



המחלקה למתמטיקה
Department of Mathematics

**פרויקט מסכם לתואר בוגר במדעים (B.Sc)
במתמטיקה שימושית**

רגרסיה לוגיסטית: תיאוריה ושימושים

סבטלנה קוסטיוקוביץ'

Logistic Regression: Theory and Applications

Svetlana Kostukovich

**פרויקט מסכם לתואר בוגר במדעים (B.Sc)
במתמטיקה שימושית**

רגרסיה לוגיסטית: תיאוריה ושימושים

סבטלנה קוסטיוקוביץ'

Logistic Regression: Theory and Applications

Svetlana Kostukovich

Advisor:

Assoc. Prof.
Haggai Katriel

מנחה:

פרופ"ח חגי כתריאל

Karmiel

2018

כרמיאל



תוכן עניינים

1	מבוא	1
3	הצגת המודל הלוגיסטי	2
3	2.1 הקדמה למודל	
3	2.2 פונקציית הלוגיט	
6	2.3 הפונקציה הלוגיסטית	
8	2.4 המודל	
9	2.5 דוגמה לשימוש במודל לוגיסטי	
13	3 שיטת הנראות המקסימלית	3
13	3.1 פונקציית הנראות	
15	3.2 אומדן הנראות המקסימלי	
18	3.3 שיטת הנראות המקסימלית ברגרסיה לוגיסטית	
21	3.4 קעירות פונקציית לוג הנראות	
27	4 בחינת מידת הצלחת המודל	4
30	5 הצגת המודל "עיכובי טיסות"	5
30	5.1 רקע	
32	5.2 ניתוח ראשוני של הנתונים	
36	5.3 שלבי בניית המודל	
36	5.3.1 משתנים קטגוריים למשתני דמה	
37	5.3.2 חלוקה לקבוצת אימון וקבוצת בדיקה	
37	5.3.3 התאמת המודל	
38	5.3.4 בחינת המודל	
41	5.4 מסקנה	
42	6 סיכום	6
43	7 ביבליוגרפיה	7
44	8 נספח	8

1 מבוא

רגרסיה לוגיסטית הינו מודל סטטיסטי העוסק בתאור היחסים בין ערכים הנמדדים לתוצאה הנצפית בסדרת תצפיות בה יחסים אלו אינם ידועים מראש. להבדיל ממודלי רגרסיה אחרים, ברגרסיה הלוגיסטית אנו מקבלים אומדן להסתברות לתוצאה בהינתן הערכים שנמדדו, כלומר הערך שיניב המודל יהיה תמיד בין 0 ל-1 (לעומת רגרסיה לינארית לדוגמה, שמניבה מספר ממשי).

הרציונל לקבלת ערך המייצג הסתברות מצוי באופי השאלות שעבורן הרגרסיה הלוגיסטית מתאימה, לרוב משתמשים במודל על מנת לקבל תשובה לשאלה מסוג "האם" שעבורה קיימות מספר אפשרויות בודדות, ומאחר ואין ודאות לגבי התשובה, נידרש לחזות את ההסתברות לקבל תוצאה מסוימת. לפעמים אף מוסיפים ובוחרים רף הסתברות מסוים בתור ערך סף, כך שבמידה ומתקבלת מתהליך הרגרסיה הלוגיסטית הסתברות שהיא מעל לאותו סף, החוקר יבחר לחזות כי תוצאה מסוימת תתקיים, בעוד שבמקרה ההפוך יבחר לחזות שהתוצאה לא תתקיים.

מודל זה פותח על ידי הסטטיסטיקאי דיוויד קוקס בשנת 1958. תחילה המודל לא היה נפוץ בשימוש במחקר ובמדע אך עם התפתחות המיחשוב הפך לאחד המודלים הנפוצים ביותר [5].

כיום, מודל הרגרסיה הלוגיסטית שימושי בתחומים שונים, ביניהם: למידה חישובית, רפואה ומדעי החברה. ניתן להשתמש במודל כדי לאמוד את הסיכוי של חולה לחלות במחלה מסוימת, בהתבסס על מאפיינים נצפים של המטופל (גיל, מין, מסת הגוף, תוצאות בדיקות דם שונות וכו'), ביישומים שיווקיים לחיזוי נטייה של לקוח לרכוש מוצר או להפסיק מינוי, בפוליטיקה לחיזוי הצבעת בוחר אמריקאי לדמוקרטים או לרפובליקנים, בהנדסה, לאומדן הסתברות לכשל של תהליך, מערכת או מוצר, ובעוד אינספור דוגמאות בהן המודל שימושי להפליא.

במסגרת עבודה זו אתמקד ברגרסיה לוגיסטית בינארית העוסקת במצבים שבהם תוצאת תצפית היא אחת משתי אפשרויות, לדוגמה - כישלון או הצלחה, יש או אין, חולה או בריא.

בפרק "הצגת המודל הלוגיסטי" אציג את שתי הפונקציות החשובות ברגרסיה הלוגיסטית - פונקציית הלוגיט והפונקציה הלוגיסטית. בנוסף אסביר על המודל ואתן דוגמה לשימוש בו.

בפרק "שיטת הנראות המקסימלית" אציג את שיטת הנראות המקסימלית (פונקציית הנראות ואומדן הנראות המקסימלי) המשמשת למציאת אמד לפרמטרים המאפיינים את המודל ואראה את הקשר של השיטה לרגרסיה הלוגיסטית. בנוסף אוכיח שפונקציית לוג הנראות עבור המודל הלוגיסטי הינה פונקציה קעורה, תוצאה זו תבטיח שהאמד לפרמטרים שנקבל הוא האמד האופטימלי למודל.

בפרק "בחינת מידת הצלחת המודל" אסקור את אחת מהשיטות הקיימות לבחינת הצלחת המודל- מדד המיומנות של ברייר.
בפרק "מודל עיכובי טיסות" אדגים איך הרגרסיה הלוגיסטית באה לידי ביטוי בלמידה חישובית בבעיה שבה ננסה לחזות עיכובי טיסות.

אשתדל להציג את הנושא בצורה כזו שכל בוגר מדעים או הנדסה עם רקע מתמטי שנלמד בתואר הראשון, יוכל להבין את נושא הרגרסיה הלוגיסטית, יתרונותיה, מגבלותיה ואיך להשתמש בה לצרכיו.

2 הצגת המודל הלוגיסטי

2.1 הקדמה למודל

במהלך הצגת המודל ישנם מושגים שיחזרו על עצמם שוב ושוב, ועל מנת למנוע בילבול אציג אותם כעת.

משתנה תלוי

משתנה זה מוכר גם כמשתנה התגובה, המטרה או המשתנה המוסבר, ערכו מושפע מהמשתנים הבלתי תלויים. נסמנו ב-1 במקרה שהתכונה הנחקרת מתקיימת ו-0 במידה ולא או בסימון פורמלי $y \in \{0, 1\}$.

משתנה בלתי תלוי

מוכר גם כמשתנה המנבא או המסביר, זהו משתנה שנבחר במטרה לבדוק את השפעתו על המשתנה התלוי. משתנה זה יכול לקבל ערך קטגורי (ערך המייצג שייכות לקבוצה מסויימת מתוך אוסף קבוצות ביניהן לא קיים יחס סדר, לדוגמה - מין ודת) או נומרי (לדוגמה - משקל). נסמן משתנה זה ב- X_i כאשר i מסמל תכונה מסויימת.

מטרת המודל הלוגיסטי היא להעריך מה ההסתברות שהמשתנה התלוי יהיה שווה ל-1 בהינתן ערכם של המשתנים הבלתי תלויים. במילים אחרות מה ההסתברות שתכונה מסויימת מתקיימת כתלות בתכונות אחרות. בלמידה חישובית ניתן לממש זאת בכך שנלמד את המודל בעזרת קבוצת נתונים שבה מצויין הערך של y עבור כל תצפית.

2.2 פונקציית הלוגיט

בבעיות רגרסיה הכמות המעניינת, אותה אנו מנסים למצוא, היא הערך הצפוי (התוחלת) של המשתנה התלוי בהינתן המשתנים הבלתי תלויים - $E(Y|X)$ [2].

ברגרסיה לוגיסטית המשתנה התלוי הוא משתנה בינארי בעל התפלגות מותנית ברנולי משום ש y יכול לקבל רק 2 ערכים, 1 ו-0. ההסתברות ל- $y = 1$ (הצלחה בניסוי) היא $P(y = 1)$ וההסתברות ל- $y = 0$ (כישלון בניסוי) היא $1 - P(y = 1)$ כתלות בקלט (משתנים בלתי תלויים) $(X_1, X_2, \dots, X_n)$.

ידוע כי בהתפלגות ברנולי התוחלת שווה להסתברות לכך ש- $y = 1$. לפיכך במקרה

שלנו נקבל:

$$(2.1) \quad E(Y|X) = P(Y = 1|X = X_1, X_2, \dots, X_n)$$

למעשה המודל הלוגיסטי נותן לנו קירוב ל- (2.1), נסמן קירוב זה ב- $h_\theta(X)$.

ניצור צירוף לינארי של משתני הקלט של המודל- $\theta_0 X_0 + \theta_1 X_1 + \dots + \theta_n X_n$, כאשר θ_i משמש כמשקל למשתנה X_i (נהוג להוסיף משתנה X_0 שערכו תמיד 1). נרצה לקשר בין צירוף לינארי זה כאשר ערכו יכול להיות מ- $-\infty$ עד ∞ , לתוחלת המותנית שערכה יכול להיות מ-0 עד 1. לצורך כך נשתמש בפונקציה המקשרת את הצירוף הלינארי ל- $E(Y|X)$, אותה נסמן ב- g ותכף נציגה.

נגדיר מושג חדש - הסיכוי, סיכוי של מאורע הוא ההסתברות שהמאורע יתרחש חלקי ההסתברות שאותו מאורע לא יתרחש.

$$odds = \frac{p}{1-p}$$

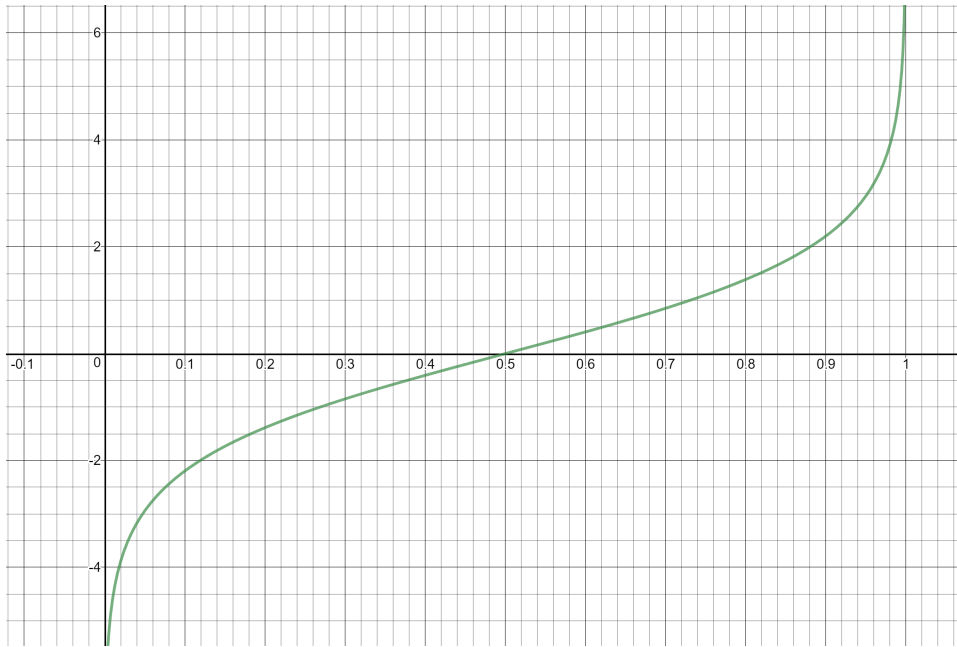
באופן אינטואיטיבי ניתן לומר שבמודל לוגיסטי הנחת הבסיס היא ששינוי ביחידה אחת של הקלט מתבטא בהכפלת הסיכויים של המאורע בקבוע. באופן פורמלי יותר, מודל לוגיסטי הוא כזה שפונקציית לוג הסיכויים של הפלט היא לינארית ביחס לקלט. כלומר:

$$(2.2) \quad g(h_\theta(x)) = \ln \left(\frac{h_\theta(X)}{1 - h_\theta(X)} \right) = \theta_0 + \theta_1 X_1 + \dots + \theta_n X_n$$

כאשר,

$$g(p) = \ln \left(\frac{p}{1-p} \right)$$

הינה פונקציית הלוגיט והיא הפונקציה המקשרת.



איור 1: גרף פונקציית הלוגיט

נשים לב כי פונקציית הלוגיט אכן מקבלת ערכים בין 0 ל-1 ופולטת ערכים בין $-\infty$ ל- ∞ .

נחלץ את $h_\theta(X)$ מהביטוי (2.2):

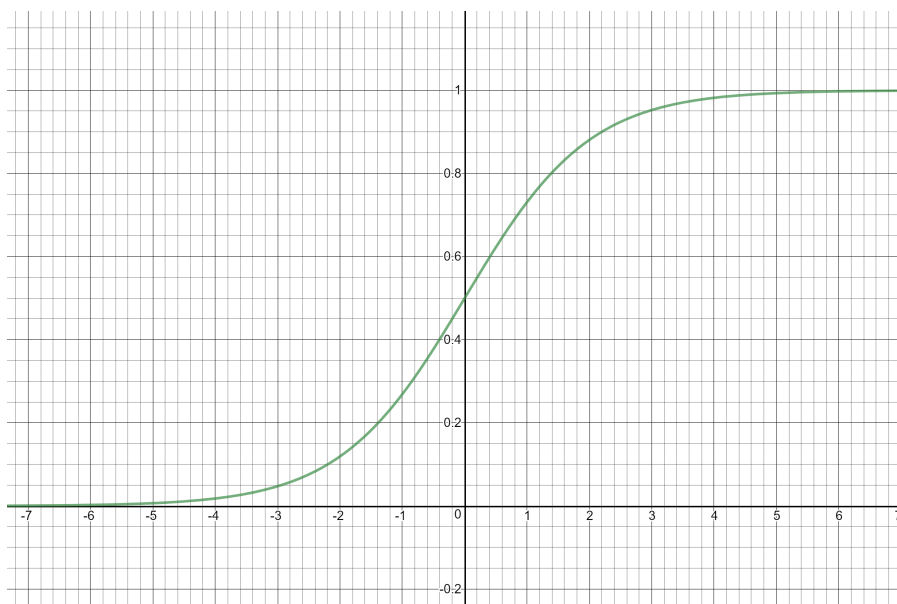
$$\begin{aligned}
 e^{\ln\left(\frac{h_\theta(X)}{1-h_\theta(X)}\right)} &= e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)} \\
 \Rightarrow \frac{h_\theta(X)}{1-h_\theta(X)} &= e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)} \\
 \Rightarrow h_\theta(X) &= (1-h_\theta(X))e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)} \\
 \Rightarrow h_\theta(X) + h_\theta(X)e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)} &= e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)} \\
 \Rightarrow h_\theta(X)(1 + e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)}) &= e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)} \\
 \Rightarrow h_\theta(X) &= \frac{e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)}}{1 + e^{(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)}} = \frac{1}{1 + e^{-(\theta_0 + \theta_1 X_1 + \dots + \theta_n X_n)}}
 \end{aligned}$$

קבלנו את $h_\theta(X)$, הפונקציה ההפוכה לפונקציית הלוגיט הידועה גם בשם הפונקציה הלוגיסטית.

