באופן דומה:

$$l\left(A(S),x',y'\right)-l\left(A\left(S^{(i)}\right),x',y'\right)\leq\left|l\left(A\left(S^{(i)}\right),x',y'\right)-l\left(A(S),x',y'\right)\right|\leq\rho\left\|A\left(S^{(i)}\right)-A(S)\right\|$$

נציב באי השיוויון שמצאנו בסעיף ד':

$$\lambda\left\|A\left(S^{(i)}\right)-A(S)\right\|^2\leq\frac{\rho\left\|A\left(S^{(i)}\right)-A(S)\right\|}{m}+\frac{\rho\left\|A\left(S^{(i)}\right)-A(S)\right\|}{m}=\frac{2\rho\left\|A\left(S^{(i)}\right)-A(S)\right\|}{m}$$

עבור המקרה בו $A\left(S^{(i)}\right)=A(S)$ קל לראות כי מתקיים שוויון 0=0 בין האגפים.
במקרה כי הם שונים, מתקבל:

$$\left\|A\left(S^{(i)}\right)-A(S)\right\|\leq\frac{2\rho}{\lambda m}$$

ו. כפי שהראנו בסעיף ה':

$$l\left(A\left(S^{(i)}\right),x_i,y_i\right)-l\left(A(S),x_i,y_i\right)\leq\left|l\left(A\left(S^{(i)}\right),x_i,y_i\right)-l\left(A(S),x_i,y_i\right)\right|\leq\rho\left\|A\left(S^{(i)}\right)-A(S)\right\|$$

ולפי המסקנה שהוכחנו בסעיף ה':

$$l\left(A\left(S^{(i)}\right),x_i,y_i\right)-l\left(A(S),x_i,y_i\right)\leq\rho\left\|A\left(S^{(i)}\right)-A(S)\right\|\leq\frac{2\rho^2}{\lambda m}$$

ז. $L_S(w)$ – השגיאה האמפירית. זהו ממוצע השגיאות (שמוגדרות על ידי פונקצ' ה-$loss$) שההיפותזה w ביצעה על סט האימון S. בהנתן סט אימון S ו-w, ניתן לחשב את ערך הביטויי, אם נגדיר את פונק' ה$loss$ אז כל הנתונים לחישוב הביטויי יהיה ברשותנו.

$L_D(w)$ – תוחלת השגיאה על כל הדוגמאות שהגיעו מההתפלגות D. זוהי תוחלת השגיאות שהמסווג שלנו יבצע על כל דוגמא שאי פעם הגיעה או תגיע מהתפלגות D. כמובן, שלא ניתן לחשב את ביטויי זה בהנתן סט אימון S והיפותזה w, כיוון שסט האימון הוא רק מדגם מייצג, ובשביל לחשב את L_D במדוייק נצטרך או לעבור על כל

הדוגמאות בהסטוריה, או לדעת במפורש מהיא התפלגות *D* – שני מאורעות שיקרו בהסתברות אפסית גם אם ממש נתאמץ.

ח. לפי המשפט שנלמד בהרצאה (מצגת 6 שקף 18):

$$\mathbb{E}_{S\sim D^m}\left[L_D\big(A(S)\big)-L_S\big(A(S)\big)\right]=\mathbb{E}_{(S,z')\sim D^{m+1},i\sim U(m)}\left[l\big(A\big(S^{(i)}\big),x_i,y_i\big)-l\big(A(S),x_i,y_i\big)\right]$$

ולפי סעיף ו' מתקיים:

$$\mathbb{E}_{(S,z')\sim D^{m+1},i\sim U(m)}\left[l\big(A\big(S^{(i)}\big),x_i,y_i\big)-l\big(A(S),x_i,y_i\big)\right]\le\mathbb{E}_{(S,z')\sim D^{m+1},i\sim U(m)}\left[\frac{2\rho^2}{\lambda m}\right]=\frac{2\rho^2}{\lambda m}$$

היות ואין לנו משתנים מקריים יותר בתוך ביטויי התוחלת. אם כך מתקיים:

$$\mathbb{E}_{S\sim D^m}\left[L_D\big(A(S)\big)-L_S\big(A(S)\big)\right]\le\frac{2\rho^2}{\lambda m}$$

שאלה 2:

א.

$$recall=\frac{TP}{TP+FN},\qquad precision=\frac{TP}{TP+FP}\ ,\qquad FPR=\frac{FP}{FP+TN}$$

Recall – מספר האנשים שסיווגנו כ*Positive* מתוך כל קבוצת האנשים שה*label* שלהם הוא אכן *Positive*.

Precision – מספר הסיווגים הנכונים שסיווגנו כ*Positive* מתוך כל קבוצת האנשים שסיווגנו כ*Positive*.

FPR – קבוצת האנשים שסיווגנו *Positive* מתוך קבוצת כל האנשים שה*label* האמיתי שלהם הוא *Negetive*.

ב. דוגמא טובה תהייה מחלה סופנית – *recall* גבוה אומר שאנחנו שמים דגש על לאבחן כל חולה שאכן חולה במחלה, כמאומת, בשלב מוקדם ככל האפשר.
אם ה*precision* כתוצאה מכך יורד (דבר שעלול לקרות אם אנחנו מחמירים בסיווג לכיוון ה*Positive*) אז הנזק יהיה הרבה פחות נוראי כי בעצם יהיו יותר אנשים שיחשבו שהם חולים, ואחר כך יתברר שהם לא.

ג. סקר שוק לגבי מכירה של מוצר ללקוחות. אנו נרצה לייצר את המוצר תוך כדי התחשבות בכמות האנשים שאכן ירצו לקנות אותו, על מנת לא לצאת בהפסד בעלות הייצור.
במצב זה נרצה *Precision* גבוה, היות וזה אומר שנוכל לייצר כמות מדוייקת של מוצרים לפי ה*Precision*.
בעקבות כך, כניראה, ה*recall* שלנו יירד, וזה אומר שנפספס כמה קונים פוטנציאלים בסקר, מתוך קבוצת כל הקונים שאכן היו קונים את המוצר – דבר שלא יעלה לנו ביותר מידי כסף על עלות הייצור.

ד.

$$P_w(y=1|x)=\frac{e^{w^Tx}}{1+e^{w^Tx}}=\frac{e^{-0.3*1-0.5x_1+0.5x_2}}{1+e^{-0.3*1-0.5x_1+0.5x_2}}$$

$$P_w(y=1|x_1)=\frac{e^{-3.3}}{1+e^{-3.3}}\approx 0.0355,\qquad P_w(y=1|x_2)=\frac{e^{0.7}}{1+e^{0.7}}\approx 0.6682$$

$$P_w(y=1|x_3)=\frac{e^{-7.3}}{1+e^{-7.3}}\approx 6*10^{-4},\qquad P_w(y=1|x_4)=\frac{e^{-1.3}}{1+e^{-1.3}}\approx 0.214$$

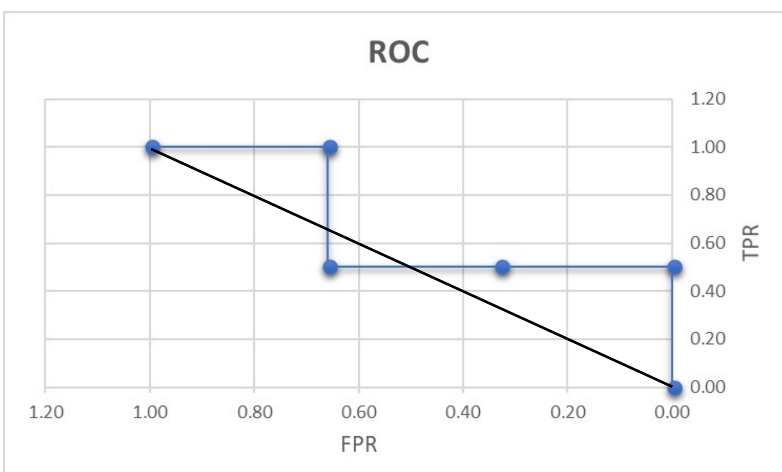
$$P_w(y=1|x_5) = \frac{e^{-8.8}}{1+e^{-8.8}} \approx 1.5 * 10^{-4}$$

ה.

Trheshold	x1	x2	x3	x4	x5
$1.5 * 10^{-5}$	1	1	1	1	1
$1.5 * 10^{-3}$	1	1	1	1	0
$6 * 10^{-3}$	1	1	0	1	0
0.04	0	1	0	1	0
0.3	0	1	0	0	0
0.7	0	0	0	0	0
lable	0	1	1	0	0

					x	y
Trheshold	TP	TN	FP	FN	TPR	FPR
$1.5 * 10^{-5}$	2	0	3	0	1.00	1.00
$1.5 * 10^{-3}$	2	1	2	0	1.00	0.66
$6 * 10^{-3}$	1	1	2	1	0.50	0.66
0.04	1	2	1	1	0.50	0.33
0.3	1	3	0	1	0.50	0.00
0.7	0	3	0	2	0.00	0.00

הגרף שהתקבל:



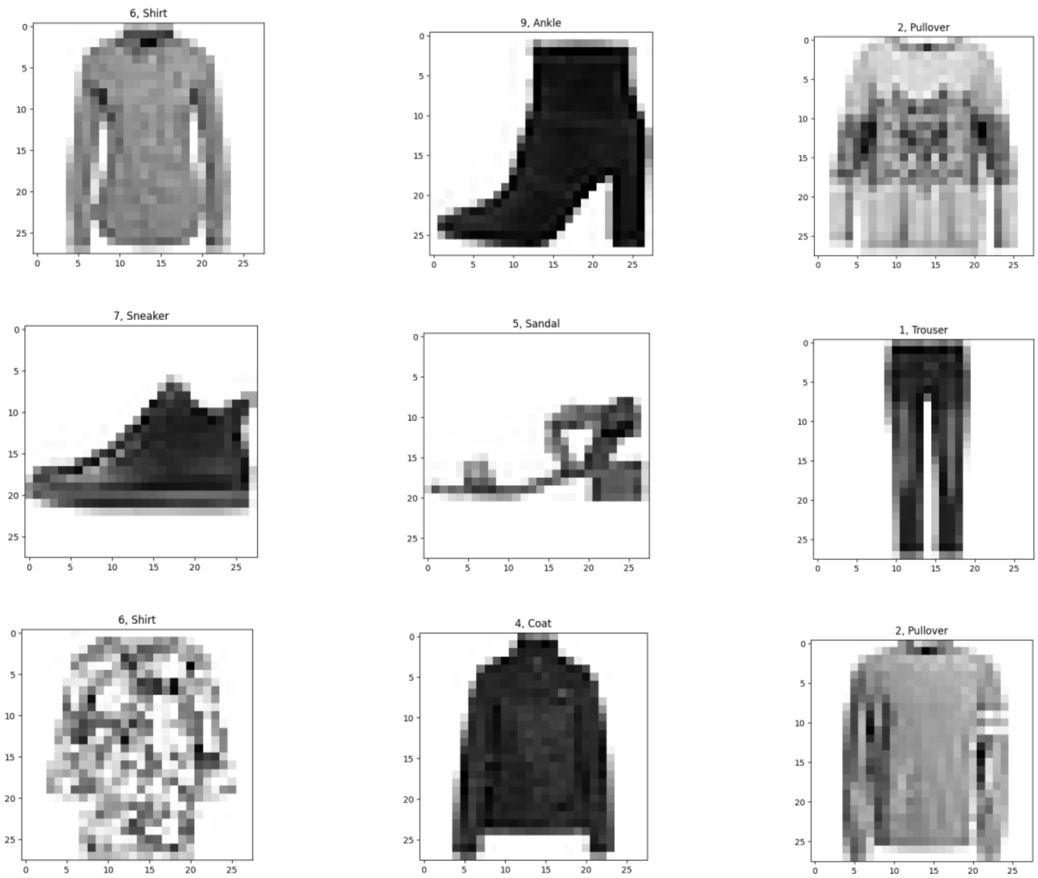
הקו השחור הוא המודל האקראי.

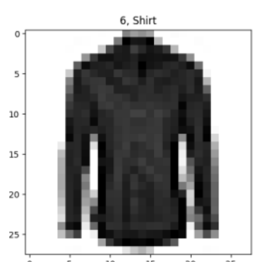
ניתן לראות כי עבור ערכי *Threshold* $\theta \in [0,0.5]$ השגיאה של המודל שלנו תהייה נמוכה יותר. ניתן לראות כי אנו נקבל *TPR* גבוה יותר עבור כל ערך *FPR* נתון. ולכן נוכל להגיד שהמודל שלנו טוב יותר.

<u>שאלה 3:</u>

א. הקוד כולו מצורף בסוף המסמך.

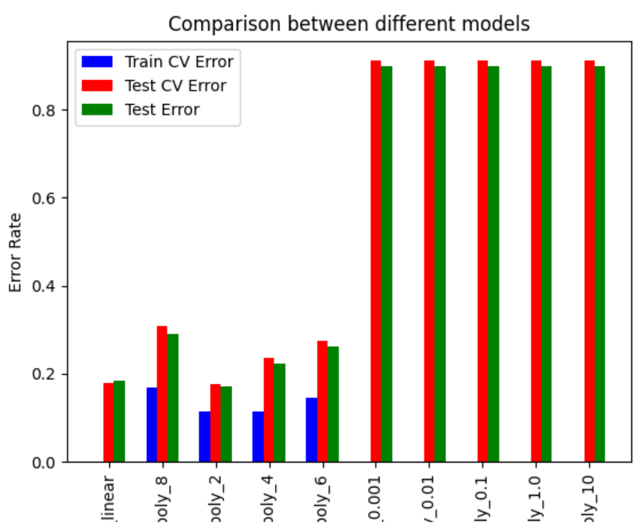
ב. התמונות שהתקבלו:





ג. הקוד מצורף בסוף המסמך.

ד. הגרף שהתקבל:



ניתן לראות כי המודל שהיה הכי מוצלח לפי שיטת הCV הינו המודל *SVM_Linear* היות ושגיאת האימון היא 0 ושגיאת הולידציה הכי נמוכה. למרות זאת, שגיאת הולידציה של *SVM_poly_2* זהה גם היא לשגיאת הולידציה שקיבלנו ב *SVM_Linear*.

שגיאת המבחן הנמוכה ביותר ללא *CV* התקבלה עבור *SVM_poly_2*, ולכן נוכל לומר ששיטת ה*CV* אכן איתרה את המודל בעל שגיאת המבחן הנמוכה ביותר (או לפחות, אחד מהם).

```python
import numpy as np
from sklearn.datasets import fetch_openml
from sklearn.svm import SVC
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt


def cross_validation_error(X, y, model, folds):
    samples_sub_arrays = np.array_split(X, folds)
    labels_sub_arrays = np.array_split(y, folds)
    error_k_train = []
    error_k_test = []
    for k in range(folds):
        # Split the array into sub-arrays of approximately the same size.
        # Choose the k-th array as the test array, and combine all others as the training set.
        X_test = samples_sub_arrays.pop(k)
        y_test = labels_sub_arrays.pop(k)
        X_train = np.vstack(samples_sub_arrays)
        y_train = np.concatenate(labels_sub_arrays)
        model.fit(X_train, y_train)
        # Calculate error on training set
        y_pred_train = model.predict(X_train)
        error_count = 0
        for i in range(len(y_train)):
            if y_pred_train[i] != y_train[i]:
                error_count += 1
        error_k_train.append(error_count / len(y_train))
        # Calculate error on test set
        y_pred_test = model.predict(X_test)
        error_count = 0
        for i in range(len(y_pred_test)):
            if y_pred_test[i] != y_test[i]:
                error_count += 1
        error_k_test.append(error_count / len(y_test))
        # Return the chosen sub-arrays back with all training points
        samples_sub_arrays.insert(k, X_test)
        labels_sub_arrays.insert(k, y_test)
    avg_error = (sum(error_k_train)/folds, sum(error_k_test)/folds)
    return avg_error


def svm_results(X_train, y_train, X_test, y_test):
    error_results = {}
    error = 0
    model = SVC(kernel='linear')
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)
    for i in range(len(y_test)):
        if y_test[i] != y_pred[i]:
            error = error + 1
    test_error = error/len(y_test)
    CV_error = cross_validation_error(X_train, y_train, model, 4)
    error_results['SVM_linear'] = (CV_error[0], CV_error[1], test_error)
    d_list = {2, 4, 6, 8}
    for d in d_list:
        model = SVC(kernel='poly', degree=d)
        error = 0
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
```

```
        for i in range(len(y_test)):
            if y_test[i] != y_pred[i]:
                error = error + 1
        test_error = error/len(y_test)
        CV_error = cross_validation_error(X_train, y_train, model, 4)
        error_results[f'SVM_poly_{d}'] = (CV_error[0], CV_error[1], test_error)
    rbf_values = [0.001, 0.01, 0.1, 1.0, 10]
    for val in rbf_values:
        model = SVC(kernel='rbf', gamma=val)
        error = 0
        model.fit(X_train, y_train)
        y_pred = model.predict(X_test)
        for i in range(len(y_test)):
            if y_test[i] != y_pred[i]:
                error = error + 1
        test_error = error / len(y_test)
        CV_error = cross_validation_error(X_train, y_train, model, 4)
        error_results[f'SVM_poly_{val}'] = (CV_error[0], CV_error[1], test_error)
    return error_results


def fetch_mnist():
    # Download MNIST dataset
    X, y = fetch_openml('Fashion-MNIST', version=1, return_X_y=True)
    X = X.to_numpy()
    y = y.to_numpy()

    # Randomly sample 7000 images
    np.random.seed(2)
    indices = np.random.choice(len(X), 7000, replace=False)
    X, y = X[indices], y[indices]
    return X, y


def main():
    X, y = fetch_mnist()
    print(X.shape, y.shape)
    idx2class = {'0': 'T-shirt/top', '1': 'Trouser', '2': 'Pullover', '3': 'Dress', '4': 'Coat', '5': 'Sandal',
                 '6': 'Shirt', '7': 'Sneaker', '8': 'Bag', '9': 'Ankle'}
    for i in range(10):
        image = X[i].reshape(28, 28)
        s = idx2class[y[i]]
        plt.title(f'{y[i]}, {s}')
        plt.imshow(image, cmap='binary')
        plt.show()

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)
    error_res = svm_results(X_train, y_train, X_test, y_test)

    train_cv_error = []
    test_cv_error = []
    test_error = []
    labels_names = []
    index_range = range(len(error_res.keys()))
    for key in error_res.keys():
        train_cv_error.append(error_res[key][0])
```

```
        test_cv_error.append(error_res[key][1])
        test_error.append(error_res[key][2])
        labels_names.append(key)

    fig1, ax1 = plt.subplots()
    ax1.set_xlabel("Model")
    ax1.set_ylabel("Error Rate")
    width = 0.2
    ax1.bar([val - width for val in index_range], train_cv_error, width=width
, color='blue', label='Train CV Error')
    ax1.bar([val for val in index_range], test_cv_error, width=width, color='
red', label='Test CV Error')
    ax1.bar([val + width for val in index_range], test_error, width=width,
color='green', label='Test Error')
    ax1.set_xticks(index_range, labels_names, rotation=90)
    plt.title("Comparison between different models")
    plt.gcf().subplots_adjust(bottom=0.3)
    plt.legend()
    plt.show()



if __name__ == '__main__':
    main()
```