ממ"ן 12 מבוא לראייה ממוחשבת

שם: נועם שדה

תאריך: 16.12.2022

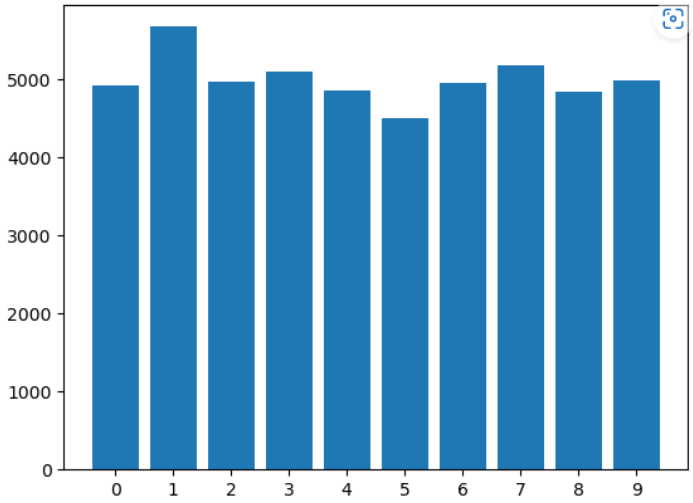
הקדמה:

יש לוודא שהקובץ mnist.pkl נמצא בתיקייה עם קובץ הפייתון, וגם התיקייה images עם כל התמונות של spatial_envelope.

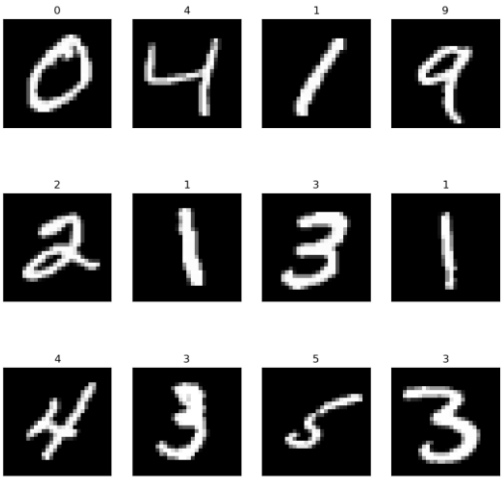
מומלץ להריץ את הקובץ main.ipynb .

נציג כמה תמונות יש מכל ספרה:

```
0: 4932
1: 5678
2: 4968
3: 5101
4: 4859
5: 4506
6: 4951
7: 5175
8: 4842
9: 4988
```



נציג את 12 התמונות הראשונות בסט הנתונים:



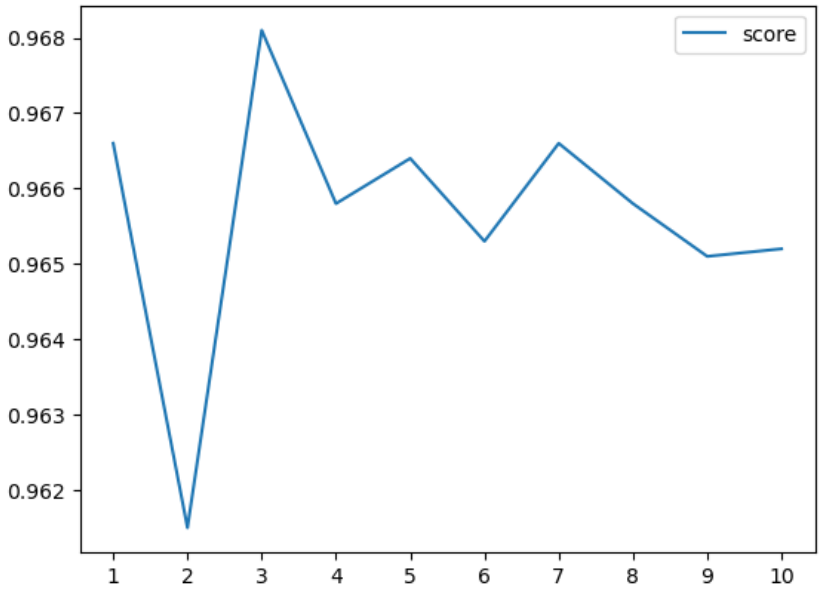
חלק 1 - MNIST

שאלה 1:

א. חשבו את ביצועי מסווג KNN עבור . k=1...10 ציירו את התוצאות על גרף.
מה התוצאה האופטימלית? איך מתבצעת ההחלטה כאשר k זוגי?

ביצעתי חישוב knn עבור אינדקסים מ-1 עד 10.
ביצעתי אימון מה dataset על ה- train
בדקתי אץ האלגוריתם על ה- test
כאשר K זוגי, ניקח את הרעך הראשון שהגענו אליו (מה שהופיע ראשון בסט של האימון). ניתן גם לפתור בעיה זו בלבצע בחירה רנדומלית של אחת הקטגוריות, לקחת את הקטגוריה הראשונה שמצאנו או לעלות\להקטין את מספר הK עד שנקודה זו תסווג ללא מצב של שיווין.

התוצאות:



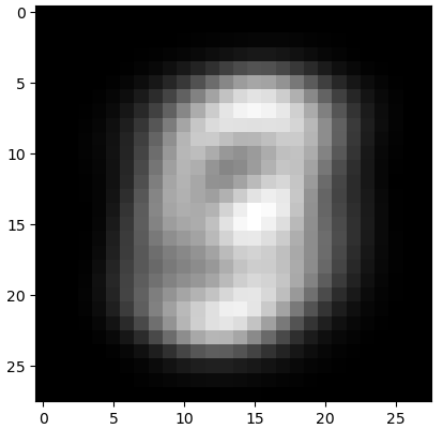
על פי הגרף התוצאה האופטימלית היא עבור k=3 .

שאלה 2:

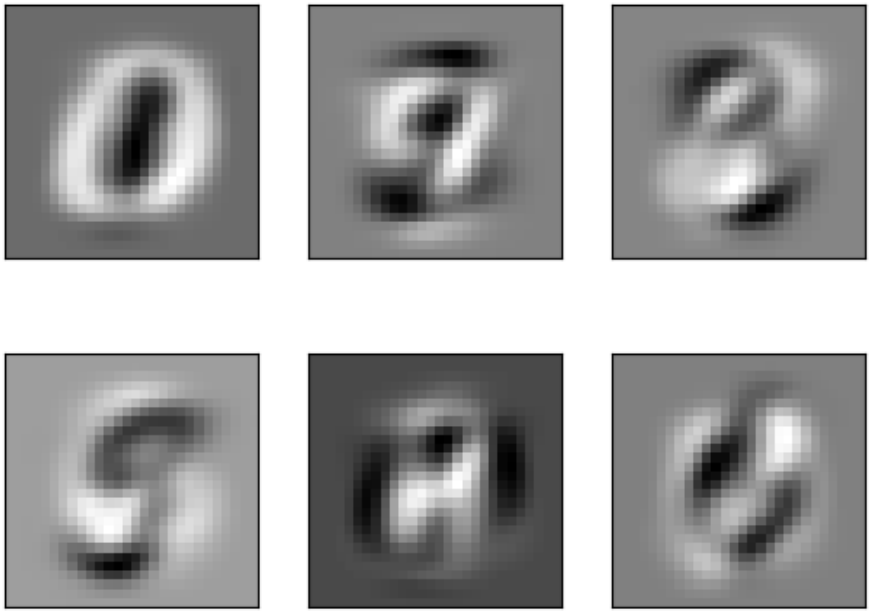
א. חשבו את טרנספורמציית PCA על סט האימון. ציירו את הסיפרה הממוצעת ואת 6 ה principle components - הראשונים.

חישבתי את טרנספורמציית PCA על סט האימון.