2.3 הפונקציה הלוגיסטית

בסעיף הקודם הכרנו את הפונקציה הלוגיסטית עליה מבוססת הרגרסיה הלוגיסטית.

$$h(z) = \frac{1}{1 + e^{-z}}$$



איור 2: גרף הפונקציה הלוגיסטית

הפונקציה מקיימת את התכונות הבאות:

1. גבולות הפונקציה

כאשר z שואף לאינסוף הפונקציה שואפת ל-1 וכאשר z שואף למינוס אינסוף הפונקציה שואפת ל-0.

$$\lim_{z \rightarrow -\infty} h(z) = \lim_{z \rightarrow -\infty} \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-(-\infty)}} = \frac{1}{1 + e^\infty} = 0$$

$$\lim_{z \rightarrow \infty} h(z) = \lim_{z \rightarrow \infty} \frac{1}{1 + e^{-z}} = \frac{1}{1 + e^{-\infty}} = 1$$

2. פונקציה מונוטונית עולה

$h(z)$ היא פונקציה גזירה משום שהיא הרכבה של פונקציות אלמנטריות. בנוסף נגזרתה חיובית לכל z .

$$h'(z) = \frac{e^{-z}}{(e^{-z} + 1)^2} > 0$$

$e^{-z} > 0$ וגם $(e^{-z} + 1)^2 > 0$ ולכן גם המנה חיובית.

נזכיר משפט על מונוטוניות של פונקציה:

אם $f(x)$ גזירה בקטע פתוח (a, b) ומתקיים $f'(x) > 0$ לכל $x \in (a, b)$, אזי $f(x)$ מונוטונית עולה ממש בקטע (a, b) .

$h(z)$ מקיימת את תנאי המשפט ולכן מונוטונית עולה ממש.

3. חסומה בין 0 ל-1.

משלושת תכונות אלו מתקבל כי ערכי הפונקציה נעים בין 0 ל-1 כתלות ב z . זאת אחת הסיבות לשימוש בפונקציה זו במודל הרגרסיה הלוגיסטית, משום שגם פונקציית הסתברות מקבלת ערכים בטווח זה ולכן הפונקציה הלוגיסטית שימושית לתאור הסתברות.

תכונה שימושית נוספת היא שהשפעת ערכי z הנמוכים היא מינימלית עד לסף מסוים ($z = threshold$) ועבור ערכים גדולים מסף זה הפונקציה עולה במהירות לסביבת 1.

קיימת מודלים נוספים שמשתמשים בפונקציות אחרות בעלות אופי דומה (לדוגמה *Probit model*), אם כי המודל הלוגיסטי הוא הנפוץ ביותר ביניהם.

2.4 המודל

כדי לקבל את המודל נציב בפונקציה הלוגיסטית:

$$z = \theta_0 X_0 + \theta_1 X_1 + \theta_2 X_2 + \dots + \theta_n X_n$$

כאשר:

X_i - ערך המשתנה i (משתנה בלתי תלוי) עבור תצפית מסויימת. כאשר $0 \leq i \leq n$.
עבור $X_0 = 1, i = 0$.

נסמן את קבוצת ערכי משתנים אלו ב- $X = (X_0, X_1, \dots, X_n)$

n - מספר המשתנים הבלתי תלויים.

θ_i - פרמטרים לא ידועים שעלינו לאמוד על סמך נתונים המתקבלים מ- X ומ- y (וקטור המטרה) $0 \leq i \leq n$. קיימים $n + 1$ פרמטרים.

נרצה לאמוד את הפרמטרים הלא ידועים על סמך מדגם תצפיות עבור אוכלוסיה מסויימת, כדי שנוכל להציב אותם בפונקציה הלוגיסטית ולקבל חיזוי ל- y עבור תצפית חדשה. נשתמש בשיטת הנראות המקסימלית כדי לאמוד את הפרמטרים, עליה אפרט בהמשך.

לאחר ההצבה נקבל פונקציה מהצורה:

$$h_\theta(X) = \frac{1}{1 + e^{-\sum_0^n \theta_i X_i}}$$

הפונקציה $h_\theta(X)$ נקראת ההיפוטזה והיא נותנת לנו את תחזית המודל לגבי ההסתברות שמשתנה המטרה יהיה שווה ל- 1 בהינתן המשתנים הבלתי תלויים.

$$h_\theta(X) = P(y = 1|X) = 1 - P(y = 0|X)$$

לדוגמה, אם קבלנו ש $h_\theta(X) = 0.7$ זאת אומרת, על פי המודל, שבהסתברות של 70 % y יהיה שווה ל- 1.

2.5 דוגמה לשימוש במודל לוגיסטי

נסתכל על דוגמה בה נרצה לחזות אם גידול הוא סרטני או לא אצל אדם כלשהו. במקרה זה משתנה המטרה מסמן 1 אם הגידול הוא סרטני ו-0 אם לא. התכונות שעליהן נרצה להסתכל (משתנים בלתי תלויים) הן:

$$X_1 = \text{tumor size(cm)}, X_2 = \text{age}, X_3 = \text{gender}$$

"גודל הגידול" ו"גיל" הם משתנים רציפים, "מין" משתנה קטגורי.

כאשר משתנה מקבל ערך קטגורי כגון מין, דת, צבע וכו' לא ניתן להשתמש בהם במודל משום שלערכים אלו אין משמעות נומרית. במקרה זה נהפוך את המשתנים הקטגוריים למשתני דמה. לדוגמה במקרה שלנו אחד המשתנים הוא מין. עבור משתנה זה אנו נצטרך משתנה דמה כאשר ערך משתנה זה יהיה שווה ל-1 עבור אישה ו-0 עבור גבר. באופן כללי אם למשתנה קטגורי ישנם k אפשרויות אז נזדקק ל $k - 1$ משתני דמה.

נציב את המשתנים במודל ונקבל:

$$h_{\theta}(X) = \frac{1}{1 + e^{-(\theta_0 + \theta_1 \cdot X_1 + \theta_2 \cdot X_2 + \theta_3 \cdot X_3)}}$$

יש לנו נתונים על 700 אנשים עם גידולים כך שעל כל אחד מהם ידוע לנו אם גידולו היה סרטני או לא.

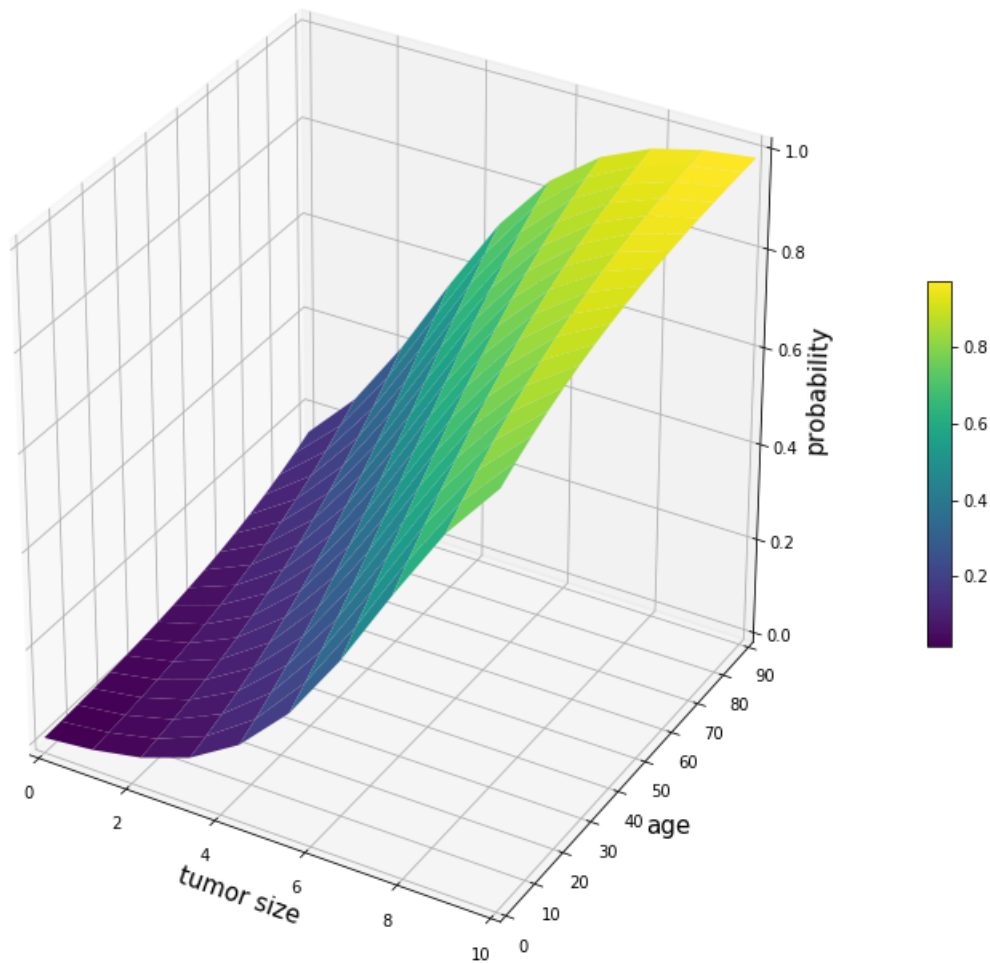
נניח בשלב זה כי הצלחנו לאמוד את הפרמטרים הלא ידועים בעזרת מדגם זה (נסביר איך עושים זאת בפרק "שיטת הנראות המקסימלית"), וקבלנו ערכים עבור θ (לא אמיתיים, להמחשה בלבד):

$$\theta_0 = -4.5, \theta_1 = 0.6, \theta_2 = 0.03, \theta_3 = 0.02$$

נציב ערכים אלו ב- $h_{\theta}(X)$ ונקבל את פונקציית ההסתברות המשוערת.

$$h_{\theta}(X) = \frac{1}{1 + e^{-(4.5 + 0.6X_1 + 0.03X_2 + 0.02X_3)}}$$

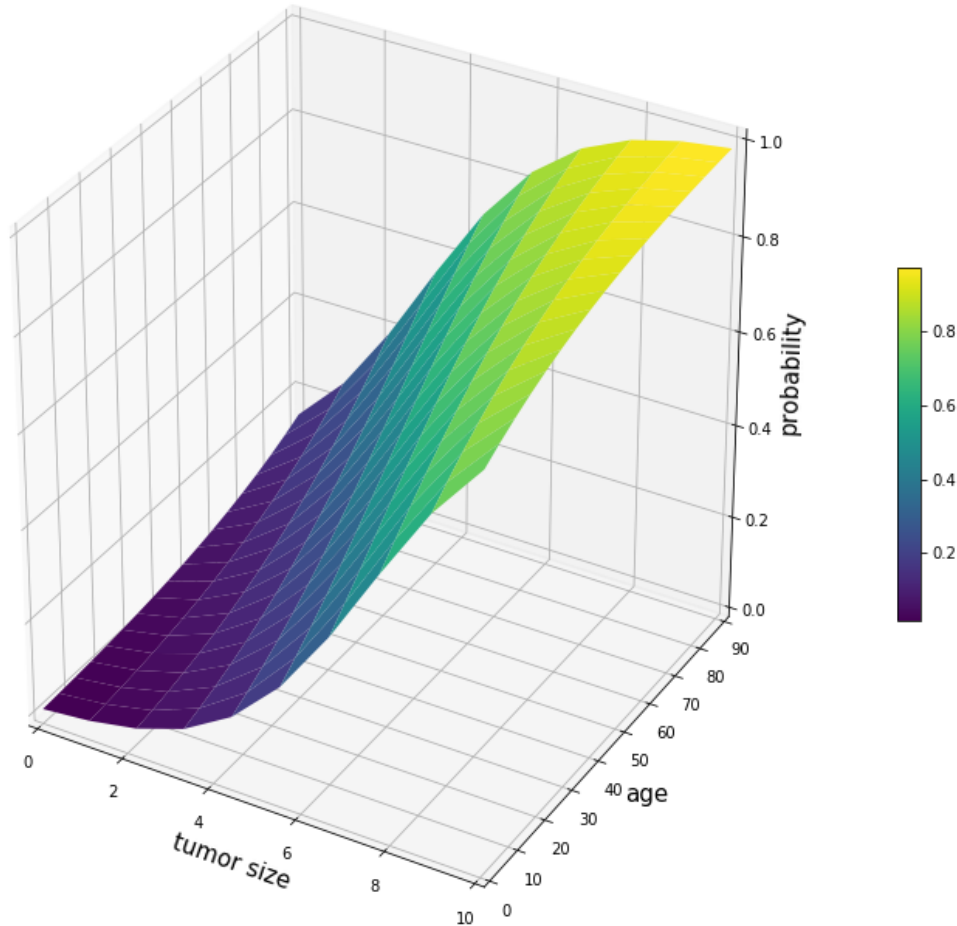
נציג גרף תלת מימדי של ההסתברות המשוערת לגידול סרטני $h_\theta(X)$ עבור אוכלוסיית הנשים ($X_3 = 1$) כתלות בגודל הגידול ובגיל:



איור 3: גרף ההסתברות המשוערת של אוכלוסיית הנשים

ניתן לראות מהגרף שככל שהגודל של הגידול גדל והגיל עולה כך גם ההסתברות המשוערת לגידול סרטני גדלה. בנוסף נשים לב שכאשר גודל הגידול כ- 10 cm ההסתברות המשוערת היא כמעט 100 אחוז שהגידול סרטני ללא תלות בגיל.

נדגים גרף תלת מימדי של ההסתברות המשוערת לגידול סרטני $h_\theta(X)$ עבור אוכלוסיית הגברים ($X_3 = 0$) כתלות בגודל הגידול ובגיל:



איור 4: גרף ההסתברות המשוערת של אוכלוסיית הגברים

נשים לב ששני הגרפים כמעט זהים, לפיכך ניתן לחשוד שהמשתנה "מין" (X_3) לא משמעותי לחיזוי ההסתברות המשוערת, ואולי אף נרצה להימנע משימוש בו כדי לפשט את המודל. כדי לקבל החלטה כזו, נצטרך לערוך מבחנים נוספים אשר יכריעו אם משתנה זה משמעותי לחיזוי או לא.

על מנת להשתמש במודל זה כדי לבדוק את ההסתברות שגידול של אדם מסויים הוא סרטני (תצפית חדשה), נבדוק מה הם ערכי המשתנים הבלתי תלויים של האדם (גודל הגידול, גיל ומין) ולאחר מכן נציב את הערכים ב- $h_\theta(X)$ כדי לקבל את ההסתברות המשוערת.

לדוגמה, עבור אדם עם ערכים אלו:

$$\text{tumor size} = 4, \text{ age} = 40, \text{ gender} = 0$$

נציב את הערכים ב- $h_\theta(X)$:

$$h_\theta(X) = \frac{1}{1 + e^{-(-4.5 + 0.6(4) + 0.03(40) + 0.02(0))}} \approx 0.29$$

ונקבל שההסתברות היא 29% לכך שגידולו הוא סרטני.

3 שיטת הנראות המקסימלית

בפרק זה נציג את שיטת הנראות המקסימלית ונראה את שימושה בגרסיה הלוגיסטית.

