**שאלה 2**

**סעיף ג'**

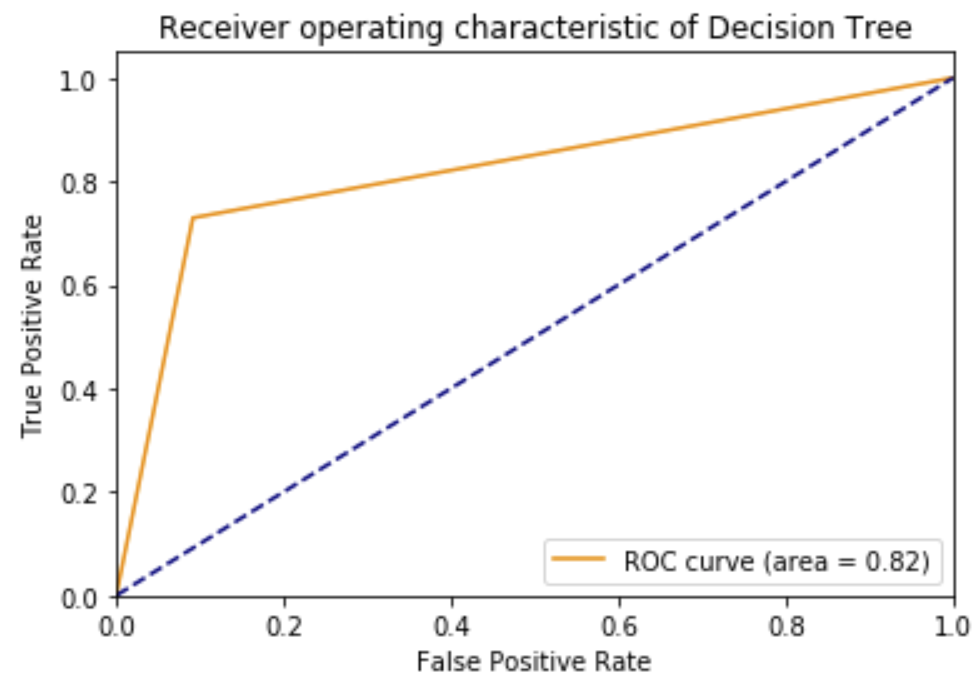
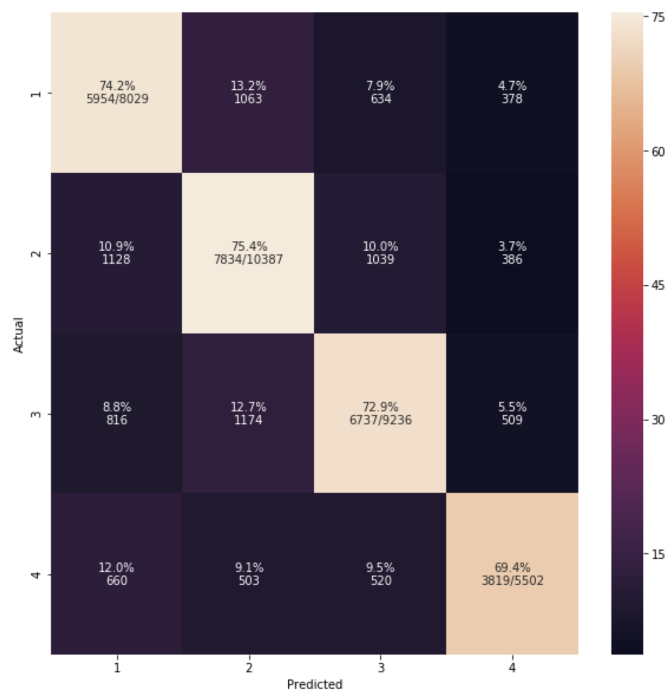
קיבלנו שעץ ההחלטה מראה על דיוק של סביבות 73% ואלגוריתם Naive Bayes מראה על דיוק של סביבות 33%. כלומר עץ ההחלטה בעל דיוק גבוה יותר.