ציור של הספרה הממוצעת:

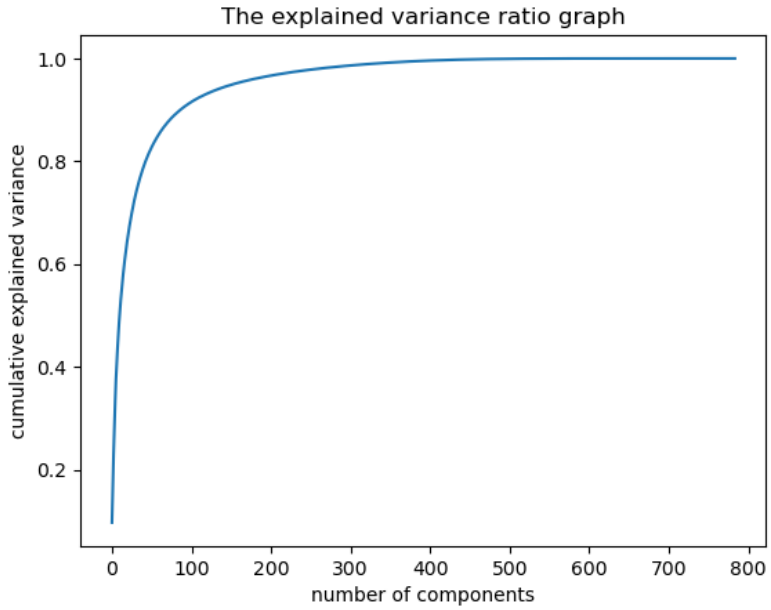


6 principle components:



ב. ציירו גרף של הוריאנס הכולל ש"מוסבר" ע"י n הרכיבים הראשיים.

חישבתי את גרף הוריאנס והצגתי אותו:

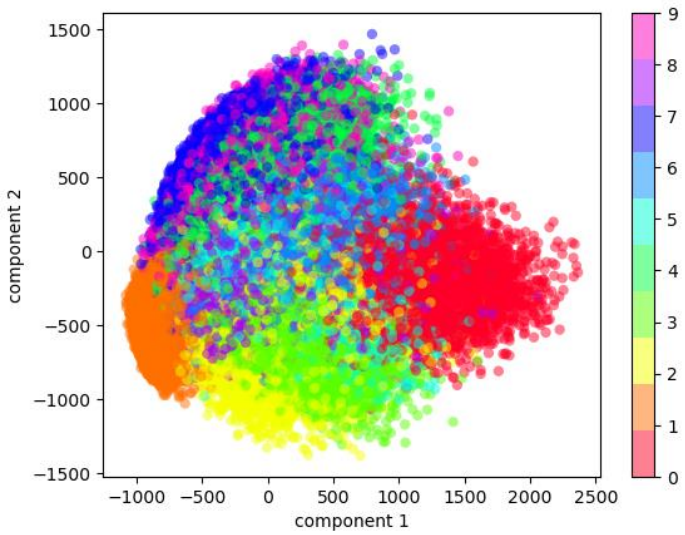


ג. כמה בסיסים צריך בשביל להגיע ל- 95% וריאנס? כמה ל- 80?

צריך 43 בסיסים בשביל להגיע ל- 80% וריאנס, 153 בסיסים להגיע ל- 95% וריאנס

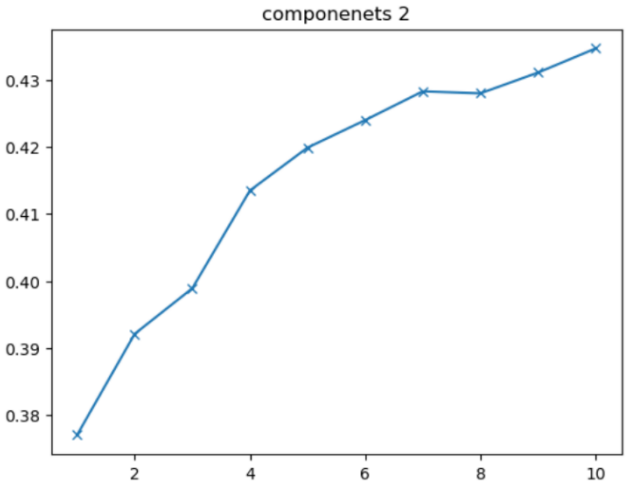
ד. הטילו את הספרות למימד 2 וצייר את הווקטרים המתקבלים (נקודות על המישור , scatter plot) כל סיפרה בצבע אחר.