שיטת הנראות המקסימלית היא אחת השיטות שסטטיסטיקאים פיתחו להתאמת מודל סטטיסטי לנתונים, כלומר היא משמשת למציאת אמד לפרמטרים המאפיינים את המודל. באופן כללי, שיטה זו מניבה ערכים עבור הפרמטרים הלא ידועים הממקסמים את ההסתברות לקבלת המדגם הנצפה.

שיטה זו היא מקבילה לשיטת הריבועים הפחותים שבאה לאמוד פרמטרים במודלים של רגרסיה לינארית. בהשוואה לשיטת הריבועים הפחותים שיטת הנראות המקסימלית יודעת לאמוד פרמטרים במודלים לא לינארים מורכבים. ומכיוון שהמודל הלוגיסטי הוא מודל לא לינארי, שיטת הנראות המקסימלית היא שיטת האמידה המועדפת לרגרסיה לוגיסטית [4].

3.1 פונקציית הנראות

כדי לתאר את שיטת הנראות המקסימלית נשתמש בפונקציית הנראות, אותה נסמן ב- L . זו היא פונקציה של הפרמטרים הלא ידועים $\theta = \theta_0, \theta_1, \dots, \theta_n$ שאותם נרצה לאמוד בהינתן מדגם מסויים. המשמעות של פונקציית הנראות היא הסבירות לקבל את המדגם שאכן התקבל במציאות מתוך אוכלוסיה מסויימת, כפונקציה של הפרמטר המבוקש θ .

נציג את פונקציית הנראות:

נניח ש Y משתנה מקרי בדיד עם פונקציית הסתברות $P(Y; \theta)$ כאשר θ פרמטר לא ידוע.

תוצאות מדגם מקרי בגודל m שהתקבלו מאוכלוסיה: $Y = Y_1, Y_2, \dots, Y_m$. מניחים שתוצאות המדגם הן בלתי תלויות.

במקרה ואנו יודעים את תוצאות המדגם ולא את הפרמטר נשתמש בפונקציית הנראות שמורכבת ממכפלת ההסתברויות של כל תצפיות המדגם:

$$L(\theta|Y = Y_1, Y_2, \dots, Y_m) =$$

$$P(Y = Y_1; \theta) \cdot P(Y = Y_2; \theta) \cdot \dots \cdot P(Y = Y_m; \theta) = \prod_{i=1}^m P(Y = Y_i; \theta)$$

אם מדובר במשתנה רציף נשתמש במכפלת פונקציות הצפיפות במקום ההסתברות:

$$L(\theta|Y = Y_1, Y_2, \dots, Y_m) = f(Y = Y_1; \theta) \cdot f(Y = Y_2; \theta) \cdot \dots \cdot f(Y = Y_m; \theta) =$$

$$\prod_{i=1}^m f(Y = Y_i; \theta)$$

נהוג להשתמש בלוגריתם הנראות $(l(\theta) = \ln L(\theta))$ אשר נוח יותר מבחינה חישובית. מאחר שהלוגריתם הוא פונקציה מונוטונית עולה ולכן שומרת על יחס סדר, הערך שימקסם את לוג הנראות ימקסם גם את הנראות.

דוגמה:

שדה תעופה ערך מחקר ובחן 100 טיסות בלתי תלויות זו בזו. התקבל מדגם של תוצאות האומרות האם טיסה מסויימת הגיעה בזמן לשדה תעופה או איחרה. הנחת המודל היא שלכל הטיסות יש הסתברות שווה להגיע בזמן לשדה תעופה, אותה אנו מעוניינים להעריך.

מתוצאות המחקר עולה כי 75 טיסות הגיעו בזמן ו-25 איחרו.

אנו מעוניינים לדעת מה הסבירות לקבל את המדגם שאכן התקבל במחקר, בהנחה של ערך מסוים p עבור ההסתברות להגיע בזמן. מידע זה יוכל לשמש אותנו בהמשך למציאת ההסתברות להגיע של טיסה בודדת בזמן.

בדוגמה זו על פי הנחות המודל, תוצאות המחקר מתפלגות ברנולי: Y_i - משתנה מקרי המציין 1 אם טיסה i הגיעה בזמן, ו-0 אם איחרה. p - הסתברות להגיע בזמן של טיסה בודדת. m - מספר הטיסות המשתתפות במחקר (גודל המדגם).

$$Y_i \sim \text{Bernoulli}(p)$$

נציב את תוצאות המדגם שהתקבל ונקבל את פונקציית הנראות עבור מדגם זה, כאשר p הוא הפרמטר הלא ידוע.

$$L(p) = \prod_{i=1}^{100} P(Y = Y_i; p) = \prod_{i=1}^{100} p^{Y_i} (1-p)^{1-Y_i} = p^{75} (1-p)^{25}$$

עבור 75 טיסות שהגיעו בזמן $y_i = 1$ ועבור 25 שאיחרו $y_i = 0$.

פונקציית לוג הנראות:

$$l(p) = \ln(L(p)) = \ln \left(\prod_{i=1}^{100} p^{Y_i} (1-p)^{1-Y_i} \right) = \ln(p^{75} (1-p)^{25})$$

3.2 אומדן הנראות המקסימלי

שיטת הנראות המקסימלית מנסה למצוא את ערך הפרמטר שיוכל להסביר בצורה הטובה ביותר את המדגם שהתקבל. אומדן הנראות המקסימלי הוא הפרמטר שאם היינו מציבים בפונקציית ההסתברות מראש, היה נותן את ההסתברות הגבוהה ביותר לקבל את המדגם שאכן התקבל, ובכך ממקסם את פונקציית הנראות. נסמן את אומדן הנראות המקסימלי ב- $\hat{\theta} = \hat{\theta}_1, \dots, \hat{\theta}_n$.

מדובר בעצם בבעיית אופטימיזציה, נרצה למצוא אומדן עבור קבוצת הפרמטרים הלא ידועים $\theta = \theta_1, \dots, \theta_n$ שממקסמת את פונקציית הנראות $L(\theta)$ עבור מדגם מקרי $Y = Y_1, Y_2, \dots, Y_m$.

$$\hat{\theta} = \arg \max_{\theta \in \Theta} \ln(L(\theta|Y)) = \arg \max_{\theta \in \Theta} \ln \left(\prod_{i=1}^m P(Y_i; \theta) \right)$$

כאשר Θ מרחב הפרמטרים של θ .

דרך אנליטית לפתרון של בעיית אופטימיזציה זו, בהנחה שהנקודה האופטימלית היא נקודה פנימית של Θ , היא לגזור את פונקציית הנראות, להשוות את הנגזרת שלה ל-0 ולמצוא את המקסימום המוחלט (במידה והוא קיים).

ניתן להשתמש בתכונות הלוג כדי להפוך את המכפלה $\ln(\prod_{i=1}^m P(Y_i; \theta))$ לסכום, ובכך להקל על הגזירה.

$$\hat{\theta} = \arg \max_{\theta \in \Theta} \sum_{i=1}^m \ln P(Y_i; \theta)$$

במקרים בהם פונקציית הנראות מורכבת, לא ניתן למצוא פתרון אנליטי, לכן נשתמש בשיטות נומריות/איטרטיביות למציאת מקסימום מוחלט.

נחזור לדוגמת הטיסות ונמצא את אומדן הנראות עבור p :

$$\begin{aligned} l(p) &= \ln \left(\prod_{i=1}^{100} p^{Y_i} (1-p)^{1-Y_i} \right) = \sum_{i=1}^{100} (Y_i \ln(p) + (1-Y_i) \ln(1-p)) = \\ &= \ln((1-p)^{25}) = 75 \ln(p) + 25 \ln(1-p) \end{aligned}$$

נגזור את פונקציית לוג הנראות:

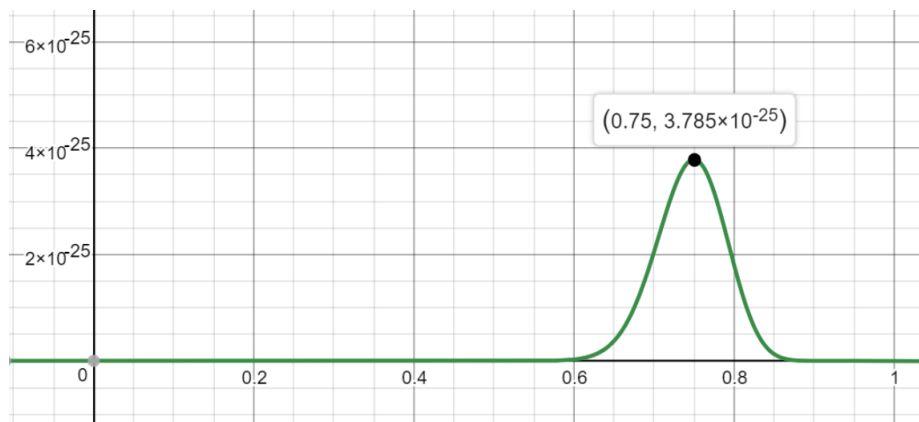
$$\frac{dl(p)}{dp} = \frac{75}{p} - \frac{25}{1-p} = \frac{75-100p}{p(1-p)}$$

נשווה את הנגזרת ל-0:

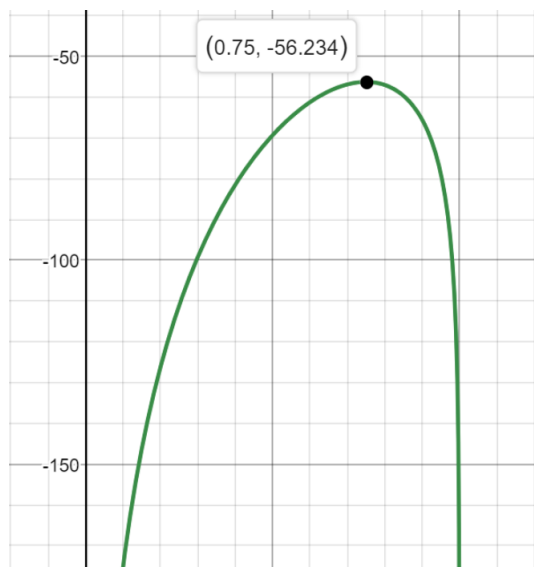
$$\frac{dl(p)}{dp} = 0 \Rightarrow \frac{75-100p}{p(1-p)} = 0$$

$$p = 0.75 \Rightarrow L(p) = 3.785 \cdot 10^{-25}, \quad l(p) = -56.234$$

לכן הערך שממקסם את פונקציית הנראות הוא $\hat{p} = 0.75$. עבור ערך זה $L(p)$ ו- $l(p)$ יהיו מקסימליות ולכן $\hat{p} = 0.75$ זו היא ההסתברות המשוערת לטיסה להגיע בזמן.



איור 5: גרף פונקציית הנראות (דוגמת הטיסות)



איור 6: גרף פונקציית לוג הנראות (דוגמת הטיסות)

3.3 שיטת הנראות המקסימלית בגרסיה לוגיסטית

נראה איך שיטת הנראות המקסימלית באה לידי ביטוי במודל הלוגיסטי.

נזכיר שבמודל לוגיסטי המשתנה התלוי y , בעל התפלגות ברנולי, לכן:

$$(3.1) \quad P(y = 1|X) = h_\theta(X)$$

$$(3.2) \quad P(y = 0|X) = 1 - h_\theta(X)$$

נתאר את תרומתה של תצפית (X_i, y_i) לפונקציית הנראות באמצעות הביטוי:

$$(3.3) \quad [h_\theta(X_i)]^{y_i} [1 - h_\theta(X_i)]^{1-y_i}$$

נשים לב כי עבור $y_i = 1$ בהינתן X_i , ביטוי (3.3) יהיה שווה ל-:

$$[h_\theta(X_i)]^1 [1 - h_\theta(X_i)]^0 = h_\theta(X_i)$$

ועבור $y_i = 0$ בהינתן X_i , הביטוי (3.3) יהיה שווה ל-:

$$[h_\theta(X_i)]^0 [1 - h_\theta(X_i)]^1 = 1 - h_\theta(X_i)$$

בהסתמך על (3.1) ו- (3.2) ניתן לראות כי הביטוי (3.3) מוגדר בצורה נכונה.

אם כל תצפיות הניסוי הן בלתי תלויות, פונקציית הנראות מתקבלת כמכפלת הביטויים (3.3) עבור כל התצפית $i = 1, \dots, m$, כאשר m הוא מספר התצפיות.

$$(3.4) \quad L(\theta|y, X) = \prod_{i=1}^m [h_\theta(X_i)]^{y_i} [1 - h_\theta(X_i)]^{1-y_i}$$

נמצא את פונקציית לוג הנראות:

$$\begin{aligned}
 l(\theta|y, X) &= \ln[L(\theta|y, X)] = \ln \left(\prod_{i=1}^m [h_\theta(X_i)]^{y_i} \cdot [1 - h_\theta(X_i)]^{1-y_i} \right) \\
 &= \sum_{i=1}^m (y_i \cdot \ln(h_\theta(X_i)) + (1 - y_i) \cdot \ln(1 - h_\theta(X_i))) = \\
 &= \sum_{i=1}^m \left(y_i \cdot \ln \left(\frac{1}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) + (1 - y_i) \cdot \ln \left(1 - \frac{1}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) \right) = \\
 &= \sum_{i=1}^m \left(y_i \cdot \ln \left(\frac{1}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) + (1 - y_i) \cdot \ln \left(\frac{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})} - 1}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) \right) = \\
 &= \sum_{i=1}^m \left(\ln \left(\frac{e^{-(\sum_{j=0}^n \theta_j X_{i,j})}}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) + y_i \cdot \left(\ln \left(\frac{1}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) - \ln \left(\frac{e^{-(\sum_{j=0}^n \theta_j X_{i,j})}}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) \right) \right) = \\
 &= \sum_{i=1}^m \left(\ln \left(\frac{e^{-(\sum_{j=0}^n \theta_j X_{i,j})}}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \cdot \frac{e^{(\sum_{j=0}^n \theta_j X_{i,j})}}{e^{(\sum_{j=0}^n \theta_j X_{i,j})}} \right) + y_i \cdot \ln \left(\frac{1}{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \cdot \frac{1 + e^{-(\sum_{j=0}^n \theta_j X_{i,j})}}{e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) \right) = \\
 &= \sum_{i=1}^m \left(\ln \left(\frac{1}{1 + e^{(\sum_{j=0}^n \theta_j X_{i,j})}} \right) + y_i \cdot \left(\ln \frac{1}{e^{-(\sum_{j=0}^n \theta_j X_{i,j})}} \right) \right) = \\
 &= \sum_{i=1}^m \left(\ln(1) - \ln \left(1 + e^{(\sum_{j=0}^n \theta_j X_{i,j})} \right) + y_i \cdot \left(\ln(1) - \ln \left(e^{-(\sum_{j=0}^n \theta_j X_{i,j})} \right) \right) \right) = \\
 &= \sum_{i=1}^m \left(y_i \left(\sum_{j=0}^n \theta_j X_{i,j} \right) - \ln \left(1 + e^{(\sum_{j=0}^n \theta_j X_{i,j})} \right) \right) \tag{3.5}
 \end{aligned}$$

כאשר מרחב הפרמטרים Θ הוא המרחב $\mathbb{R}^{n+1}$.