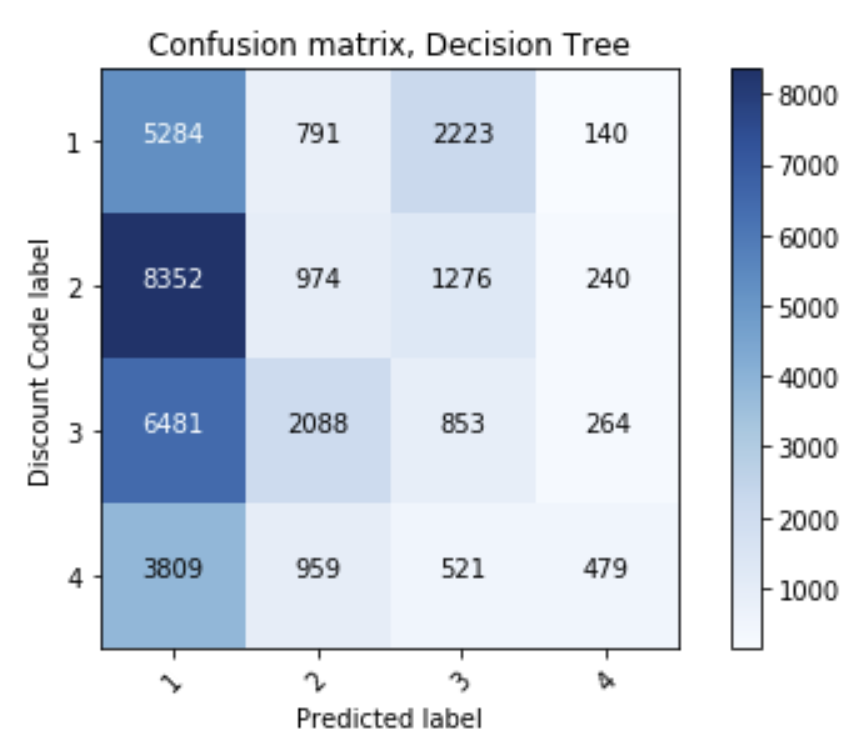
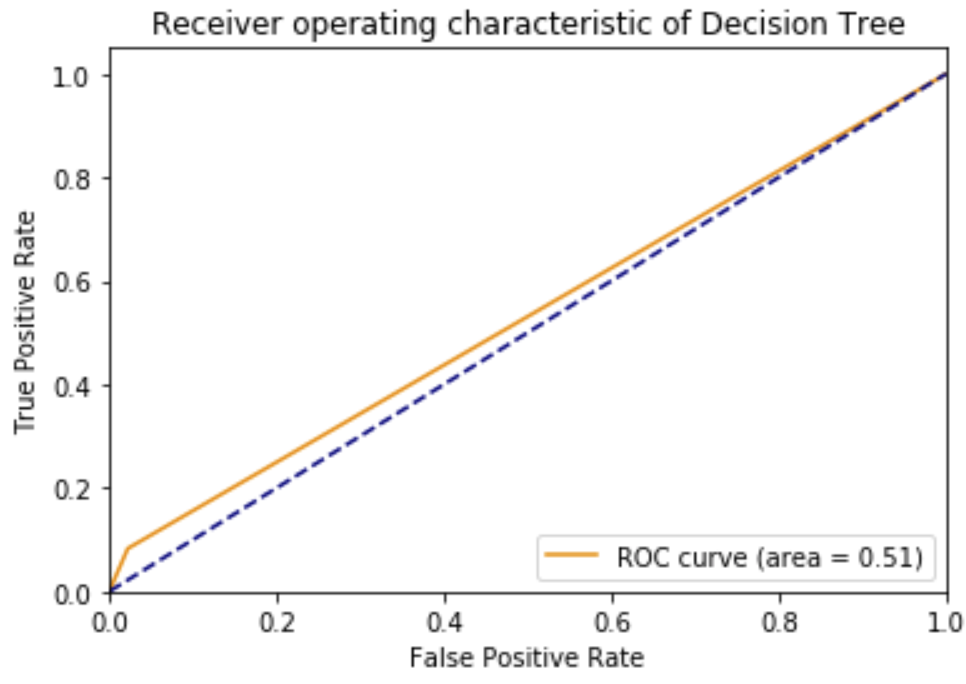
הסיבה לתוצאה הנמוכה היא שאלגוריתם Naive Bayes מסתמך (independence assumption) על כך שה-attributes לא תלויים אחד בשני. אך בנתונים שלנו הם כן תלויים ולכן מספק רמת דיוק נמוכה.

**עצי החלטה**

השתמשנו בעץ החלטה בבעיית הקלסיפיקציה (ניתן להשתמש גם בבעיות רגרסיה) עצי החלטה מחלקים את הפריטים לשניים או יותר סטים הומוגניים. התהליך הזה מבוסס על ה attributes המשמעותיים ביותר/המשתנים הבלתי תלויים (מה שנתון ולא מה שאנו מנסים לחזות) על מנת לבצע הפרדה משמעותית ככל שניתן בין קבוצות. אנחנו רוצים לקחת את הפריטים שלנו ולעשות קלסיפיקציה לכמה קבוצות שונות על בסיס attributes. נחזה את קוד ההנחה שיניב את ההנחה המקסימלית.

Overall Accuracy – חיבור של כל הTP חלקי סך כל הנתונים.

