**משימה ד – השוואת המחיר וחיזוי של הפרש המחירים**

**באופן כללי על ההבדל בין האתרים:**

מגמות כלליות והשפעת גורמים שונים על המחיר (בחלק השני של המחברת)

* Momondo יקרה יותר מ-Kayak במרבית המקרים, עם שינויים חדים יותר במחירים.
* Kayak מציגה מחירים יציבים יותר עם פחות שינויים קיצוניים.
* פערי מחירים משמעותיים יותר ב-Momondo בסופי שבוע, בשעות עומס ובטיסות עם עצירות ביניים מרובות.
* משך השהייה (LOS) משפיע בצורה שונה על האתרים – ב-Kayak המחיר יורד מעט ככל שהשהייה מתארכת, בעוד שב-Momondo המחיר דווקא עולה.
* לשעת ההמראה והחזרה יש השפעה על ההבדלים במחירים – Momondo מציגה פערים גבוהים יותר בבוקר ובערב, בעוד שב-Kayak הפערים קטנים יותר לאורך היום.
* מרחק שדה התעופה בהמראה ובחזרה משפיע על ההבדל במחירים – ב-Momondo המחירים עולים בצורה חדה יותר ככל שהמרחק גדל, לעומת שינוי מתון יותר ב-Kayak.
* יום ההמראה והחזרה משפיע באופן שונה בכל אתר – הפערים בין המחירים גבוהים יותר ב-Momondo בימי סוף השבוע, בעוד שב-Kayak ההפרשים מתונים יותר.
* מספר עצירות ביניים משפיע בעיקר על Momondo – ככל שיש יותר עצירות, המחירים עולים משמעותית יותר לעומת השפעה מתונה יותר ב-Kayak.

במשימה זו התבקשנו לבנות מודל החוזה את הפרש המחירים בין האתרים. צריך להשתמש באלגוריתמים שונים ופרמטרים שונים כדי לנסות לחזות הכי טוב שאפשר. הבעיה העיקרית היא שאין דרך ישירה להשוות טיסות ספציפיות בין האתרים, כיוון שאין זיהוי ייחודי לכל טיסה. לכן, אנחנו צריכים למצוא דרך שתאפשר חישוב עקיף של הפרשי המחירים.

אפשרויות שחשבנו עליהן:

1. התאמת טיסות לפי מאפיינים דומים - מכיוון שאין מזהה ייחודי, ניתן להתאים טיסות בין האתרים על סמך מאפיינים משותפים (מסלול, חברת תעופה ועוד). אחרי ההתאמה של טיסות דומות בין האתרים, ניתן לחשב את הפרש המחירים.
2. גישת Clustering - כאן מחלקים את כל הטיסות לקבוצות דומות באמצעות אלגוריתם קיבוץ. לאחר מכן, בתוך כל אשכול משווים טיסות מאותו אשכול שמופיעות בשני האתרים. בגישה הזו, ההתאמה בין טיסות מבוססת על דמיון כללי ולא על התאמה אחד-לאחד.

החלטנו להשתמש בגישה הראשונה (כי בקלסטרינג כבר השתמשנו במחברת אחרת) כך:

בקוד הזה בוצע קיבוץ של הנתונים לפי משתנים עיקריים (מוצא, יעד, סוג מסע, משך שהייה, תאריך צילום והאתר), ולאחר מכן חישוב ערכים מאוגדים – עבור כל קומבינציה של TTT, LOS, סנאפשוט ומסלול (מקור + יעד) יש 100 טיסות, ובסך הכול קיימות 2,700 קומבינציות כאלה. על כל אחת מהן בוצעה אגרגציה מותאמת לפיצ'ר – חציון (median) עבור מחירים וזמנים (כדי למנוע השפעות של ערכים קיצוניים), שכיח (mode) עבור משתנים קטגוריים (שדות תעופה, חברות תעופה), וממוצע (mean) במקרים בהם זה הגיוני יותר. לאחר מכן, הנתונים פוצלו לשני דאטה-פריימים – אחד עבור Kayak ואחד עבור Momondo, ולבסוף חושב הפרש המחירים (Momondo - Kayak). בזכות התהליך הזה, ניתן כעת להשוות בצורה מדויקת יותר בין המחירים של שני האתרים, שכן ההשוואה מתבצעת ברמת קבוצות טיסות דומות במקום על בסיס רשומות בודדות. בהמשך ניתן להשתמש בערך מוחלט או לבצע חיזוי של הפערים על סמך מאפייני הטיסות.