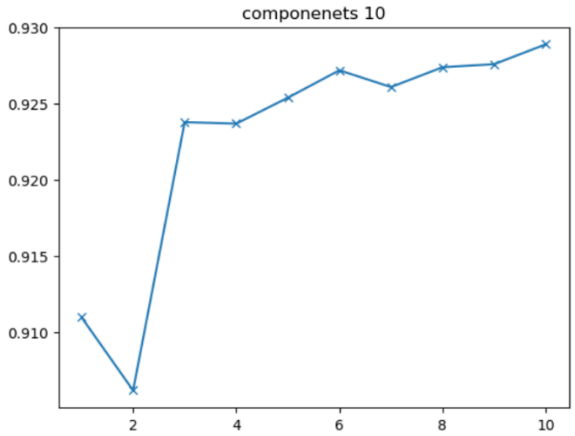
הטלתי את הספרות למימד 2 וציירתי את הווקטורים עבור כל הקלסטרים:

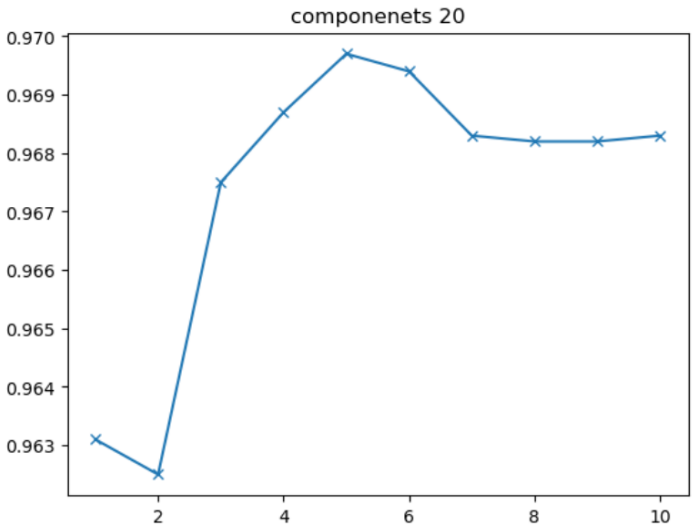


ה. חזרו על שאלה 1 באשר כל סיפרה מיוצגת ע"י ההטלה שלה למימד 2 , 10 ו- 20.

ייצגתי כל סיפרה על ידי ההטלה שלה למימד 2, 10 ו-20:

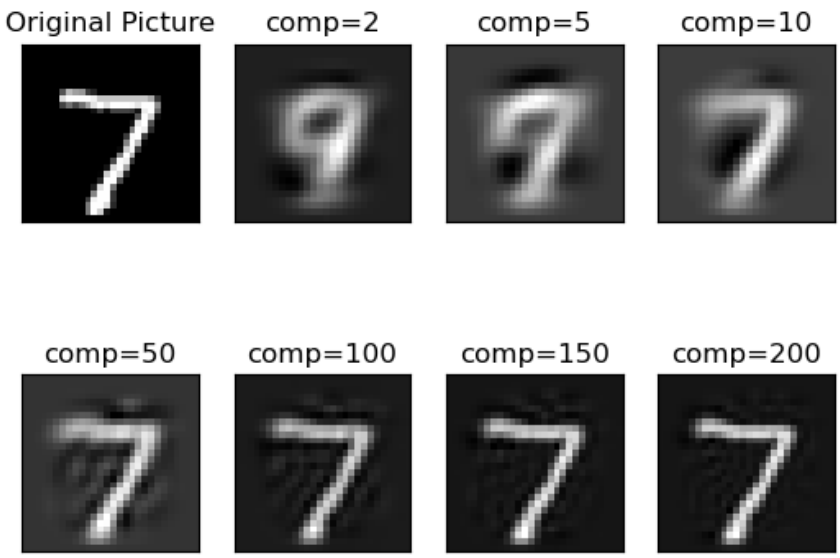






ו. עבוד סיפרה כלשהי, הטילו את הסיפרה למימד k' ושחזרו אותה בחזרה. ציירו את השיחזור עבור k=2,5,10,50,100, 150 כתבו את הנוסחה שבה השתמשתם להטלה ולשיחזור.

ביצעתי המרה ושחזור של תמונה לדוגמא:



הנוסחה שהשתמשתי להטלה:

test_tran = pca.transform(x_test[0].reshape(1, -1))

הנוסחה שהשתמשתי לשיחזור:

inverse_trans = pca.inverse_transform(test_tran).reshape([28,28])

ז. הבעיה העיקרית בשימוש ב PCA -היא הקושי בתיאור של dataset מורכב, כמו במקרה של שלנו. הפתרון הוא לחשב טרנספורמציית PCA לכל מחלקה/סיפרה בנפרד בעזרת סט האימון. ולמדוד את ההתאמה של סיפרה חדשה לכל אחד מהמודלים.

1. חשבו מודל PCA לכל סיפרה בנפרד. הציגו את 6 ה principle components - של כל מודל.

חישבתי מודל PCA לכל סיפרה. הצגה של 6 principle components מצורף בסוף הקובץ.

2. מה ההבדל יחסית למודל של כל הספרות יחד?

ההבדל בין principle components שבמודלים הנפרדים הם נראים בצורה יותר ברורה

3. חשבו את ההטלה של התמונות מה test set -ל כל אחד מהמודלים (סה"כ 10 הטלות לכל תמונה).
4. שחזרו את התמונות מכל אחת מההטלות (סה"כ 10 תמונות משוחזרות מכל תמונה). ציירו את כל השיחזורים של תמונה אחת לדוגמה.
5. חשבו את המרחק בין השיחזורים לתמונה המקורית.
6. המודל ששיחזר "הכי טוב" ייבחר כמודל הנכון.

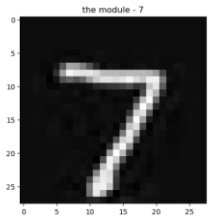
מה הביצועים של מסווג כזה?

חישבתי את ההטלה של התמונות בצורה הבאה: digits_pca[i].transform(x_test[j].reshape(1, -1))

שחזרתי את ההטלות בצורה הבאה:

inverse_trans = digits_pca[i].inverse_transform(test_tran).reshape([28,28])

ציירתי את השיחזור של תמונה אחת לדוגמא:



חישבתי את המרחק בין בין השיחזורים לתמונה המקורית:

np.linalg.norm(x_test[j] - inverse_trans.ravel())

אחוז ההצלחה של המסווג לפי המודלים עבור כל ספרה הוא 0.95% .

חלק 2 – סיווג ע"י BOW:

בחרתי שתי קטגוריות מתוך ה-8: חופים ויערות. חילקתי את סט האימון וסט הבדיקה ביחס של 1:4

חישבתי את מאפייני ה- dense-SIFT מכל התמונות כדי לקבל נקודות עניין בתמונה, כך קיבלתי את סט המאפיינים.