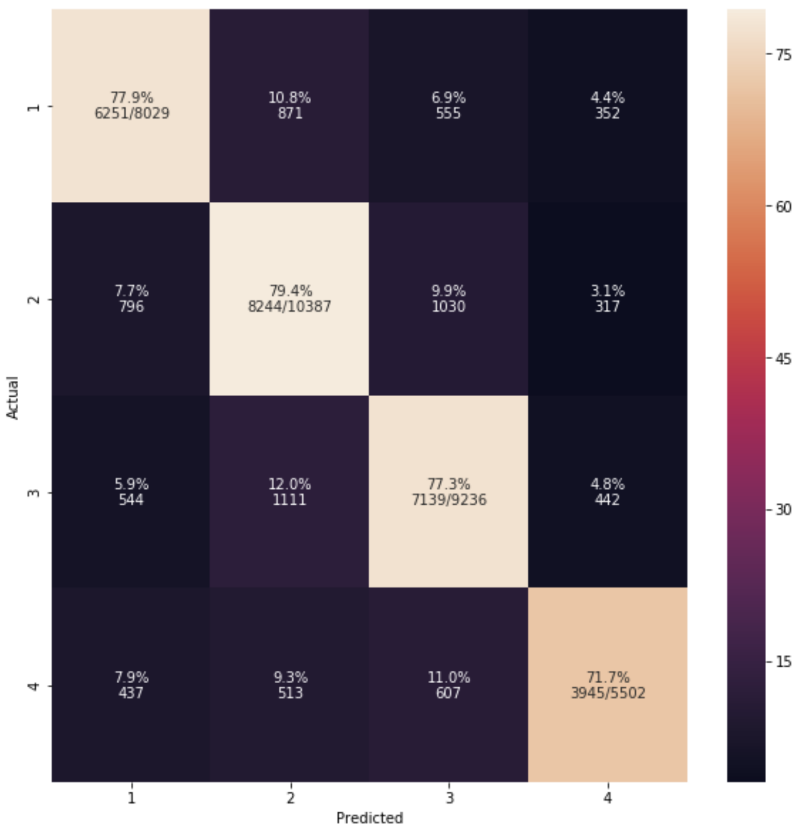
Overall Accuracy= (5954+7834+6737+3819)/(8029+10387+9236+5502)=24344/33154=0.73427

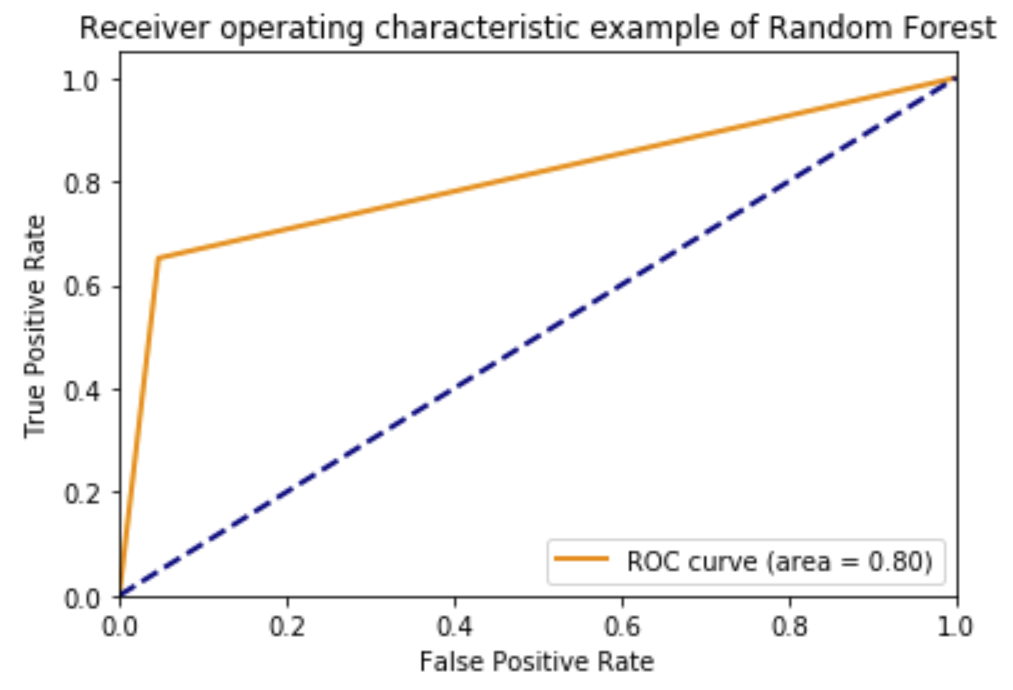
****

**Random Forest Classifier**

זוהי שיטת רגרסיה הבונה מספר רב של עצי החלטה בזמן האימון על מנת לשפר את רמת הדיוק. ואכן קיבלנו שאלגוריתם זה מגיע לרמת דיוק של סביבות 78%, רמת הדיוק הכי גבוהה שקיבלנו.