לאחר הרצה ראשונית של אלגוריתמים שונים, התוצאות R2 היו כמעת אפסיות בכל האלגו השונים. הסקנו שזה מכיוון שיש לנו רק 70% מ- 2,700 נתונים ולכן אין מספיק מידע בשביל שילמד לחזות כמו שצריך (under-fitting).

לכן חילקנו כל קומבינציה של 100 רשומות לתתי קבוצות של 10 רשומות ולכל קבוצה כזו ביצענו אגרגציה וככה הכפלנו את מספר הרשומות פי 10. המדדים עלו בהתאם.

שקלנו האם כדאי לפצל כל מסלול ולבדוק לפי איך יצרנו בוקס-פלוט וראינו שחוץ מרעש, המסלולים מתפלגים בהפרש מחיר באופן דומה ולכן הוחלט להשאיר אותם יחד (ובנוסף חסכנו הרצות אלגוריתמים על פני 12 חלוקות שונות).

כעת עברנו לשלב ה – פיצ'ר סלקשן שבו השתמשנו ב 4 שיטות שונות כדי לפלטר את מפיצ'רים שאינם מועילים ליעילות המודל.

1. Variance Threshold – ככל שהנתונים של פיצ'ר מסווים יותר מגוונים, כך המודל יוכל ללמוד טוב יותר את ההבדלים המשפיעים על המטרה, לכן נבחר להוריד את אילו שבעלי שונות נמוכה.
2. Random Forrest Feature Importance – בעזרת מדד החשיבות המובנה של אלגוריתם הראנדום פורסט, היה נוח להבין אילו פיצ'רים הם אלו שמשפיעים הכי הרבה על המודל ובהתאם סיננו את אלו שלא.
3. Correlation With Target – המדד אולי הכי חשוב. ככל שהקורלציה של משתנה מסוים עם משתנה המטרה גבוה, כך ידע המודל להשתמש בו בשביל לחזות באופן מדויק יותר. בהתאם הורדנו את אלו שבעלי ערך נמוך.
4. K-Best Features – שיטה המריצה חזרות, כאשר בכל חזרה משתמשת ב K פיצ'רים בלבד, ובהתאם ניתן לדעת כמה ואילו מהפיצ'רים שלנו יניבו את התוצאות המדויקות ביותר.

* - היה גם ניתן להשתמש בשיטת RFE המוציאה כל פעם פיצ'ר בודד ובודקת את התוצאה אך הדבר דורש זמני ריצה ארוכים במיוחד ובנוסף הוא בודק את מדדיו בהתאם לאלגוריתם שהוא רץ עליו, בניגוד לשיטה שלנו שבודקת באופן כללי יותר. זה משנה כי אנחנו נריץ מספר אלגוריתמים שונים ולכן חשוב שיהיה כמה שיותר כללי בשלב זה.

לאחר מכן הרצנו את האלגו: XGBoost, Gradiant Boosting and Random Forrest בשל היותם המהירים ביותר וגם מכיוון שהשיגו את התוצאות הטובות ביותר בבדיקה הראשונית.

על כל אחד מהם בוצע gridSearch וקיבלנו ש Gradiant Boosting הוא האלגו הטוב ביותר לשני האתרים, אך עם תוצאות מאוד נמוכות:

|  |  |  |
| --- | --- | --- |
| Momondo | Kayak |  |
| 17752.06 | 12561.60 | MSE |
| 0.5179 | 0.6589 | R² |
| 99.73 | 82.79 | MAE |
| 133.24 | 112.08 | RMSE |

מסקנות:

✅ התכונות שלנו עשויות להסביר מגמות כלליות בתמחור טיסות, אך לא אסטרטגיות תמחור ייחודיות לכל אתר.

✅ הנחות, עמלות ותזמון עדכונים ספציפיים לכל אתר יוצרים הבדלים בלתי צפויים.

✅ בשל הקושי להעריך את ההפרשים במחירים בין האתרים, נבחן את ההבדלים הכלליים כדי לקבל רמז ברור יותר למגמות.

סיכום סופי: מכיוון שמודל החיזוי שלנו אינו מדויק כל כך, בעזרת ההשוואות מהמידע הקיים ניתן להניח ולהסיק המון מסקנות שונות על הבדלי המחירים בין האתרים השונים.