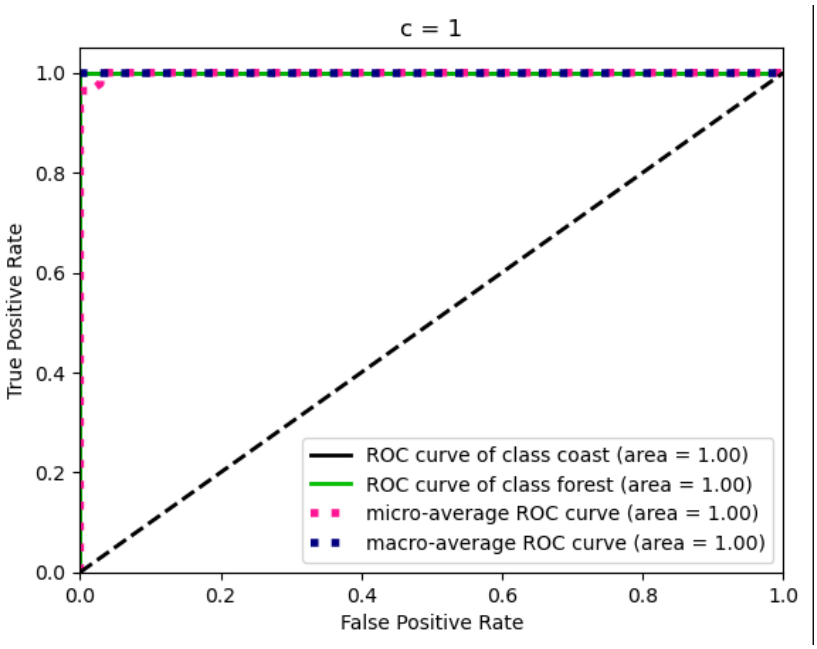
בעזרת אלגוריתם SIFT שנמצא בספרייה Contrib Opencv יצרתי MiniBatchKMeans, בניתי היסטוגרמה למימוש האלגוריתם BagOfWords.

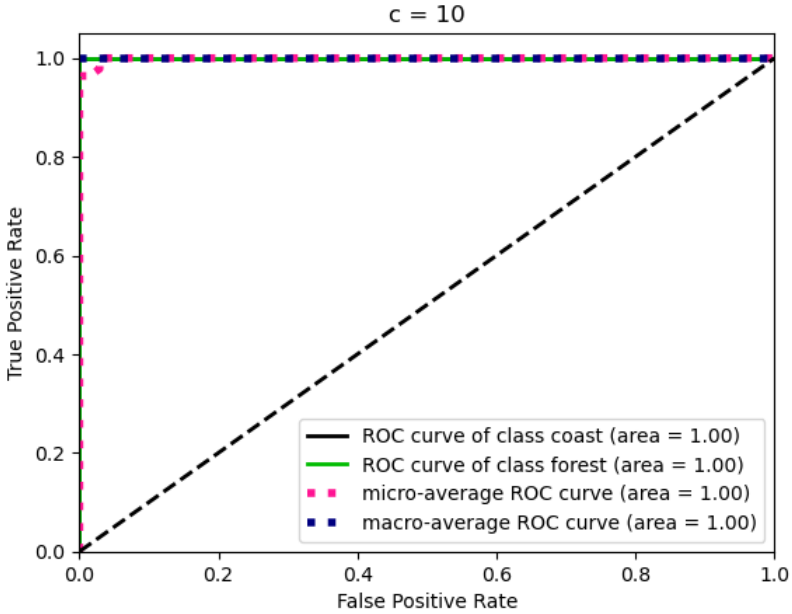
לאחר מכן השתמשתי בפונקציית SVM לחיזוי התיוג של תמונות על סט המבחן.