Overall Accuracy= (6251+8244+7139+3945)/(8029+10387+9236+5502)=25579/33154=0.7715





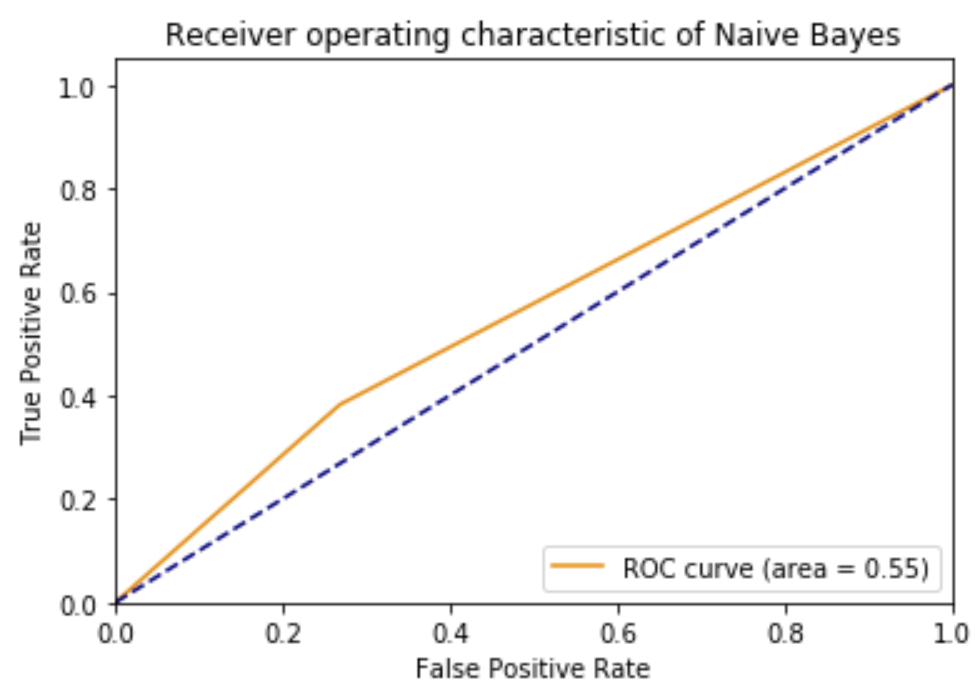
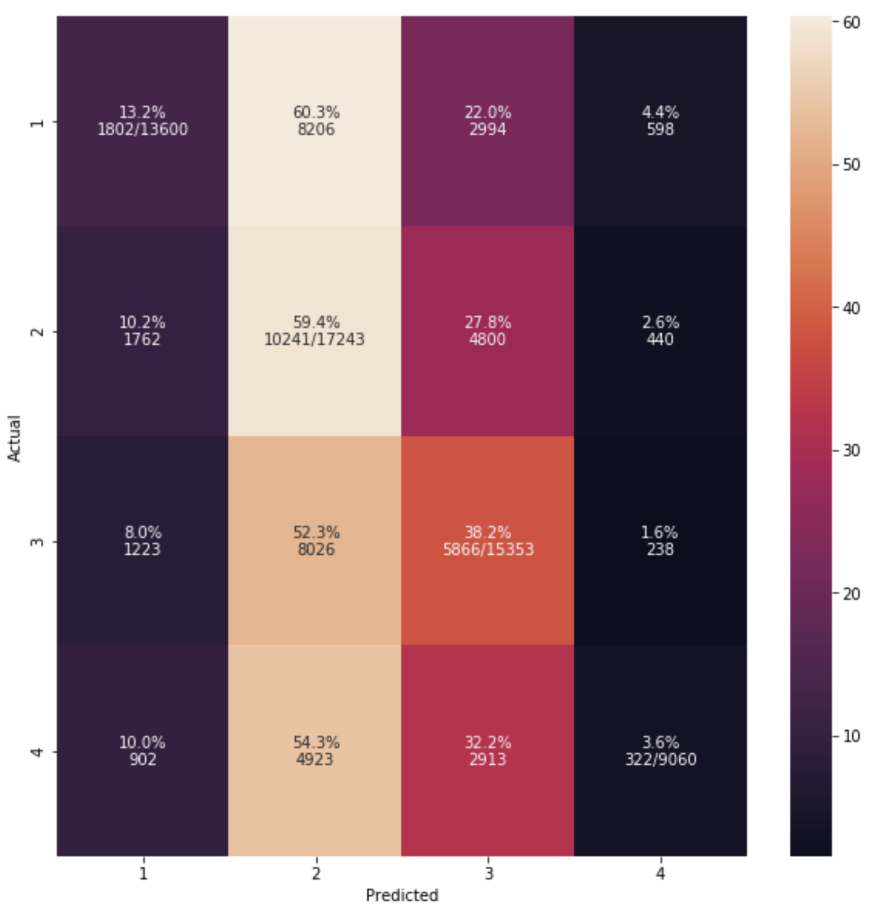
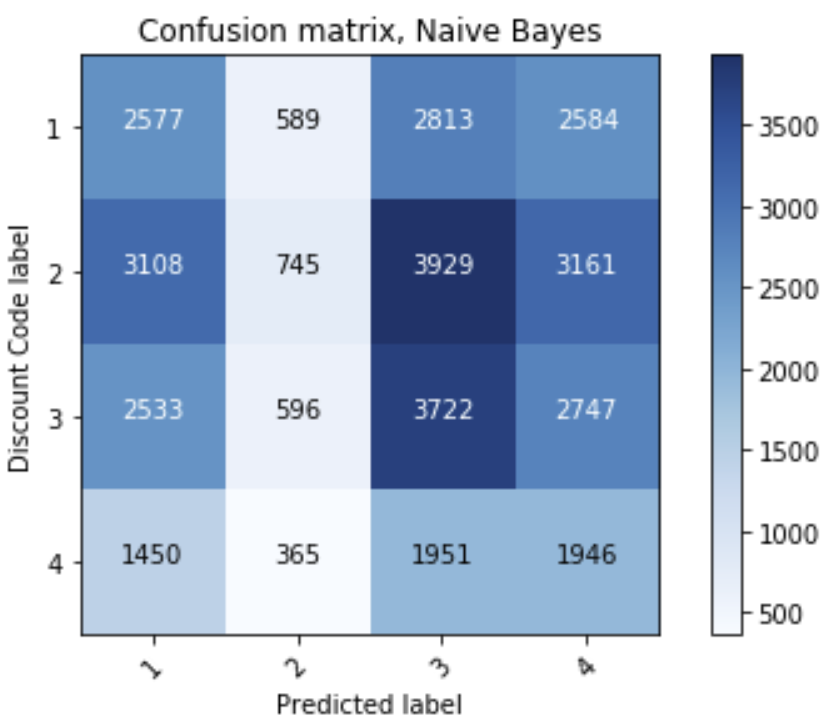
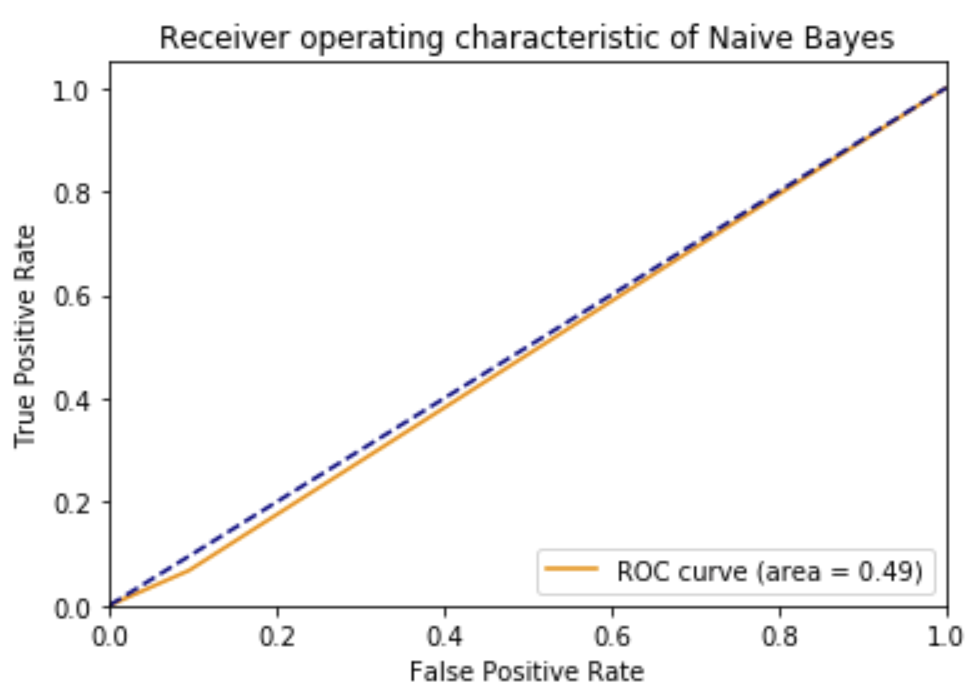
**Naive Bayes**

טכניקת קלסיפיקציה עם הנחה של אי-תלות בין predictors, כלומר feature מסוים נמצא במחלקה ללא

תלות ב feature אחר. האלגוריתם שימושי עבור datasets גדולים.

בסוג קלסיפיקציה זה אנחנו מעוניינים למצוא את ההסתברות ל label מסוים בהינתן features.

Overall Accuracy= (1802+10241+5866+322)/(13600+17243+15353+9060)= 18231/55256=0.33

****

בML בספארק נתקלנו בבעיה בתוצאות החיזוי, קיבלנו לייבלים של 0 עד 3 במקום 1 עד 4.

אחרי שחקרנו בעניין הבנו ש:

all classifiers in ML expect 0 based labels (0.0, 1.0, 2.0, ...) but there is no validations step in ml.NaiveBayes step.

מקור: <https://issues.apache.org/jira/browse/SPARK-9137>

אז עשינו מניפולציה לנתונים כך שכן יהיו מבוססי 0, ורק בשביל החישוב לקחנו את 1:0,2:1,3:2,4:3

בתצוגת הסטטיסטיקה יכולנו להשאיר את הלייבלים המקוריים בגלל ההתאמה הזאת.

**סעיף ד**

על פי בדיקה עם אלגוריתם הInfo Gain הפרמטר שהשפיע הכי הרבה על ההחלטה הוא ה Hotel Name, לאחריו הChecking Date ואז Week Day.

השפעתם של הSnapshot Date ושל ה DayDiff זניחה.

אנו יכולים להניח שיש מלונות שנותנות קוד הנחה מסוים ושגובה ההנחה נקבע לפי המועד (למשל מחיר גבוה יותר בחגים וכו') ולכן הHotel Name וה Checking Date בעלי השפעה חזקה על קוד ההנחה.

לגבי הWeek Day אנו יכולים להניח שגובה ההנחה משתנה לפי היום בשבוע (תחילת שבוע/אמצע שבוע/סוף שבוע).

לשאר הפרמטרים יש פחות קשר להנחה ולכן ההשפעה שלהם על קוד ההנחה קטנה יותר.

**שאלה 3**

**סעיף י"א**

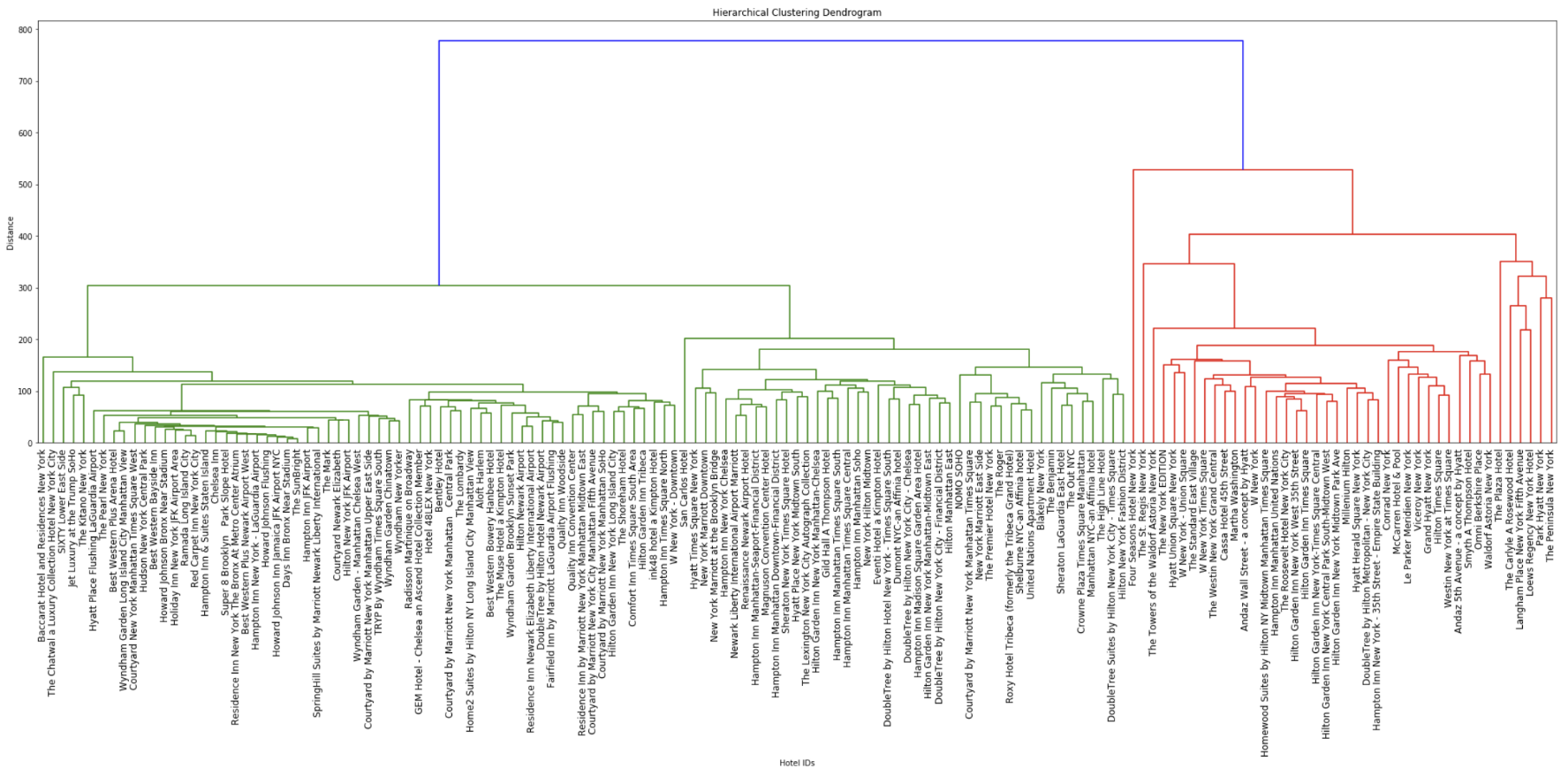
**Dendogram**

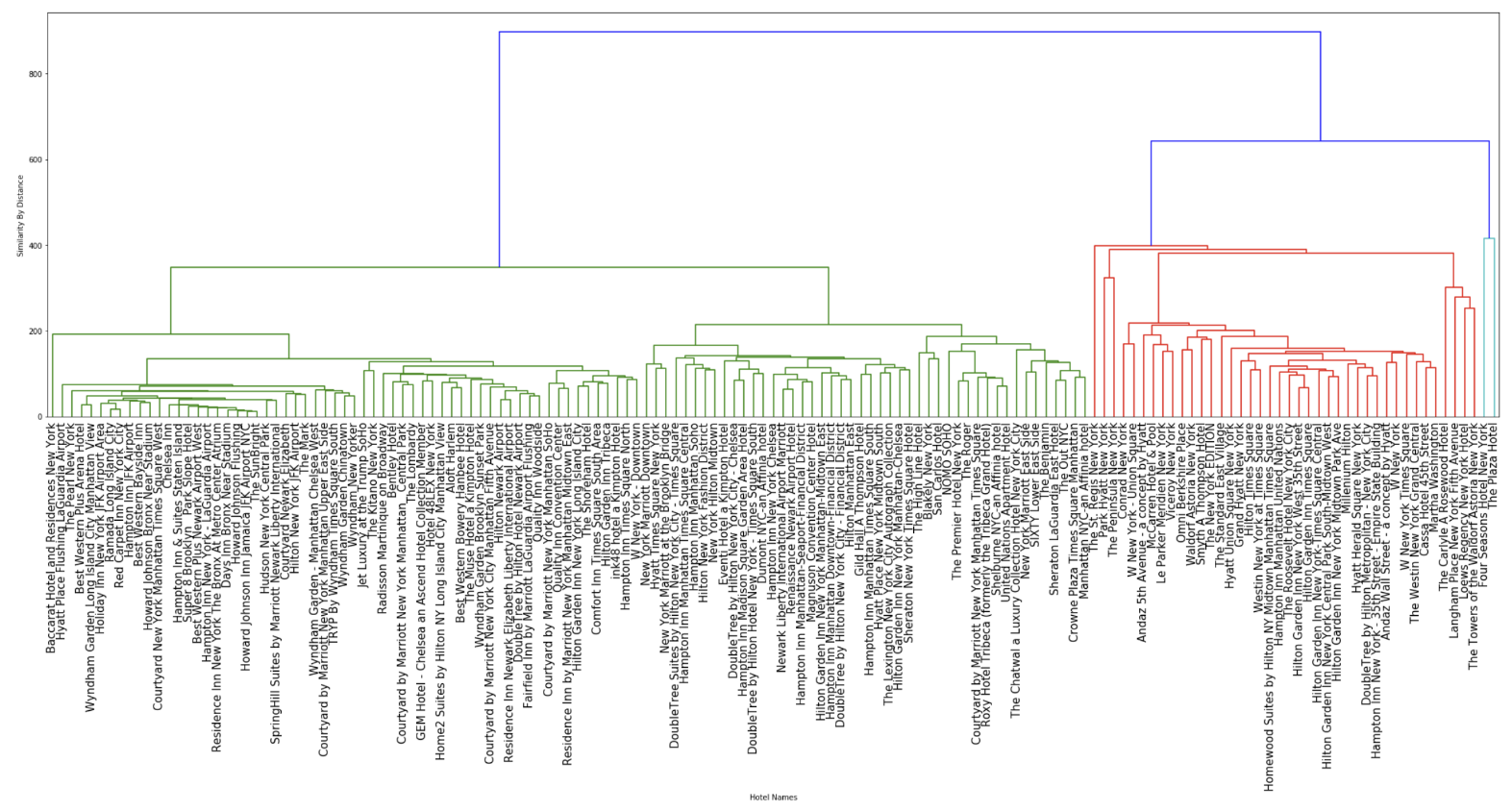
צורה להצגת נתונים כעץ שמציג את הסדר והמרחקים של האיחוד בעת ביצוע ניתוח אשכולות היררכי.

ציר ה- X מציין את שמות המלונות (האינדקסים של הדגימות).

ציר ה- Y מציין את המרחקים.

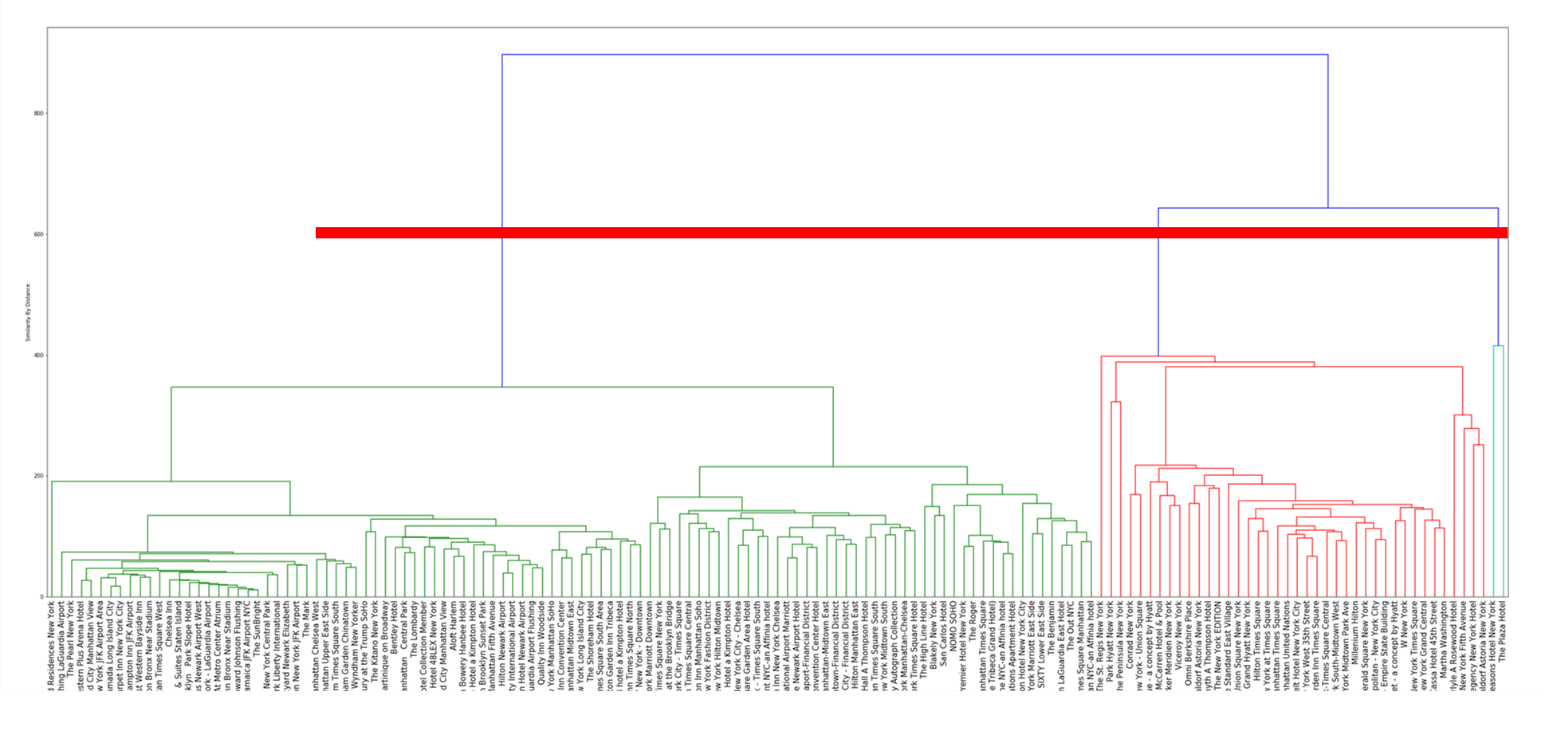
מכל אינדקס יוצא קו אנכי שמתחבר לקו אופקי. הקו האנכי מציין את המרחק שבו התאחדנו לערך אחר או cluster. כל קו אופקי הוא איחוד של clusters.





כדי להבין לכמה קלאסטרים ממליץ הdendogram שיתחלק הdata, מסתכלים על הקו האנכי הכי ארוך ובמרכזו מותחים קו אופקי (threshold)**.**

כמות הקווים שהthreshold נוגע בהם שווה לכמות ה clustersהאופטימלי.אנו רואים ש3 קלאסטרים הם הכמות האופטימלית

****

בשימוש בפונקציה bisecting means הזנו מספרים שונים של קלאסטרים וקיבלנו את הגרף הבא כשציר הX הוא מספרי הקלאסטרים וציר הY הוא הsum of squared roots עשינו זאת כדי למצוא את נקודת הברך וכך נבחין במספר הקלאסטרים המומלץ לחילוק הdata.

הקפיצה הכי גבוהה התבצעה בין 2 ל3. ולאחר מכן ההבדלים קטנים יותר**.** כלומר 3 קלאסטרים. ונתון זה דומה לתוצאה שנתן הdendogram