כדי לאמוד את הערכים של הפרמטרים $\theta_0, \dots, \theta_n$ שממקסמים את פונקציית לוג הנראות נמצא את הנקודות הקריטיות של הפונקציה. הדרך לעשות זאת היא למצוא $n + 1$ הנגזרות החלקיות מסדר ראשון של $l(\theta|y, X)$ לפי $\theta_0, \dots, \theta_n$ ולהשוות אותן ל-0. בפרק 3.4- "קעירות פונקציית לוג הנראות" נוכיח שהנקודות הקריטיות המתקבלות הינן אופטימליות.

נחשב את הנגזרות החלקיות מסדר ראשון:

$$\begin{aligned}
 \frac{\partial l(\theta|y, X)}{\partial \theta_j} &= \frac{\partial}{\partial \theta_j} \sum_{i=1}^m \left(y_i \left(\sum_{j=0}^n \theta_j X_{i,j} \right) - \ln \left(1 + e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)} \right) \right) = \\
 &= \sum_{i=1}^m \frac{\partial}{\partial \theta_j} \left(y_i \left(\sum_{j=0}^n \theta_j X_{i,j} \right) - \ln \left(1 + e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)} \right) \right) = \\
 &= \sum_{i=1}^m \left(y_i X_{i,j} - \frac{e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)}}{1 + e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)}} X_{i,j} \right) = \sum_{i=1}^m \left(y_i - \frac{e^{\left(\theta_0 + X_i \theta_1 \right)}}{1 + e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)}} \right) X_{i,j} = \\
 &= \sum_{i=1}^m \left(y_i - \frac{e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)}}{1 + e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)}} \cdot \frac{e^{-\left(\sum_{j=0}^n \theta_j X_{i,j} \right)}}{e^{-\left(\sum_{j=0}^n \theta_j X_{i,j} \right)}} \right) X_{i,j} = \\
 &= \sum_{i=1}^m \left(y_i - \frac{1}{1 + e^{-\left(\sum_{j=0}^n \theta_j X_{i,j} \right)}} \right) X_{i,j} = \\
 &= \sum_{i=1}^m (y_i - h_\theta(X_i)) X_{i,j} \tag{3.6}
 \end{aligned}$$

כאשר אנו משווים את הביטוי (3.6) ל-0, מתקבלת מערכת של $n + 1$ משוואות לא לינאריות, כאשר כל אחת מהן עם $n + 1$ משתנים לא ידועים. פתרון מערכת זו יתן לנו את וקטור הפרמטרים הלא ידועים. עם זאת, פתרון מערכת משוואות לא לינארית הוא לא קל ולא ניתן לפתור אותה בדרך אנליטית, אך ניתן להשתמש בשיטות איטרטיביות לפתרון מערכות מסוג זה, לדוגמה שיטת ניוטון - רפסון.

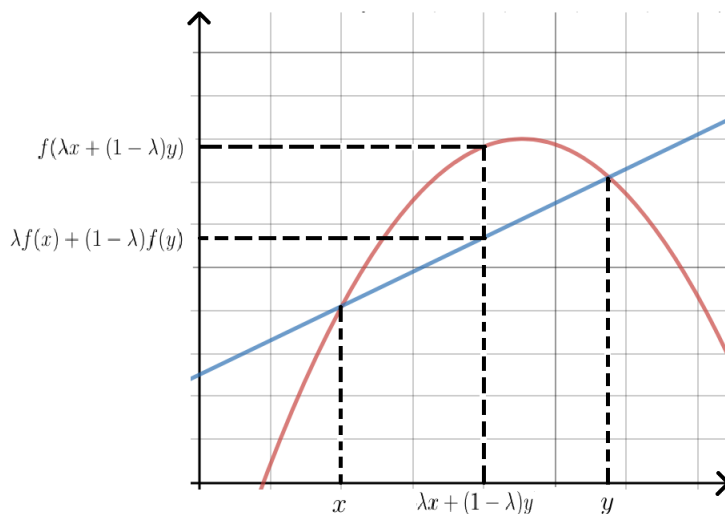
3.4 קעירות פונקציית לוג הנראות

בעיה שנוכל להיתקל בה בשימוש בשיטה איטרטיבית היא שהפתרון יתכנס למקסימום מקומי במקום להתכנס למקסימום מוחלט וכך פונקציית לוג הנראות לא תקבל ערך מקסימלי. לפיכך עלינו לוודא שהפתרון אכן יתכנס למקסימום מוחלט. לצורך כך נראה שפונקציית לוג הנראות היא פונקציה קעורה, משום שכל נקודה קריטית של פונקציה קעורה (במידה והיא קיימת) היא נקודת מקסימום מוחלט [3].

נגדיר פונקצייה קעורה:

פונקציה $f : \mathbb{R}^d \rightarrow \mathbb{R}$ תקרא קעורה אם עבור כל $x, y \in \mathbb{R}^d$ ו- $\lambda \in (0, 1)$ מתקיים האי שוויון:

$$f(\lambda x + (1 - \lambda)y) \geq \lambda f(x) + (1 - \lambda)f(y)$$



איור 7: פונקציה קעורה

כלומר, הישר המחבר בין שתי הנקודות נמצא מתחת לגרף הפונקציה (במקרה של שוויון הישר נמצא על גרף הפונקציה).

נוכיח כעת מספר טענות אשר יישמשו אותנו בהמשך.

טענה 3.1 תהי $f \in C^1(\mathbb{R}^d, \mathbb{R})$ פונקציה קעורה, אזי לכל $x, y \in \mathbb{R}^d$ מתקיים האי שוויון:

$$f(y) - f(x) \leq \nabla f(x)(y - x)$$

הוכחה f קעורה ולכן לכל $x, y \in \mathbb{R}^d$ ו- $\lambda \in (0, 1)$ מתקיים:

$$f(\lambda y + (1 - \lambda)x) \geq \lambda f(y) + (1 - \lambda)f(x)$$

נחלק ב λ :

$$\frac{f(x + \lambda(y - x)) - f(x)}{\lambda} \geq f(y) - f(x)$$

נשאיף את λ ל-0 ונקבל:

$$f(y) - f(x) \leq \nabla f(x)(y - x)$$

■

משפט 3.1 תהיי פונקציה $f \in C^1(\mathbb{R}^d, \mathbb{R})$ קעורה ו- $\bar{x} \in \mathbb{R}^d$ נקודה קריטית, אזי $\bar{x}$ נקודת מקסימום מוחלט.

הוכחה לפי טענה 3.1 פונקציה קעורה מקיימת את האי שוויון לכל $\bar{x}, y \in \mathbb{R}^d$:

$$f(y) - f(\bar{x}) \leq \nabla f(\bar{x})(y - \bar{x})$$

משום ש- $\bar{x}$ נקודה קריטית $\nabla f(\bar{x}) = 0$ נקבל:

$$f(y) \leq f(\bar{x})$$

הנקודה הקריטית $\bar{x}$ היא נקודת מקסימום מוחלט. ■

טענה 3.2 $f(x) \in C^2(\mathbb{R}, \mathbb{R})$ היא פונקציה קעורה אם $f''(x) \leq 0$ לכל $x \in \mathbb{R}$.

הוכחה יהיו $a, b, c \in \mathbb{R}$ כך ש- $a < b < c$. $f(x) \in C^2(\mathbb{R}, \mathbb{R})$ ולכן מקיימת את תנאי משפט הערך הממוצע של לגראנז'. לפי המשפט קיימת נקודה $x' \in \mathbb{R}$ כאשר $a < x' < b$ כך ש:

$$f'(x') = \frac{f(b) - f(a)}{b - a}$$

בנוסף קיימת נקודה $x'' \in \mathbb{R}$ כאשר $b < x'' < c$ כך ש:

$$f'(x'') = \frac{f(c) - f(b)}{c - b}$$

אם לכל $x \in \mathbb{R}$ מתקיים ש- $f''(x) \leq 0$, אזי $f'(x)$ יורדת במובן החלש לכל $x \in \mathbb{R}$. לפיכך, $x' < x''$

$$f'(x'') \leq f'(x') \Rightarrow$$

$$\frac{f(c) - f(b)}{c - b} \leq \frac{f(b) - f(a)}{b - a} \Rightarrow$$

$$(f(c) - f(b))(b - a) \leq (f(b) - f(a))(c - b) \Rightarrow$$

$$f(c)(b - a) + f(a)(c - b) \leq f(b)(c - a) \Rightarrow$$

$$f(a) \left(\frac{c - b}{c - a} \right) + f(c) \left(\frac{b - a}{c - a} \right) \leq f(b) \quad (3.2)$$

בהינתן $a, c \in \mathbb{R}$ ניתן לקבל כל ערך של λ בין 0 ל-1 על ידי בחירה מתאימה של b בין a ל- c :

$$b = c(1 - \lambda) + a\lambda \Rightarrow \lambda = \frac{c - b}{c - a} \Rightarrow 1 - \lambda = \frac{b - a}{c - a}$$

נציב את הערכים אלו ב-(3.2):

$$f(c(1 - \lambda) + a\lambda) \geq f(a)\lambda + f(c)(1 - \lambda)$$

קבלנו תנאי לקעירות ולכן הפונקציה f קעורה. ■

טענה 3.3 סכום פונקציות קעורות הוא פונקציה קעורה.

הוכחה נגדיר פונקציה $h : \mathbb{R}^d \rightarrow \mathbb{R}$:

$$h(x) = f(x) + g(x)$$

כאשר $f(x)$ ו- $g(x)$ פונקציות קעורות.

עבור $x, y \in \mathbb{R}^d$ ו- $\lambda \in (0, 1)$ נראה כי האי שוויון -

$$h(\lambda x + (1 - \lambda)y) \geq \lambda h(x) + (1 - \lambda)h(y)$$

מתקיים ובכך נוכיח את הטענה.

$$h(\lambda x + (1 - \lambda)y) = f(\lambda x + (1 - \lambda)y) + g(\lambda x + (1 - \lambda)y) \geq$$

$$\geq \lambda f(x) + (1 - \lambda)f(y) + \lambda g(x) + (1 - \lambda)g(y) =$$

$$= \lambda(f(x) + g(x)) + (1 - \lambda)(f(y) + g(y)) =$$

$$= \lambda h(x) + (1 - \lambda)h(y)$$

■

טענה 3.4 כל פונקציה מהצורה $f(\theta) = a^T \theta$ כאשר $\theta \in \mathbb{R}^d$ ו- $a \in \mathbb{R}^d$ קעורה.

הוכחה נקח $\theta_1, \theta_2 \in \mathbb{R}^d$ ו- $\lambda \in (0, 1)$.

$$(1) \quad f(\lambda \theta_1 + (1 - \lambda)\theta_2) = a^T(\lambda \theta_1 + (1 - \lambda)\theta_2) = \lambda a^T \theta_1 + (1 - \lambda)a^T \theta_2$$

$$(2) \quad \lambda f(\theta_1) + (1 - \lambda)f(\theta_2) = \lambda(a^T \theta_1) + (1 - \lambda)a^T \theta_2$$

קיבלנו שוויון בין שני הביטויים ולכן התנאי לקעירות מתקיים. ■

טענה 3.5 הפונקציה $f(u) = -\ln(1 + e^u)$ קעורה.

הוכחה נבדוק את הנגזרת השנייה של $f(u)$:

$$f'(u) = \frac{-e^u}{1 + e^u}$$

$\Downarrow$

$$f''(u) = \frac{-e^u(1+e^u) - e^u(-e^u)}{(1+e^u)^2} = \frac{-e^u}{(1+e^u)^2} \leq 0$$

הביטויים e^u ו- $(1+e^u)^2$ אי שליליים, לפיכך הנגזרת השניה של $f(u)$ אי חיובית לכל $u \in \mathbb{R}$ ולכן לפי טענה 3.2 היא פונקציה קעורה. ■

טענה 3.6 $g(\theta) = f(a^T \theta)$ פונקציה קעורה, כאשר $\theta \in \mathbb{R}^d, a \in \mathbb{R}^d$ ו- $f(u)$ פונקציה קעורה.

הוכחה נקח $\theta_1, \theta_2 \in \mathbb{R}^d$ ו- $\lambda \in (0, 1)$:

$$g(\lambda\theta_1 + (1-\lambda)\theta_2) = f(a^T(\lambda\theta_1 + (1-\lambda)\theta_2)) = f(\lambda a^T \theta_1 + (1-\lambda)a^T \theta_2) \geq$$

$$\geq \lambda f(a^T \theta_1) + (1-\lambda)f(a^T \theta_2) = \lambda g(\theta_1) + (1-\lambda)g(\theta_2)$$

קבלנו

$$g(\lambda\theta_1 + (1-\lambda)\theta_2) \geq \lambda g(\theta_1) + (1-\lambda)g(\theta_2)$$

כלומר, $g(\theta)$ פונקציה קעורה. ■

כעת הגענו לתוצאה המרכזית בסעיף זה:

משפט 3.2 פונקציה לוג הנראות $l \in C^2(\mathbb{R}^{n+1}, \mathbb{R})$:

$$l(\theta|y, X) = \sum_{i=1}^m \left(y_i \left(\sum_{j=0}^n \theta_j X_{i,j} \right) - \ln \left(1 + e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)} \right) \right)$$

היא פונקציה קעורה.

הוכחה נשים לב שפונקציית לוג הנראות $l(\theta|y, X)$ מורכבת משני סכומים:

$$(1) \quad \sum_{i=1}^m y_i \left(\sum_{j=0}^n \theta_j X_{i,j} \right)$$

$$(2) \quad - \sum_{i=1}^m \ln \left(1 + e^{\left(\sum_{j=0}^n \theta_j X_{i,j} \right)} \right)$$

לפי הטענה 3.3 מספיק להראות שפונקציית לוג הנראות היא סכום של פונקציות קעורות.

הסכום הראשון של פונקציית לוג הנראות הוא סכום פונקציות מהצורה $a^T \theta$ לפיכך לפי טענה 3.4 הוא קעור.

הסכום השני של פונקציית לוג הנראות הוא סכום פונקציות מהצורה:

$$g(\theta) = -\ln(1 + e^{a^T \theta})$$

לפיכך לפי טענות 3.5 ו- 3.5 הסכום קעור.

מטענה 3.3 נובע שפונקציית לוג הנראות $l(\theta|y, X)$ קעורה. ■

לאחר שהראינו שפונקציית לוג הנראות היא פונקצייה קעורה נוכל להסיק מתוך משפט 3.2 שכל נקודת מקסימום מקומי של הפונקציה הינה נקודת מקסימום מוחלט, ולכן וקטור הפתרונות שנקבל משיטה איטרטיבית (במידה והשיטה האיטרטיבית מתכנסת לנקודה קריטית) מכיל את ערכי הפרמטרים שעבורם הסבירות לקבל את המדגם היא מקסימלית.

4 בחינת מידת הצלחת המודל

ברצונינו לבדוק אם ההסתברויות המחושבות על ידי המודל עבור תצפיות חדשות (תחזיות), משקפות בצורה איכותית את התוצאות האמיתיות של אותן תצפיות. נרצה לבדוק זאת בשביל שנוכל למדוד את איכות התחזיות - עד כמה הן מדויקות? וכדי להשוות בין מודלים שונים - עד כמה מודל חיזוי אחד טוב יותר מאחר? לצורך כך קיימות מספר שיטות שונות, במסגרת עבודה זו נסקור את שיטת ציון ברייר [1].

ציון ברייר (Brier score)

ציון ברייר הוא מדד להערכת הדיוק של תחזית הסתברותית על ידי מתן ציון שנע בין 0 ל-1, כאשר 0 מעיד על דיוק מלא ו-1 מעיד על חוסר דיוק מוחלט. תחזית להסתברות כלשהי מתייחסת לאירוע מסוים, לדוגמה בהסתברות של 25% יהיה עיכוב בטיסת אל על מישראל ליוון.

שיטה זו שימושית רק לתוצאות בינאריות, כאשר יש רק שני אירועים אפשריים, כמו "הטיסה התעכבה" או "הטיסה לא התעכבה". לפיכך שיטה זו מתאימה לרגרסיה הלוגיסטית.

הציון מחושב באופן הבא עבור תצפית i :

$$BS = (y_i - P(y_i = 1))^2$$

ניתן לחשב ציון ברייר עבור מספר תחזיות (נוסחת טעות ריבועית ממוצעת):

$$BS = \frac{1}{m} \sum_{i=1}^m (y_i - P(y_i = 1))^2$$

במקרה של רגרסיה לוגיסטית:

$$BS = \frac{1}{m} \sum_{i=1}^m (y_i - h_{\theta}(X_i))^2$$

כאשר:

m - מספר התחזיות החדשות שעבורם נרצה לקבל ציון ברייר.

$h_\theta(X_i)$ ההסתברות לקבל $y_i = 1$ עבור תצפית i .

y_i תוצאה בפועל של תצפית i .

ציון ברייר הינה דרך סבירה להערכת הדיוק של תחזית הסתברותית משום שעבור תצפית i מסויימת כאשר התוצאה בפועל עבורה שווה ל-1 ($y_i = 1$) השאיפה היא שההסתברות הנחזית ל- $y_i = 1$ תהיה כמה שיותר גבוהה, וכאשר התוצאה בפועל שווה ל-0 ההסתברות ל- $y_i = 1$ תהיה כמה שיותר נמוכה. ולכן נרצה שההפרש בין התוצאה בפועל לבין ההסתברות ל- $y_i = 1$ יהיה מינימלי.

דוגמה

התחזית אתמול הייתה שההסתברות לגשם היא 89%.
אתמול בשעה 4 אחר הצהריים ירד גשם.

כדי לחשב את ציון ברייר עבור התחזית נכניס את הנתונים לנוסחה:

ההסתברות לגשם היא 0.89 ולכן $P(y = 1) = 0.89$. בנוסף ידוע שהיה גשם ולכן $y = 1$.

$$BS = (1 - 0.89)^2 = 0.0121$$

קבלנו ציון נמוך ולכן ניתן לומר שתחזית ההסתברות היא מדוייקת.

ציון ברייר מודד עד כמה תחזיות הינן מדוייקות, אך לא ביחס למודל אחר. אפילו אם הציון במודל שאנו בוחנים עשוי להיות נמוך, מודל פשוט שנותן תחזית קבועה לכל התצפיות החדשות עשוי להיות טוב יותר.

על מנת להשוות בין שני מודלים שונים נשתמש ב-**מדד המיומנות של ברייר**. בעוד שציון ברייר עונה על השאלה "מהו גודל השגיאה של התחזיות?" מדד המיומנות של ברייר עונה על השאלה "עד כמה התחזיות במודל שלנו מדוייקות ביחס למודל אחר?". מקובל להשתמש במדד המיומנות של ברייר לצורך השוואה עם מודל שנותן תחזית מסויימת זהה לכל תצפית חדשה, מודל כזה נקרא מודל סטנדרטי.

לדוגמה, במדגם מסויים היו m תצפיות שעבורם $y_i = 1$ ו- n תצפיות שעבורם $y_i = 0$. התחזית שמודל סטנדרטי יתן עבור התצפיות החדשות תהיה $\frac{m}{m+n}$.

נוסחת מדד המיומנות של ברייר:

$$BSS = 1 - \frac{BS}{BS_{standard}}$$

BS - ציון ברייר של המודל הנבחן.

$BS_{standard}$ - ציון ברייר של המודל הסטנדרטי.

BSS נע בין $-\infty$ ל-1. ערך שלילי מעיד כי התחזיות פחות מדויקות מאשר תחזיות במודל הסטנדרטי, 0 מעיד כי איכות התחזיות זהה בשני המודלים ו-1 מעיד כי התחזיות מדויקות ביחס למודל סטנדרטי.

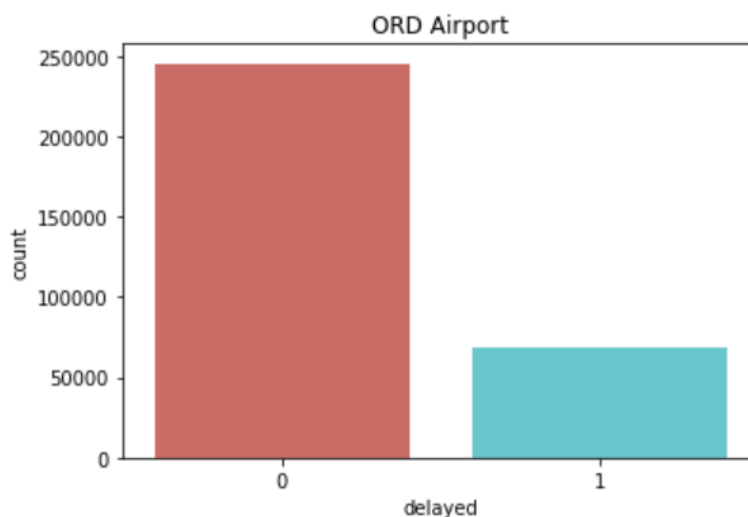
5 הצגת המודל "עיקובי טיסות"

5.1 רקע

בחרתי להמחיש את השימוש בגרסיה הלוגיסטית באמצעות דוגמה בה אבנה מודל שינסה לאמוד את ההסתברות לעיכוב טיסה. בעיה זו מעניינת במיוחד לחברות ביטוח לטיסות, ולחברות התעופה שכן האפשרות לחזות מראש איחור טיסה יכולה לעזור להן להיערך בהתאם ואף לחסוך כסף רב.

לקחתי מאתר *kaggle* מאגר נתונים על טיסות פנימיות בארצות הברית בשנת 2015. המאגר כלל 5,819,079 טיסות כך שעל כל טיסה סופקו ערכים שנמדדו ויכולים לשמש כמשתנים בלתי תלויים, כגון - תאריך הטיסה, משך הטיסה, זמן אוויר, זמן המראה, מוצא, יעד ועוד, ובנוסף מידע לגבי זמן האיחור של הטיסה שיכול לשמש כמשתנה התלוי (לאחר הפיכתו למשתנה בינארי).

מאחר וישנה כמות גדולה של מידע וברצוני רק להדגים את השימוש בגרסיה הלוגיסטית, די בכך שאבחר תת קבוצה של הנתונים - שדה תעופה מוצא ספציפי. מתוך שדות התעופה עם כמות הטיסות הכי גבוהה בחרתי את שדה התעופה עם אחוז העיכובים הכי גבוה, על מנת לוודא שיש בידי מספיק נתונים על טיסות שאיחרו. שדה התעופה שנבחר הוא *ORD* - שיקגו או'הייר עם 304,135 טיסות יוצאות בשנה וכ- 22% של עיכובים (נספח פרק 4).



המידע שבחרתי להשתמש בו מתוך מאגר הנתונים:

המשתנים הבלתי תלויים:

1. חברת תעופה - משתנה קטגורי.
2. שדה תעופה היעד - משתנה קטגורי.
3. חודש בשנה - משתנה קטגורי.
4. יום בשבוע - משתנה קטגורי.
5. שעת יציאה מתוכנית - משתנה קטגורי.
6. עומס - משתנה נומרי. ניתן למדוד את עומס הטיסות בדרכים שונות. אני בחרתי לספור כמה טיסות המריאו באותה שעה.

המשתנה התלוי:

עיכוכי הטיסה - 1 אם הטיסה התעכבה ביותר מ- 15 דקות בהגעה ליעד ו-0 אם לא.

MONTH	DAY	DAY_OF_WEEK	AIRLINE	DESTINATION_AIRPORT	delayed	SCHEDULED_DEPARTURE_intervals	LOAD
0	5	10	7	MQ	TVC	0	21.0 49
1	6	6	6	NK	IAH	0	9.0 62
2	6	21	7	MQ	LEX	0	21.0 34
3	1	22	4	AA	RDU	0	10.0 45
4	11	24	2	EV	ORF	0	14.0 37
5	12	4	5	UA	IAH	0	13.0 68
6	3	30	1	AA	RDU	0	15.0 67
7	8	11	2	AA	PDX	1	20.0 75
8	8	28	5	EV	GSO	0	18.0 52
9	9	6	7	OO	SYR	0	13.0 68
10	9	30	3	AA	DFW	0	14.0 38
11	2	13	5	EV	CVG	0	9.0 60
12	11	15	7	UA	AUS	0	15.0 66
13	11	12	4	OO	MSN	0	22.0 22
14	11	25	3	DL	ATL	0	12.0 65
15	8	8	6	UA	LAX	0	15.0 54
16	8	17	1	MQ	ROC	1	15.0 57

איור 8: כמה שורות מנתוני המאגר

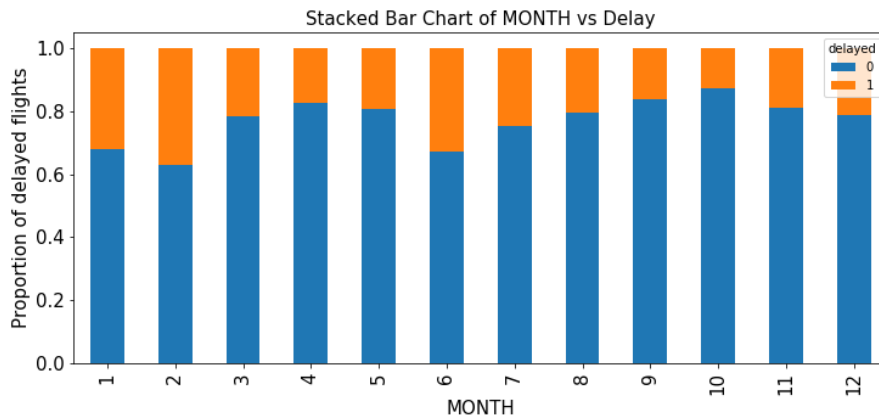
מטרתי היא לבנות מודל שיתן את ההסתברות לעיכוכי טיסה היוצאת משדה התעופה שיקגו או'הייר - *ORD* בהינתן הנתונים שצויינו.

5.2 ניתוח ראשוני של הנתונים

(נספח פרק 7)

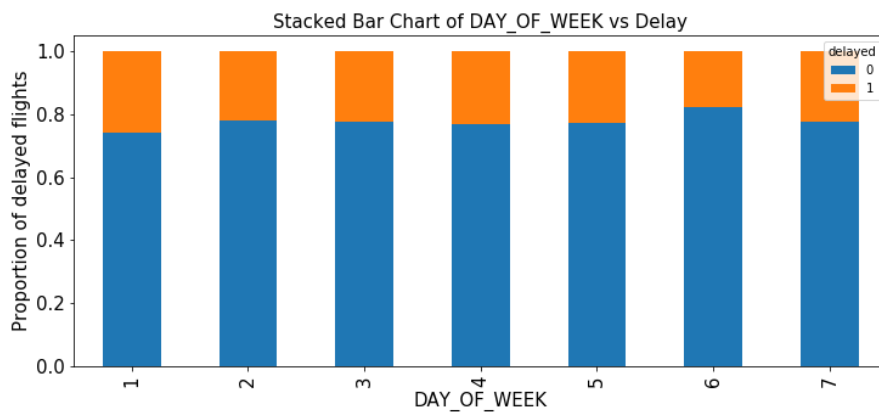
נציג עבור כל משתנה את גרף היחס בין הטיסות המאוחרות לטיסות שהגיעו בזמן.

אחוז עיכובי טיסות לפי חודשי שנה:



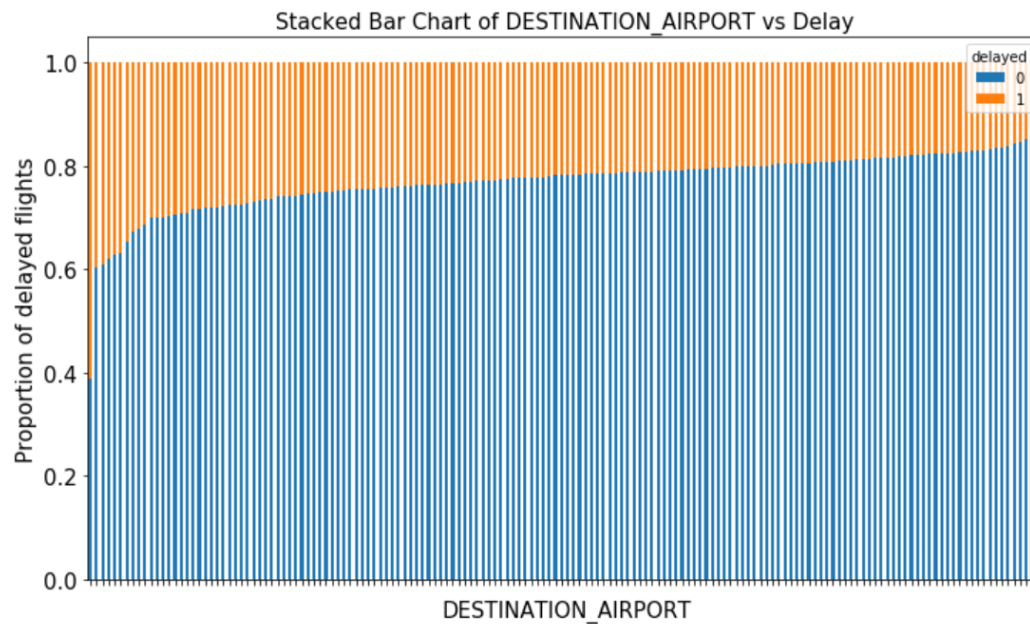
נשים לב שבחודשים ינואר, פברואר ויוני אחוז עיכובי הטיסות גבוה יותר משאר החודשים. נראה שהמשתנה "חודש בשנה" יכול להיות משמעותי לחיזוי עיכובי הטיסות.

אחוז עיכובי טיסות לפי יום בשבוע:



ניתן לראות מהגרף שבכל ימות השבוע אחוז עיכובי טיסות כמעט זהה למעט יום שישי, שעבורו ניתן לראות שינוי קל באחוז עיכובי הטיסות. לפיכך קיים חשד שהמשתנה "יום בשבוע" לא משמעותי לחיזוי עיכובי הטיסות.

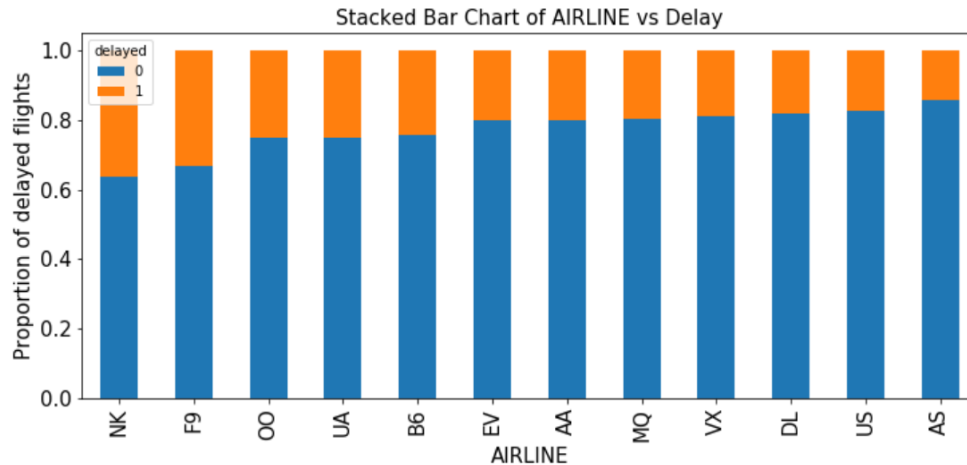
אחוז עיכובי טיסות לפי שדה תעופה יעד בסדר יורד:



שדות תעופה היעד עם פחות מ- 100 טיסות בשנה הם בעלי אחוז עיכובי טיסות הגבוה ביותר. הבולט מביניהם הוא *FAI* - נמל תעופה באלסקה עם יותר מ- 60% עיכובים.

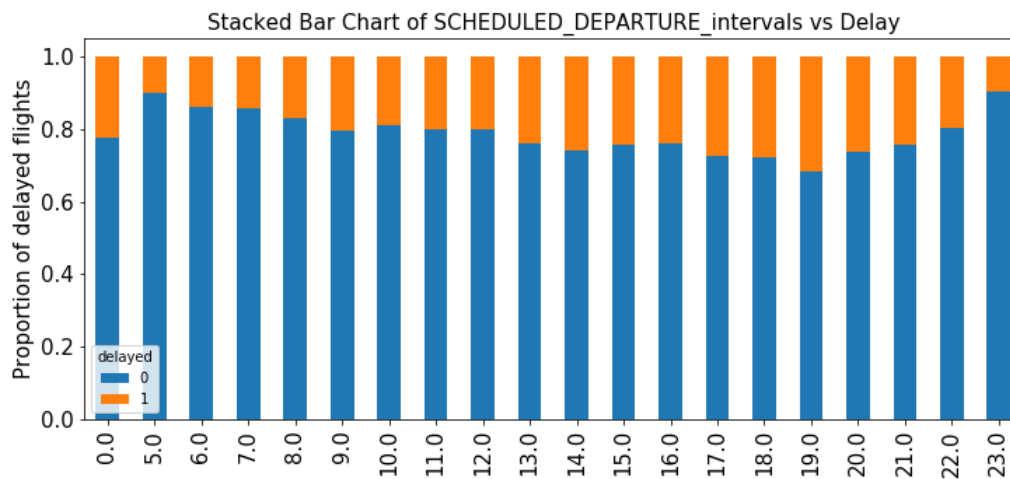
מהגרף ניתן לראות שהמשתנה "שדה תעופה היעד" יכול להיות משמעותי לחיזוי עיכובי הטיסה.

אחוז עיכובי טיסות לפי חברת תעופה:

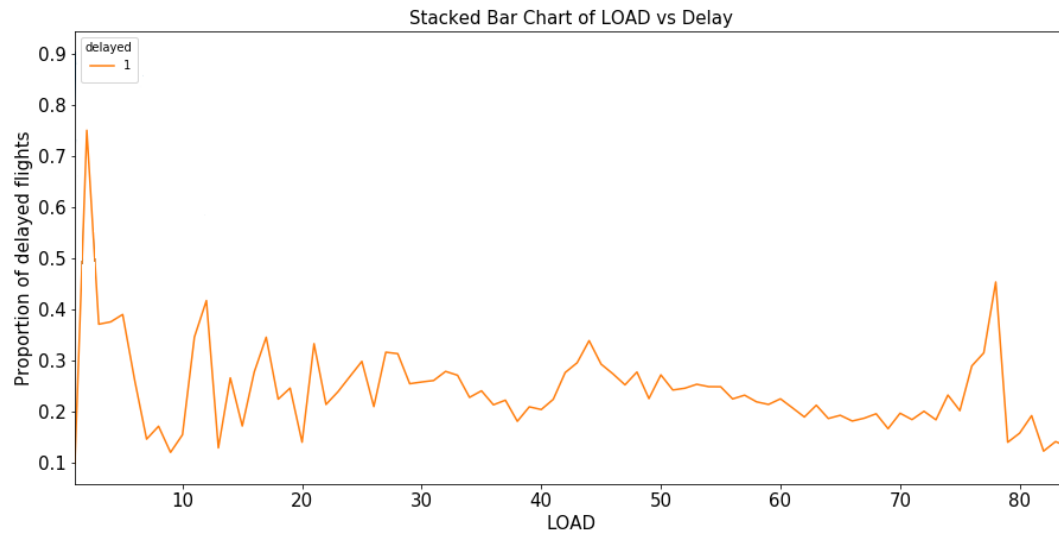


קיימות שתי חברות תעופה $F9$ - פרונטיר איירליינס ו- NK - ספרייט איירליינס, עם אחוז עיכובי טיסות גבוה במיוחד. החברות הנ"ל הן חברות שמספקות טיסות זולות לנוסעים. נראה שהמשתנה "חברת תעופה" יכול להיות משמעותי לחיזוי עיכובי הטיסה.

אחוז עיכובי טיסות לפי שעת יציאה מתוכננת:



נשים לב שקיימת הפסקה בטיסות בין השעות 01 : 00 ל- 00 : 05. נראה שהמשתנה "שעת יציאה מתוכננת" יכול להיות משמעותי לחיזוי עיכובי הטיסה.

אחוז עיכובי טיסות לפי עומס בשעת ההמראה:

נשים לב שבעומס של 2 ו- 78 מטוסים בשעה אחוז עיכובי טיסות הוא מקסימלי. נראה שהמשתנה "עומס" יכול להיות משמעותי לחיזוי עיכובי הטיסה.

5.3 שלבי בניית המודל

5.3.1 משתנים קטגוריים למשתני דמה

(נספח פרק 8)

כפי שהוסבר בפרק 2, לא נוכל להשתמש במשתנים הקטגוריים בצורתם הנוכחית. לפיכך, נהפוך את כל המשתנים הקטגוריים למשתני דמה ובהם נשתמש במודל שלנו. למשל, את המשתנה "חודש בשנה" נהפוך ל-11 משתני דמה:

$$month\ 2, month\ 3, month\ 4, \dots, month\ 12$$

אם אחד הנתונים של תצפית i הוא חודש ספטמבר אז ערכו של משתנה הדמה $month\ 9$ יהיה שווה ל-1 וכל שאר משתני הדמה של חודשי השנה, 0.

הערה: חודש ינואר לא מופיע, משום שאם למשתנה קטגורי ישנם k אפשרויות אז נזדקק ל- $k - 1$ משתני דמה. ולכן כאשר כל משתני הדמה של חודשי השנה בעלי ערך 0 נדע שמדובר בחודש הינואר.

	MONTH_2	MONTH_3	MONTH_4	MONTH_5	MONTH_6	MONTH_7	MONTH_8	MONTH_9	MONTH_10	MONTH_11	MONTH_12
0	0	0	0	0	0	1	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	1	0	0
3	0	0	1	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	1	0	0	0
6	0	0	0	0	0	0	0	0	0	1	0
7	0	0	0	0	0	0	0	0	0	0	1
8	0	0	0	0	0	0	0	0	0	0	1
9	0	0	0	0	0	0	0	0	1	0	0
10	0	0	0	0	0	0	0	1	0	0	0
11	0	0	0	0	1	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	1	0
13	0	0	0	0	0	0	0	0	0	0	1

איור 9: טיסות לפי חודשי השנה

נשתמש בשיטה זו לכל שאר המשתנים הקטגוריים. לבסוף נקבל 207 משתנים בלתי תלויים.

5.3.2 חלוקה לקבוצת אימון וקבוצת בדיקה

(נספח פרק 9)

על מנת שנוכל גם לאמן את המודל שלנו לחיזוי עיכוב טיסה וגם לבדוק את נכונות החיזויים נחלק את נתוני הטיסות לקבוצת אימון (*training set*) וקבוצת בדיקה (*testing set*). בעזרת קבוצת האימון נלמד את המודל ונמצא את וקטור הפרמטרים $\hat{\theta}$, ובעזרת קבוצת הבדיקה נקבל תחזיות עבור תצפיות שהמודל עוד לא נתקל בהם.

חלוקת נתונים נעשתה לפי 70% קבוצת אימון ו- 30% קבוצת בדיקה. כל קבוצה כללה בניפרד את התצפיות והתוצאות עבור אותן תצפיות שנבחרו לכל קבוצה.

וידאתי שאין טיסות מאותו סוג בשתי הקבוצות כדי להבטיח שבקבוצת הבדיקה יהיו טיסות חדשות בלבד שהמודל לא ראה לפני כן.

5.3.3 התאמת המודל

(נספח פרק 10)

למידת המודל התבצעה בעזרת קבוצת האימון, פונקציית לוג הנראות ואחת מהשיטות האיטרטיביות לפתרון מערכות משוואות לא לינאריות, כאשר המטרה היא למצוא את וקטור הפרמטרים הלא ידועים $\hat{\theta}$. לשם כך השתמשתי בפונקציה *LogisticRegression* מספריית הפייטון - *sklearn*.

לאחר הלמידה התקבלו ערכי $\hat{\theta}$ עבור המשתנים הבלתי תלויים. ניתן לראות אותם בנספח פרק 10.

השוואה בין ניתוח ראשוני לפרמטרים שהתקבלו:

לאחר ביצוע הניתוח הראשוני לנתוני המאגר קיבלנו מידע לגבי אחוז עיכובי הטיסות עבור המשתנים הבלתי תלויים. ראינו שהחודשים ינואר, פברואר ויוני בעלי אחוזי עיכוב הטיסות הגבוהים ביותר. נראה היה שהחודשים האלו מגדילים את ההסתברות לעיכוב טיסה ולכן נצפה שערכי הפרמטרים הלא ידועים של אותם חודשים יהיו חיוביים וגבוהים יותר משאר החודשים. נצפה לראות תופעה דומה גם עבור שדות תעופה היעד וחברות התעופה שעבורם אחוז עיכובי הטיסה גבוה. לעומת זאת, עבור המשתנה יום בשבוע נצפה לראות שינוי קטן בערכי הפרמטרים למעט יום שישי שעבורו ערך הפרמטר יהיה הנמוך ביותר. עבור שעת יציאה מתוכננת נראה שערכי הפרמטרים נמוכים במיוחד בשעות 23.0 ו- 5.0 (השעות עם אחוז עיכובי הטיסות

הנמוך ביותר). עבור שעות אלו נקבל את ההסתברות הנמוכה ביותר לעיכוב הטיסה.

Weights			Weights			Weights		
LOAD	-0.030350	AIRLINE_DL	-0.041496	DESTINATION_AIRPORT_XNA	0.084697			
MONTH_10	-0.948988	AIRLINE_EV	0.176824	SCHEDULED_DEPARTURE_intervals_10.0	-0.032407			
MONTH_11	-0.544356	AIRLINE_F9	0.758465	SCHEDULED_DEPARTURE_intervals_11.0	-0.446376			
MONTH_12	-0.455869	AIRLINE_MQ	0.207363	SCHEDULED_DEPARTURE_intervals_12.0	0.408743			
MONTH_2	0.273582	AIRLINE_NK	0.907488	SCHEDULED_DEPARTURE_intervals_13.0	0.837839			
MONTH_3	-0.392045	AIRLINE_00	0.527945	SCHEDULED_DEPARTURE_intervals_14.0	0.153291			
MONTH_4	-0.635430	AIRLINE_UA	0.272611	SCHEDULED_DEPARTURE_intervals_15.0	0.526643			
MONTH_5	-0.512231	AIRLINE_US	-0.229309	SCHEDULED_DEPARTURE_intervals_16.0	0.289048			
MONTH_6	0.286330	AIRLINE_VX	-0.195925	SCHEDULED_DEPARTURE_intervals_17.0	0.867287			
MONTH_7	-0.109233	DESTINATION_AIRPORT_ABQ	0.373958	SCHEDULED_DEPARTURE_intervals_18.0	0.634737			
MONTH_8	-0.334322	...	...	SCHEDULED_DEPARTURE_intervals_19.0	0.775422			
MONTH_9	-0.708136	DESTINATION_AIRPORT_STT	0.039692	SCHEDULED_DEPARTURE_intervals_20.0	0.765426			
DAY_OF_WEEK_2	-0.242531	DESTINATION_AIRPORT_SUX	-0.182211	SCHEDULED_DEPARTURE_intervals_21.0	-0.034091			
DAY_OF_WEEK_3	-0.165971	DESTINATION_AIRPORT_SYR	-0.039301	SCHEDULED_DEPARTURE_intervals_22.0	-0.939922			
DAY_OF_WEEK_4	-0.135024	DESTINATION_AIRPORT_TOL	-0.485962	SCHEDULED_DEPARTURE_intervals_23.0	-0.863806			
DAY_OF_WEEK_5	-0.140418	DESTINATION_AIRPORT_TPA	0.109805	SCHEDULED_DEPARTURE_intervals_5.0	-2.131205			
DAY_OF_WEEK_6	-0.817044	DESTINATION_AIRPORT_TTN	-0.304041	SCHEDULED_DEPARTURE_intervals_6.0	-1.027011			
DAY_OF_WEEK_7	-0.255898	DESTINATION_AIRPORT_TUL	0.022627	SCHEDULED_DEPARTURE_intervals_7.0	-0.030174			
AIRLINE_AS	-0.686307	DESTINATION_AIRPORT_TUS	0.265624	SCHEDULED_DEPARTURE_intervals_8.0	0.214807			
AIRLINE_B6	0.104083	DESTINATION_AIRPORT_TVC	-0.252156	SCHEDULED_DEPARTURE_intervals_9.0	0.471456			
AIRLINE_DL	-0.041496	DESTINATION_AIRPORT_TYS	0.091546					

איור 10: ערכי הפרמטרים של המודל

ואכן, בחלק מהמקרים זהו המצב. ניתן לראות לפי איור 10 שערכי הפרמטרים של חברות תעופה $F9$ ו- NK וחודשים פברואר ויוני, חיובים וגבוהים. בנוסף ניתן לראות שינוי קל בערכי הפרמטרים של המשתנה יום בשנה למעט יום שישי הבולט מביניהם. לעומת זאת, עבור המשתנה שעת יציאה מתוכננת אנו לא מקבלים את ערכי הפרמטרים להם צפינו. לדוגמה, הפרמטר של השעה 6 נמוך יותר משעה 23. הסיבה לכך יכולה להיות שאחוז עיכובי טיסות של משתנה בלתי תלוי מסוים הושפע מאחוז עיכובי טיסות של משתנה בלתי תלוי אחר. לדוגמה, בשעה 23 אין טיסות של חברות תעופה בעלי אחוז עיכובי טיסות גבוה ולכן בניתוח ראשוני שעה זו הייתה בעלת אחוז עיכובי טיסות נמוך, אך לאחר הלמידה המודל מצא שההסתברות לעיכוב בשעה 6 תהיה יותר נמוכה משעה 23.

5.3.4 בחינת המודל

(נספח פרק 11)

לאחר קבלת התוצאות נרצה לבחון את מידת ההצלחה של המודל. לשם כך נשתמש

במדד המיומנות של ברייר שהוצג בפרק 4.

נחשב את מדד המיומנות של ברייר של המודל. לשם כך נידרש למצוא את המודל הסטנדרטי ואת התחזיות עבור תצפיות הבדיקה. במקרה שלנו המודל הסטנדרטי הוא מודל שחווה הסתברות איחור לכל טיסה ששווה לאחוז עיקובי הטיסות של תצפיות הבדיקה - 0.224. נציב את ערכי $\hat{\theta}$ בפונקציה הלוגיסטית ונקבל תחזיות עבור לתצפיות הבדיקה. ניתן לראות אותן בנספח פרק 10.

כעת אנו יכולים לחשב את מדד המיומנות של ברייר :

$$BSS = 1 - \frac{BS}{BS_{standard}} = 1 - \frac{0.1622}{0.1741} = 0.0679$$

קיבלנו תוצאה חיובית ולכן נראה שהמודל שלנו מפיק חיזויים טובים יותר מהמודל הסטנדרטי.

נרצה לוודא שהתוצאה שהתקבלה עבור ציון ברייר הינה סבירה, כלומר שבהינתן ערכים זהים עבור המשתנים הבלתי תלויים אך תוצאות שונות עבור המשתנה התלוי, הציון שקיבלנו נמצא בטווח רווח סמך סביב הציון הממוצע שעשוי להתקבל. הסיבה שנרצה להסתכל על תוצאות שונות היא שהמודל מניב תחזית הסתברותית, ולעיתים עבור ערכים זהים למשתנים הבלתי תלויים יכולות להתקבל תוצאות שונות עבור המשתנה התלוי. מאחר ואנחנו רוצים לבדוק מצב שבו ערכי המשתנים הבלתי תלויים לא משתנים, אלא רק התוצאה, נוכל לבצע סימולציה עבור התוצאות באופן הבא:

1. נניח שהמודל מניב הסתברות שמייצגת היטב את המציאות וניצור נתונים פיקטיביים בהתאם להסתברות זו. כלומר אם עבור תצפית מסוימת המודל הניב הסתברות P לתוצאה, ניצור נתונים כך שיתפלגו ברנולית בהסתברות P (מה שיקרה במציאות בהנחה ש-P מייצג אותה היטב).

2. לאחר שהנתונים בידינו, נחשב את ציון ברייר עבורם. נוכל לחזור על תהליך זה פעמים רבות כך שלבסוף נקבל כמות גדולה של ציוני ברייר עבורם נוכל לחשב ממוצע וסטיית תקן, לבחור רווח סמך של 95% תחת הנחת התפלגות נורמלית (2 - + סטיות תקן) ולוודא שציון הברייר של התוצאה האמיתית נמצא בטווח רווח הסמך.

עתה, משיש בידינו ציון ממוצע, סטיית תקן ואת הציון של המודל ביחס לנתוני המבחן ישנן 2 אפשרויות:

1. הציון המקורי נמצא מחוץ לרווח הסמך. במקרה זה נסיק כי הפרמטרים שהתקבלו אינם טובים.

2. הציון המקורי נמצא בתוך רווח הסמך. במקרה זה נקבל ביטחון רב יותר לגבי הפרמטרים שהתקבלו (יש לשים לב שמצב זה אינו מבטיח פרמטרים טובים).

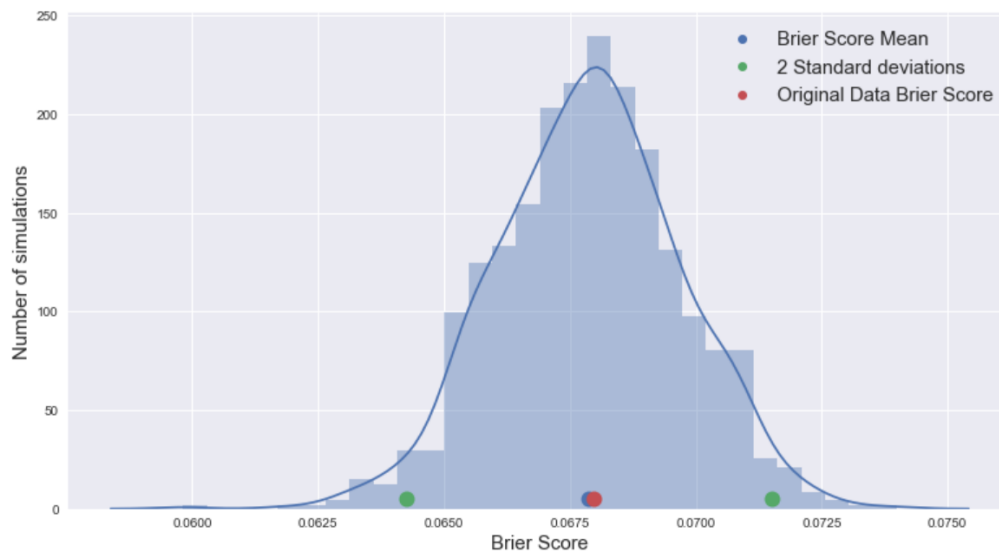
לאחר הרצת הסימולציות התקבלו התוצאות הבאות (ניספח פרק 14):

ממוצע: 0.06787

סטיית תקן: 0.00181

רווח סמך: (0.06424, 0.07151)

להלן גרף התפלגות ציוני ברייר של הסימולציות.



איור 11: "גרף ההתפלגות ציוני ברייר"

ניתן לראות שהציונים מתפלגים נורמלית ושהציון שהתקבל מהנתונים האמיתיים אכן נמצא בתוך רווח הסמך. לפיכך, אנחנו יכולים להסיק שקיבלנו את הציון האופטימלי האפשרי מהמודל, וקיבלנו ביטחון נוסף לגבי איכות המודל.

5.4 מסקנה

מהניתוח של המודל ראינו כי ההסתברויות שמתקבלות מהמודל, טובות, ויכולות לספק תחזיות מוצלחות לאיחור או הגעה של טיסה. כמובן, שיש דרכים נוספות לשפר המודל, לדוגמה, להוסיף משתנים חדשים, להוסיף משתנים מדרגה גבוהה יותר (משום שהמשתנה הבלתי תלוי יכול להתנהג בצורה לא לינארית) או לנסות ולזהות אילו משתנים דווקא פוגעים בביצועי המודל בעזרת מבחנים נוספים, ולהיפטר מהם. אך כבר עתה, באמצעות הדוגמה הפשוטה שהוצגה, ניתן לראות כי אכן הרגרסיה הלוגיסטית מתאימה לפתרון בעיית חיזוי עיכובי הטיסות.

קשיים טכניים:

במהלך העבודה על מימוש המודל, נתקלתי לא פעם בקשיים טכניים שנאלצתי להתגבר עליהם, להלן כמה מהם:

1. איחסון נתוני הטיסות- המאגר הראשוני כלל מידע על כ-5 מיליון טיסות. כאשר ניסיתי לבצע פעולות מסויימות על כל הנתונים, נתקלתי בגבולות הזיכרון של המחשב שלי, דבר שלעיתים מנע ממני לחלוטין לבצע את הפעולות המבוקשות ובמקרים אחרים פשוט האט את העבודה. בעיה זו נפתרה כשבחרתי להתמקד בשדה תעופה מוצא יחיד.

2. האתגרים התכנותיים שאיתם התמודדתי היו שונים מאילו שנתקלתי בהם במהלך הלימודים, לדוגמה נאלצתי להכיר המון תוכנות וספריות חדשות שבהן נעזרתי, ולקרוא את התייעוד עליהן לעומק.

3. גיליתי באמצע העבודה שחלק מנתוני המאגר היו "מלוכלכים" כלומר, לא בפורמט שציפיתי לו או לא קיימים בכלל. לפיכך, נאלצתי להשלים את המידע ממאגרים נוספים.

6 סיכום

במסגרת עבודה זו הצגתי את מודל הרגרסיה הלוגיסטית וסקרתי את מרכיביו - פונקציית הלוגית, הפונקציה הלוגיסטית ושיטת הנראות המקסימלית שלגביה הרחבתי את הדיון והצגתי את פונקציית הנראות, אומדן הנראות ואת הקשר של השיטה לרגרסיה לוגיסטית. בנוסף הצגתי את מדד המיומנות של ברייר לבחינת הצלחת המודל. ולבסוף הראתי איך כל החלקים מתחברים בדוגמת מודל החיזוי לעיכובי טיסות.

אני מקווה שהצלחתי להשיג את מטרתי להנגיש את הנושא לכל בוגר עם רקע מתמטי ברמה שיוכל להשתמש לצרכיו ברגרסיה הלוגיסטית.

מפאת היקף העבודה, ועל מנת לשמור על מיקוד, לא הצגתי כאן נושאים מתקדמים נוספים כמו רגרסיה לוגיסטית שבה ערך המשתנה התלוי הוא אחד מבין יותר מ-2 אפשרויות, מבחנים לרלוונטיות המשתנים הבלתי תלויים ושיטות נוספות לבדיקת הצלחת המודל. למתעניינים בנושא אני ממליצה בחום להרחיב אופקים לכיוונים אלו שכן הבסיס להם כבר הונח במסגרת עבודה זו.

עוד חשוב לי לציין שהחלק האהוב עליי בעבודה היה השילוב בין נושאים רבים שלמדתי בקורסים שונים במהלך התואר כגון אופטימיזציה לא לינארית, תורת ההסתברות, סטטיסטיקה, חשבון אינפיניטסימלי ועוד. בפרט נהניתי מהשימוש גם בכלים המתמטיים וגם בכלים התכנותיים. לדעתי השימוש בכלים אלו משקף בצורה טובה ביותר את עבודתו של מתמטיקאי שימושי בתקופתנו ואף מהווה תנאי מקדים לקבלה למשרות רבות בתחום.

אחד הקשיים שעלו במהלך העבודה היה הצורך לנסח בדיוקנות את הדברים, אך אני מרגישה שהצלחתי להתגבר עליו ואני מרוצה מהתוצאה הסופית.

לסיום, ארצה להודות לכל סגל המרצים שמהם למדתי במהלך התואר ולהקדיש תודה מיוחדת למנחה שליווה אותי לאורך כתיבת העבודה - פרופ"ח חגי כתריאל, תודה על השעות הארוכות שהקדשת ללמד ולכוון אותי, אני יודעת שללא עזרתך לא הייתי כל כך מרוצה מהתוצר הסופי.

ביבליוגרפיה 7**References**

- [1] Beth Ebert et al. Forecast verification-issues, methods and faq. *World Weather Research Programme Joint Working Group on Verification*, 2006.
- [2] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.
- [3] Hubertus Th Jongen, Klaus Meer, and Eberhard Triesch. *Optimization theory*. Springer Science & Business Media, 2007.
- [4] David G Kleinbaum and Mitchel Klein. Logistic regresion for correlated data: Gee. *Logistic Regression: A Self-Learning Text*, pages 327–375, 2002.
- [5] Wikipedia contributors. Logistic regression — Wikipedia, the free encyclopedia. [Online; accessed 2-July-2018].

1 Import dependencies

```
In [1]: %matplotlib inline

import time
import datetime
from datetime import datetime
from datetime import timedelta
import pandas as pd
import numpy as np
import seaborn as sns
import random
import matplotlib.pyplot as plt
import math
from collections import Counter
```

2 Load flights.csv data

```
In [2]: fields = ['MONTH', 'DAY', 'DAY_OF_WEEK', 'AIRLINE', 'ORIGIN_AIRPORT', '
DESTINATION_AIRPORT', 'SCHEDULED_DEPARTURE', 'ARRIVAL_DELAY', 'CANCELLED', 'DIVERTED']

flights_data = pd.read_csv('flights.csv', skipinitialspace=True, usecols=fields)

In [3]: len(flights_data)

Out[3]: 5819079

In [4]: flights_data['ORIGIN_AIRPORT'] = flights_data['ORIGIN_AIRPORT'].astype(str)
flights_data['DESTINATION_AIRPORT'] = flights_data['DESTINATION_AIRPORT'].astype(str)

In [5]: flights_data = flights_data.fillna(value = 0)

In [6]: #remove cancelled/diverted flights rows
flights_data = flights_data[flights_data['CANCELLED'] == 0 ]
flights_data = flights_data[flights_data['DIVERTED'] == 0 ]

In [7]: flights_data = flights_data.drop(['CANCELLED', 'DIVERTED'], axis = 1)
```

3 Correcting airport names

```
In [8]: fields = ['ORIGIN_AIRPORT_ID', 'ORIGIN']

df = pd.read_csv('dict.csv', skipinitialspace=True, usecols=fields)
df.head()

Out[8]:
```

	ORIGIN_AIRPORT_ID	ORIGIN
0	11292	DEN
1	14027	PBI

2	15356	TTN
3	14492	RDU
4	15356	TTN

```
In [9]: df = df.drop_duplicates()
```

```
In [10]: df_dict = df.set_index('ORIGIN_AIRPORT_ID').to_dict()['ORIGIN']
```

```
In [11]: def is_number(s):
    try:
        int(s)
        return True
    except ValueError:
        return False

    for i in flights_data['ORIGIN_AIRPORT'].values:
        if is_number(i):
            flights_data['ORIGIN_AIRPORT'].replace(i , df_dict.get(int(i)),
            inplace = True)
```

```
In [12]: def is_number(s):
    try:
        int(s)
        return True
    except ValueError:
        return False

    for i in flights_data['DESTINATION_AIRPORT'].values:
        if is_number(i):
            flights_data['DESTINATION_AIRPORT'].replace(i , df_dict.get(int(i)),
            inplace = True)
```

```
In [13]: #add target column to data
flights_data['delayed'] = flights_data['ARRIVAL_DELAY']>15
flights_data['delayed'] = flights_data['delayed'].astype(int)

flights_data = flights_data.drop(['ARRIVAL_DELAY'], axis = 1)
```

4 Select an origin airport

```
In [14]: flights_data['ORIGIN_AIRPORT'].value_counts()
```

```
Out[14]: ATL    376054
ORD    304135
DFW    253311
DEN    211479
LAX    209638
SFO    159592
```



```

PHX      158436
IAH      157110
...
CEC       173
UST       158
MMH       140
PPG       115
ADK        97
ILG        95
HYA        82
STC        77
DLG        77
GST        76
AKN        63
ITH        30
Name: ORIGIN_AIRPORT, Length: 322, dtype: int64

```

```

In [15]: atl_flights_data = flights_data[flights_data['ORIGIN_AIRPORT'] == 'ATL' ]
count=atl_flights_data['delayed'].value_counts()
non_delayed=count[0]
delayed=count[1]
print(delayed/(non_delayed+delayed))

```

0.151914352726

```

In [16]: ord_flights_data = flights_data[flights_data['ORIGIN_AIRPORT'] == 'ORD' ]
count=ord_flights_data['delayed'].value_counts()
non_delayed=count[0]
delayed=count[1]
print(delayed/(non_delayed+delayed))

```

0.224025514985

```

In [17]: dfw_flights_data = flights_data[flights_data['ORIGIN_AIRPORT'] == 'DFW' ]
count=dfw_flights_data['delayed'].value_counts()
non_delayed=count[0]
delayed=count[1]
print(delayed/(non_delayed+delayed))

```

0.202644180474

```

In [18]: ord_flights_data = ord_flights_data.drop(['ORIGIN_AIRPORT'],axis = 1)

```

```

In [19]: ord_flights_data.reset_index(drop = True , inplace = True)

```

5 Scheduled departure time to intervals

```
In [20]: ord_flights_data['SCHEDULED_DEPARTURE'] = ord_flights_data['SCHEDULED_DEPARTURE'].
replace(2400,0);
```

```
In [21]: ord_flights_data['SCHEDULED_DEPARTURE'] = ord_flights_data['SCHEDULED_DEPARTURE'].
astype(str)
s = ord_flights_data['SCHEDULED_DEPARTURE'].str.zfill(4)
for index, row in s.iteritems():
    ord_flights_data.set_value(index, 'SCHEDULED_DEPARTURE_intervals',int(((
    datetime.strptime(row,'%H%M')).hour)))
```

```
In [22]: ord_flights_data = ord_flights_data.drop(['SCHEDULED_DEPARTURE'], axis = 1)
```

6 Creating load variable

```
In [23]: match_by_1 = ['MONTH','DAY','DAY_OF_WEEK','SCHEDULED_DEPARTURE_intervals']
groups_1 = ord_flights_data.groupby(match_by_1).groups
```

```
In [24]: group_list = list(groups_1.items())
```

```
In [25]: ord_flights_data['LOAD'] = 0
```

```
In [26]: for group in group_list:
    g_len=len(group[1])
    ord_flights_data['LOAD'].update(pd.Series([g_len], index=group[1]))
```

7 Data visualization

```
In [96]: sns.countplot(x='delayed',data = ord_flights_data,palette='hls')
plt.title("ORD Airport")
plt.show()
plt.savefig('1234')
```

```
In [57]: delay_reasons = ['MONTH', 'DAY_OF_WEEK','SCHEDULED_DEPARTURE_intervals'];
for reason in delay_reasons:
    table=pd.crosstab(ord_flights_data[reason],ord_flights_data['delayed'])
    table.div(table.sum(1).astype(float), axis=0).plot(kind='bar',
    figsize = (12,5), fontsize=15, stacked=True,)
    plt.title('Stacked Bar Chart of ' + reason + ' vs Delay',fontsize=15)
    plt.xlabel(reason,fontsize=15)
    plt.ylabel('Proportion of delayed flights',fontsize=15)
    plt.savefig(reason+'_vs_delay')
```

```
In [58]: delay_reasons = ['DESTINATION_AIRPORT', 'AIRLINE'];
for reason in delay_reasons:
    table=pd.crosstab(ord_flights_data[reason],ord_flights_data['delayed'])
    table.div(table.sum(1).astype(float), axis=0).sort_values(0).plot(kind='bar',
```

```
figsize = (12,5), fontsize=15, stacked=True,)
plt.title('Stacked Bar Chart of ' + reason + ' vs Delay',fontsize=15)
plt.xlabel(reason,fontsize=15)
plt.ylabel('Proportion of delayed flights',fontsize=15)
plt.savefig(reason+'_vs_delay')
```

```
In [37]: table=pd.crosstab(ord_flights_data['LOAD'],ord_flights_data['delayed'])
table.div(table.sum(1), axis=0).plot(kind='line', figsize = (15,7), fontsize
=15, stacked=True)
plt.title('Stacked Bar Chart of LOAD' + ' vs Delay',fontsize=15)
plt.xlabel('LOAD',fontsize=15)
plt.ylabel('Proportion of delayed flights',fontsize=15)
plt.savefig('LOAD_vs_delay')
```

8 Create dummies variables

```
In [38]: ord_flights_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 303998 entries, 0 to 303997
Data columns (total 8 columns):
MONTH                303998 non-null int64
DAY                  303998 non-null int64
DAY_OF_WEEK          303998 non-null int64
AIRLINE               303998 non-null object
DESTINATION_AIRPORT   303998 non-null object
delayed               303998 non-null int32
SCHEDULED_DEPARTURE_intervals  303998 non-null float64
LOAD                  303998 non-null int64
dtypes: float64(1), int32(1), int64(4), object(2)
memory usage: 17.4+ MB
```

```
In [39]: ord_flights_data['DAY_OF_WEEK'] = ord_flights_data['DAY_OF_WEEK'].astype(str)
ord_flights_data['MONTH'] = ord_flights_data['MONTH'].astype(str)
ord_flights_data['SCHEDULED_DEPARTURE_intervals'] = ord_flights_data['SCHEDULED_DEPARTURE_intervals'].astype(str)
```

```
In [40]: ord_flights_data_dummies = pd.get_dummies(ord_flights_data , drop_first = True)
```

9 building training and testing data

```
In [41]: match_by = [x for x in ord_flights_data_dummies.columns if x!='delayed']
```

```
In [42]: from random import shuffle
```

```
#Build groups of identical data
groups = ord_flights_data_dummies.groupby(match_by).groups
```

```
print(len(groups))
groups_list = list(groups.items())

#shuffle the groups
shuffle(groups_list)
```

299827

In [43]: *#building training and testing data*

```
training_index = pd.Index([])
test_index = pd.Index([])

i=0
while len(training_index)<219486 212894:
    training_index = training_index.union(groups_list[i][1])
    i+=1

while len(training_index)+len(test_index)<len(ord_flights_data):
    test_index = test_index.union(groups_list[i][1])
    i+=1

print(training_index)
print(test_index)
print(len(training_index))
print(len(test_index))
```

```
Int64Index([    0,     2,     4,     5,     6,     8,    10,    11,
            12,    14,
            ...
            303983, 303985, 303986, 303988, 303989, 303990, 303992, 303994,
            303995, 303997],
            dtype='int64', length=219486)
Int64Index([     1,     3,     7,     9,    13,    15,    16,    18,
            19,    20,
            ...
            303966, 303970, 303972, 303974, 303977, 303984, 303987, 303991,
            303993, 303996],
            dtype='int64', length=84512)
```

219486

84512

```
In [44]: X_train = ord_flights_data_dummies.iloc[training_index]
         X_train = X_train[match_by]

         y_train = ord_flights_data_dummies.iloc[training_index]
         y_train = y_train['delayed']
```

```
X_test = ord_flights_data_dummies.iloc[test_index]
X_test = X_test[match_by]

y_test = ord_flights_data_dummies.iloc[test_index]
y_test = y_test['delayed']
```

```
In [46]: print (X_train.shape, y_train.shape)
         print (X_test.shape, y_test.shape)
```

```
(219486, 206) (219486,)
(84512, 206) (84512,)
```

```
In [47]: X_train.reset_index(drop = True , inplace = True)
         y_train.reset_index(drop = True , inplace = True)
         X_test.reset_index(drop = True , inplace = True)
         y_test.reset_index(drop = True , inplace = True)
```

10 fit a model

```
In [53]: from sklearn import linear_model
```

```
lm = linear_model.LogisticRegression()
model = lm.fit(X_train, y_train)
```

```
In [54]: model.intercept_
```

```
Out[54]: array([ 0.29935328])
```

```
In [55]: pd.DataFrame(np.transpose(model.coef_),X_test.columns,['Weights'])
```

```
Out[55]:
```

	Weights
LOAD	-0.030350
MONTH_10	-0.948988
MONTH_11	-0.544356
MONTH_12	-0.455869
MONTH_2	0.273582
MONTH_3	-0.392045
MONTH_4	-0.635430
MONTH_5	-0.512231
MONTH_6	0.286330
MONTH_7	-0.109233
MONTH_8	-0.334322
MONTH_9	-0.708136
DAY_OF_WEEK_2	-0.242531
DAY_OF_WEEK_3	-0.165971

DAY_OF_WEEK_4	-0.135024
DAY_OF_WEEK_5	-0.140418
DAY_OF_WEEK_6	-0.817044
DAY_OF_WEEK_7	-0.255898
AIRLINE_AS	-0.686307
AIRLINE_B6	0.104083
AIRLINE_DL	-0.041496
AIRLINE_EV	0.176824
AIRLINE_F9	0.758465
AIRLINE_MQ	0.207363
AIRLINE_NK	0.907488
AIRLINE_OO	0.527945
AIRLINE_UA	0.272611
AIRLINE_US	-0.229309
AIRLINE_VX	-0.195925
DESTINATION_AIRPORT_ABQ	0.373958
...	...
DESTINATION_AIRPORT_STT	0.039692
DESTINATION_AIRPORT_SUX	-0.182211
DESTINATION_AIRPORT_SYR	-0.039301
DESTINATION_AIRPORT_TOL	-0.485962
DESTINATION_AIRPORT_TPA	0.109805
DESTINATION_AIRPORT_TTN	-0.304041
DESTINATION_AIRPORT_TUL	0.022627
DESTINATION_AIRPORT_TUS	0.265624
DESTINATION_AIRPORT_TVC	-0.252156
DESTINATION_AIRPORT_TYS	0.091546
DESTINATION_AIRPORT_XNA	0.084697
SCHEDULED_DEPARTURE_intervals_10.0	-0.032407
SCHEDULED_DEPARTURE_intervals_11.0	-0.446376
SCHEDULED_DEPARTURE_intervals_12.0	0.408743
SCHEDULED_DEPARTURE_intervals_13.0	0.837839
SCHEDULED_DEPARTURE_intervals_14.0	0.153291
SCHEDULED_DEPARTURE_intervals_15.0	0.526643
SCHEDULED_DEPARTURE_intervals_16.0	0.289048
SCHEDULED_DEPARTURE_intervals_17.0	0.867287
SCHEDULED_DEPARTURE_intervals_18.0	0.634737
SCHEDULED_DEPARTURE_intervals_19.0	0.775422
SCHEDULED_DEPARTURE_intervals_20.0	0.765426
SCHEDULED_DEPARTURE_intervals_21.0	-0.034091
SCHEDULED_DEPARTURE_intervals_22.0	-0.939922
SCHEDULED_DEPARTURE_intervals_23.0	-0.863806
SCHEDULED_DEPARTURE_intervals_5.0	-2.131205
SCHEDULED_DEPARTURE_intervals_6.0	-1.027011
SCHEDULED_DEPARTURE_intervals_7.0	-0.030174
SCHEDULED_DEPARTURE_intervals_8.0	0.214807
SCHEDULED_DEPARTURE_intervals_9.0	0.471456

```
[205 rows x 1 columns]
```

```
In [56]: predict_prob_test=model.predict_proba(X_test)
         pred = pd.DataFrame(np.transpose(predict_prob_test[:,1]),X_test.
         index,['Prediction'])
```

```
In [57]: pred['y_test']=pd.Series(y_test.values, index=y_test.index)
```

```
In [80]: pred
```

```
Out[80]:
```

	Prediction	y_test
0	0.166707	0
1	0.172476	0
2	0.256154	0
3	0.322303	0
4	0.361721	0
5	0.572008	0
6	0.551658	0
7	0.544394	0
8	0.418098	0
9	0.358458	0
10	0.692731	0
11	0.612442	1
12	0.561950	0
13	0.456867	0
14	0.415802	0
15	0.446620	1
...	...	...
84493	0.577045	0
84494	0.457530	0
84495	0.485433	0
84496	0.389998	0
84497	0.602105	1
84498	0.339883	0
84499	0.277437	0
84500	0.290092	0
84501	0.329418	0
84502	0.295075	0
84503	0.304832	0
84504	0.367269	0
84505	0.448226	1
84506	0.376400	0
84507	0.339883	0
84508	0.312604	0
84509	0.624247	1
84510	0.663855	0
84511	0.239552	0

```
[84512 rows x 2 columns]
```

11 Brier Score

In [58]: *#Percentage of flight delays*

```
count=y_test.value_counts()
non_delayed=count[0]
delayed=count[1]
Percentage_test_delays = delayed/(non_delayed+delayed)
Percentage_test_delays
```

Out[58]: 0.2245006626277925

In [59]: *#Predictions of test data*

```
predict_prob_test=model.predict_proba(X_test)
predict_prob_test[:,1]
```

Out[59]: array([0.16670719, 0.17247647, 0.25615358, ..., 0.62424727,
 0.66385471, 0.23955197])

In [60]: *from sklearn.metrics import brier_score_loss*

```
print(brier_score_loss(y_test, predict_prob_test[:,1]))

predict_test_stand = np.ones(y_test.shape[0])*Percentage_test_delays
print(brier_score_loss(y_test, predict_test_stand))
```

0.162266366894

0.174100115107

```
In [61]: braier_score_skill_test=1-(brier_score_loss(y_test, predict_prob_test[:,1])
/brier_score_loss(y_test, predict_test_stand))
braier_score_skill_test
```

Out[61]: 0.067970938479554155

12 Simulating data

In [66]: *y_train_simulated = pd.Series(np.random.binomial(1, predict_prob_train[:,1]),
index=y_train.index)*

```
y_test_simulated = pd.Series(np.random.binomial(1, predict_prob_test[:,1]),
index=y_test.index)
```

13 Range of Braier Score Skill for simulated data

In [67]: *braier_score_skill_test_range = []*


```

for i in range(1000):
    y_test_simulated = pd.Series(np.random.binomial(1, predict_prob_test[:,1]),
                                index=y_test.index)

    count=y_test_simulated.value_counts()
    non_delayed=count[0]
    delayed=count[1]
    Percentage_test_delays_simulat = delayed/(non_delayed+delayed)

    predict_test_stand_simulate = np.ones(y_test.shape[0])
    *Percentage_test_delays_simulat
    braier_score_skill_test_simulate = 1-(brier_score_loss(y_test_simulated,
    predict_prob_test[:,1])/brier_score_loss(y_test_simulated
    , predict_test_stand_simulate))
    braier_score_skill_test_range.append(braier_score_skill_test_simulate)

```

14 Analysis of simulations

```

In [83]: braier_scores_array = np.asarray(braier_score_skill_test_range)
         braier_scores_array.mean()

```

```

Out[83]: 0.067876651909922481

```

```

In [84]: braier_scores_array.std()

```

```

Out[84]: 0.0018168088592089404

```

```

In [106]: revah_samah_smol = braier_scores_array.mean()-2*braier_scores_array.std()
          revah_samah_yamin = braier_scores_array.mean()+2*braier_scores_array.std()

```

```

In [182]: print(revah_samah_smol,revah_samah_yamin)

```

```

0.0642430341915 0.0715102696283

```

```

In [186]: import seaborn as sns
          import matplotlib.pyplot as plt
          sns.set(color_codes=True)

          mean = braier_scores_array.mean()
          plt.figure(figsize=(13,7))

          sns.distplot(braier_scores_array)

          plt.plot(mean, 5, "bo", label="Brier Score Mean", markersize=12)
          plt.plot(revah_samah_smol, 5, "go", label="2 Standard deviations", markersize=12)
          plt.plot(revah_samah_yamin, 5, "go", markersize=12)
          plt.plot(braier_score_skill_test, 5, "ro", label="Original Data Brier Score",

```

```
markersize=12)

plt.legend(handles=[a,b,d],fontsize=15)
plt.xlabel('Brier Score',fontsize=15)
plt.ylabel('Number of simulations',fontsize=15)
plt.show()
print(braier_scores_array.mean())
```

0.0678766519099

```
In [69]: if braier_score_skill_test >= min_braier_score_skill_test and
braier_score_skill_test <= max_braier_score_skill_test:
    print('in range')
else:
    print('not in range')
```

in range