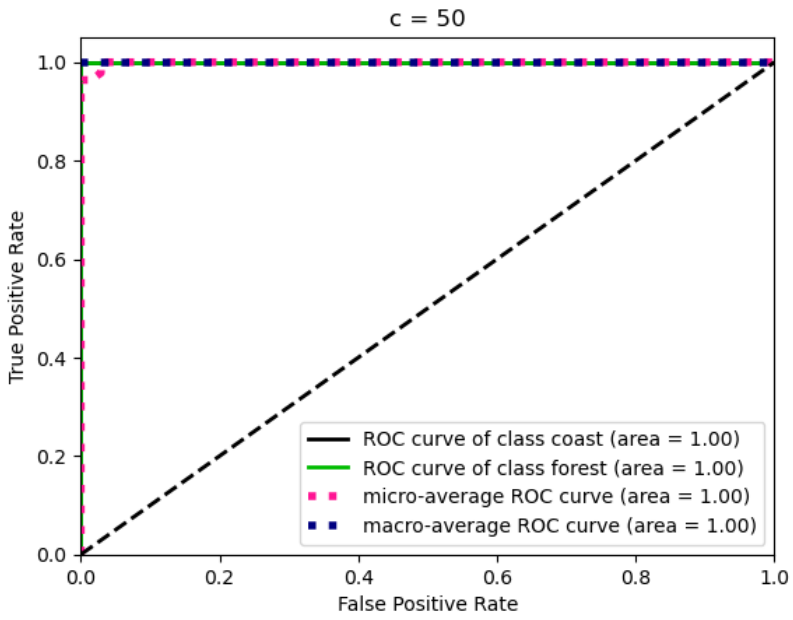
שלב הבדיקה:

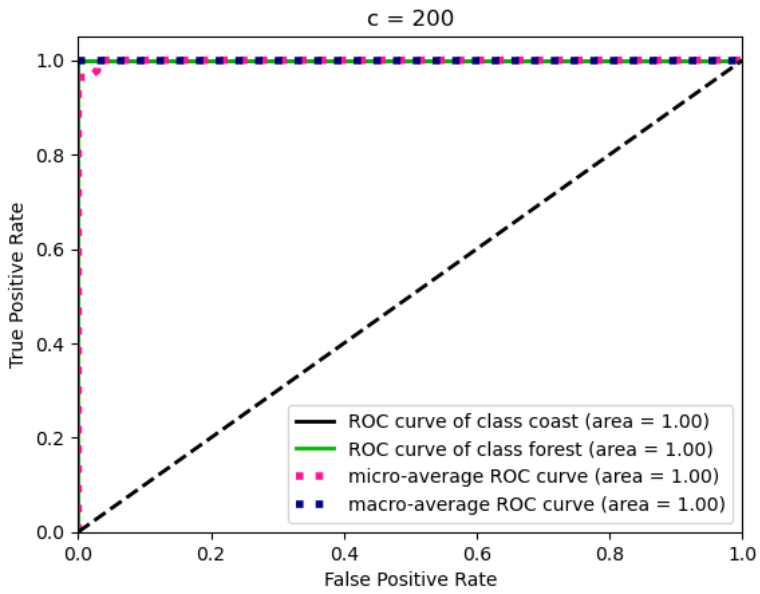
הדפסתי את דיוק המודל באמצעות שימוש ב metrics.classification_report לשם קבלת הדיוק של המודל.
בנוסף ניתן גם להשתמש ב- metrics.confusion_matrix .

השתמשתי ב- skplt.metrics.plot_roc על מנת להציג את הביצועים עבור ערכי מספר ערכים של הקבוע C ע"י עקומות ROC ו AUC. הפרמטרים האופטימליים הם:









קיבלתי דיוק של 99% לכל c.

הביצועים ישתנו עבור חלוקות שונות ל train ו test .

עבור חלוקה של 0.6 אימון ו0.4 מבחן קיבלתי דיוק של 95%

עבור חלוקה של 0.7 אימון ו0.3 מבחן קיבלתי דיוק של 98%

עבור חלוקה של 0.9 אימון ו 0.2 מבחן קיבלתי דיוק של 100%

כלומר התוצאה האופטימלית היא עבור 0.9 אימון ו0.1 מבחן, למרות שעבור 0.1 המבחן פחות אפקטיבי.

נספח לסעיף ז'1:

הצגה של 6 principle components עבור כל מודל: