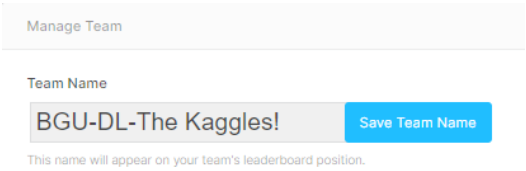
שאלה 3- תחרות Kaggle

Corporación Favorita Grocery Sales Forecasting

סעיף a- רישום לתחרות Kaggle

הצטרפנו לתחרות של Corporación Favorita Grocery Sales Forecasting - Kaggle באשר מטרת התחרות היא לחזות לכל צירוף של חנות ומוצר את ערכי המחירות של אותו מוצר בכל חנות במשך השבועיים האחרונים של חודש אוגוסט 2017 (15/8/17-31/8/17).

הקבוצה שלנו נרשמה לתחרות תחת השם: BGU-DL <The Kaggles!>

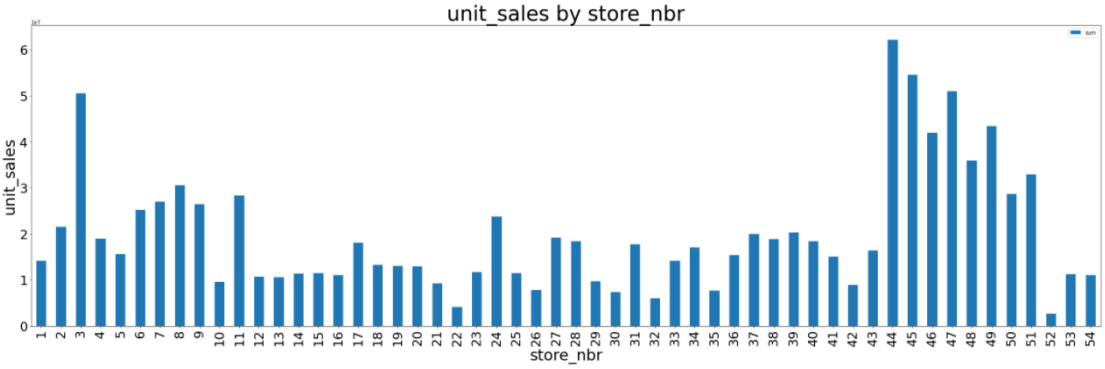


סעיף מקדים- ניתוח וגילוי ה-Data

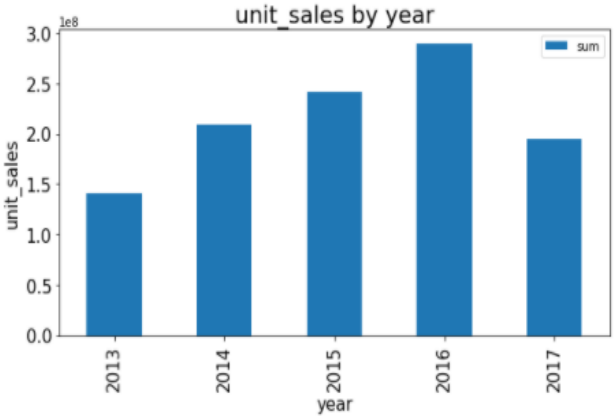
בחרנו להוסיף סעיף מקדים זה לשאלה מכיוון שקבצי הנתונים שקיבלנו עבור המשימה מכילים מידע שנאסף לאורך חמש שנים עבור חיזוי שבועיים האחרונים של אוגוסט 2017. מכיוון שמדובר בהמון Data אנו חושבים שאנו צריכים למקד את קובץ האימון שלנו לתקופות רלוונטיות עבור החיזוי המתבקש. בנוסף לכך עקב מגבלת ה-RAM במחשבים שלנו קשה לטעון ולאמן מודל על כל קבצי הנתונים וזה רק מחזק את ההחלטה שעלינו לצמצם את נתוני האימון בכך שנבחר תקופה מסויימת ובעזרתה נחזה את התקופה המתבקשת.

לשם כך טענו את כל ה-data פעם אחת, נעזרנו בפונקציה שמקטינה את גודל ה- dataframe שנטען ל-RAM וביצענו ניתוחים ויזואלים על הנתונים כדי להבין איך ה-data מתנהג בתקופות מסוימות והאם יש פיצ'רים שיותר משפיעים על צריכת המוצרים או כאלה שאפשר להתעלם מהם.

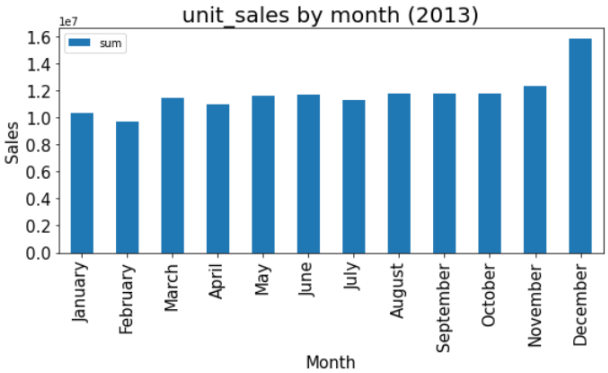
להלן התובנות

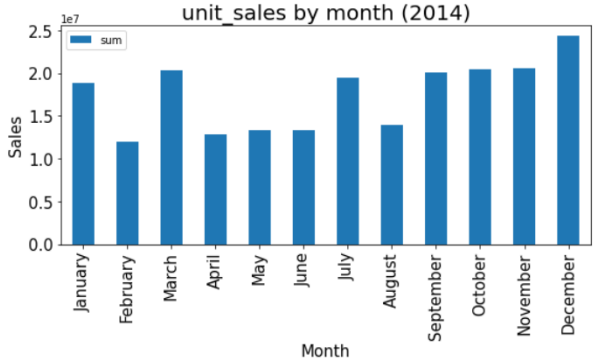


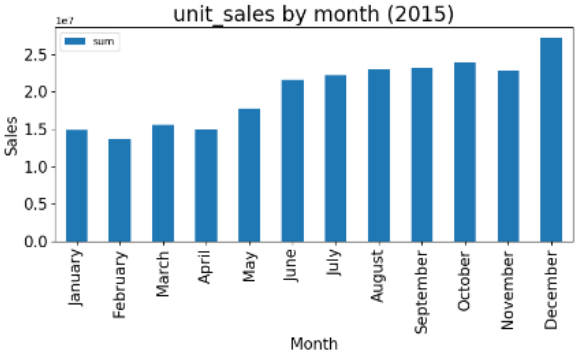
בגרף זה ניסינו לבדוק כמה יחדות מוצרים נקנו סה"כ בכל חנות במהלך השנים 2013-2017 כדי לקבל תמונה כללית על כמות היחידות הנמכרות בכל חנות. ניתן לראות כי החנות בה נקנו הכי הרבה מוצרים זו חנות 44 ואילו החנות בה נקנו הכי מעט מוצרים היא 52. סה"כ ישנן 54 חנויות ייחודיות בקובץ הנתונים.

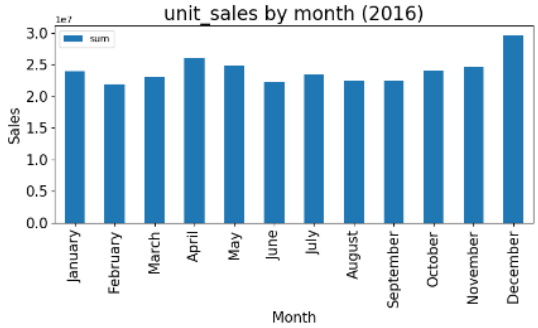


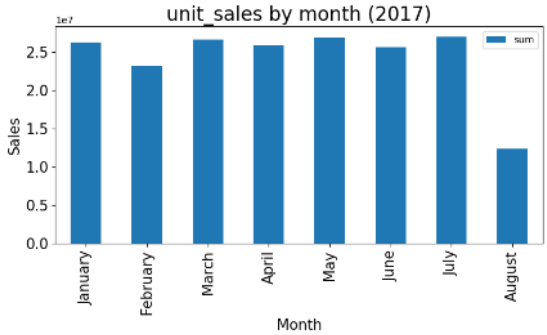
בגרף זה ניסינו לבדוק כמה יחידות מוצרים נקנו סה"כ בכל שנה שיש בקובץ הנתונים. ניתן לראות שבכל שנה הייתה עלייה בקניית המוצרים מלבד שנת 2017. העלייה הכי גדולה נראה בשנת 2016 וייתכן שזה קשור לרעידת האדמה שפקדה את אקוודור באותה שנה (צויין בהקדמה שניתנה על ה-Data). בנוסף ייתכן שהמכירות ב-2017 נראות נמכות מכיוון שעבור שנה זו יש נתונים רק עד אמצע חודש אוגוסט של אותה שנה.







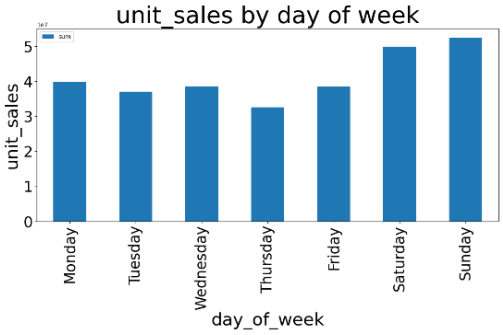




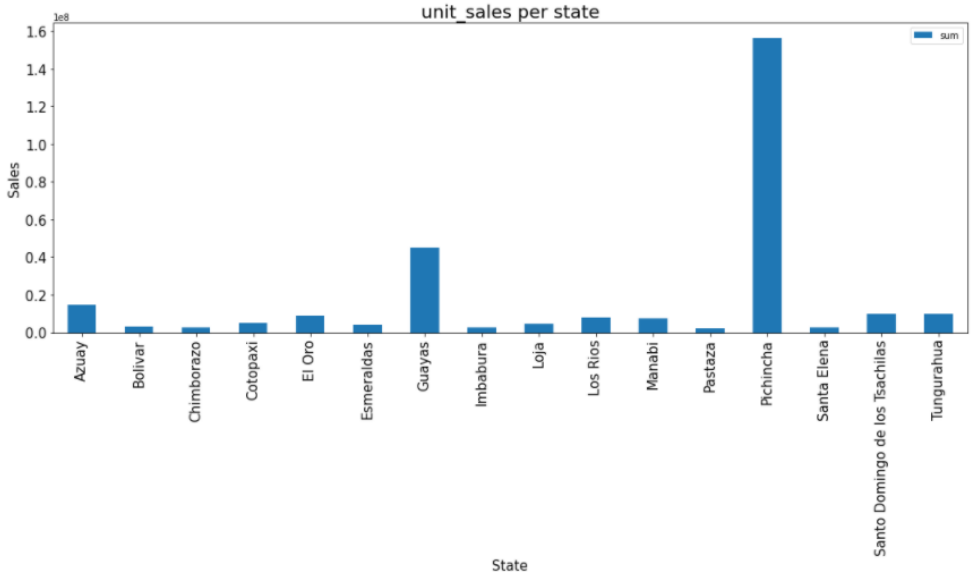
באמצעות הגרפים הנ"ל ניסינו לבדוק בכל שנה כמה יחידות נמכרות סה"כ בכל חודש במהלך השנה ולבדוק אם יש דפוס חוזר עבור חודש מסויים בכל שנה. ניתן לראות כי בחודש דצמבר בכל השנים (מלבד 2017 שאין נתונים על חודש זה ולכן הוא לא מופיע) מספר היחידות הנמכרות הוא הגבוה ביותר, נתון שבהחלט הגיוני לאור תקופת החגים והכריסמס שחלים בחודש זה. אפשר גם לראות שבחודש אוגוסט אותו אנו אמורים לחזות בשנים 2013-2014 היה מעט יותר נמוך ביחס לשנים הבאות אבל אין הבדל ניכר של עלייה חדה או ירידה בין השנים. סה"כ יש שוני בין כל חודש בכל שנה ומכאן אנו מסיקים שיש חשיבות לתקופה בה נעשו הקניות ובמידה ונבחר תקופה שהיא דומה לתקופה אותה אנו אמורים לחזות ייתכן ונקבל תוצאות טובות יותר.

בחרנו לקחת את שנת 2016 ולבצע עלייה ניתוחים נוסף בחתך שני כדי לרדת לרזולוציה ונמוכה יותר ולהבין איך המכירות מושפעות פר חודש ופר יום בחודש. בחרנו בשנה זו כי זוהי השנה האחרונה לפני השנה על אנו צריכים לבצע את החיזוי, היא מכילה את כל חודשי השנה בשונה משנת 2017 ובנוסף רצינו לבדוק אם באמת יש השפעה לרעידת האדמה שקרתה בשנה זו.

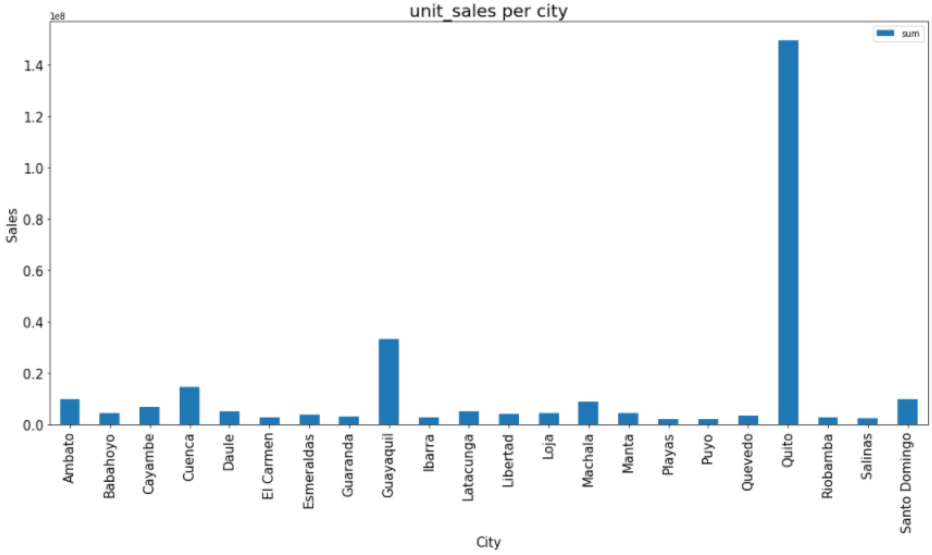
בגרפים הבאים עברנו על כל החודשים במהלך שנת 2016 וניסינו לבדוק עבור כל יום בחודש מה כמות היחידות הנמכרות ובכך לזהות אם יש השפעה ליום בחודש על המכירות. ניתן לראות שכמעט בכל החודשים ב-1 לחודש וב-31 לחודש יש עלייה ביחידות המכירות וגם באמצע החודש בין הימים 15-18 לחודש. ייתכן וזה קורה בגלל שהאזרחים באקוודור מקבלים משכורת בשתי פעימות פעם באמצע החודש ופעם בסוף החודש. בנוסף רצינו לראות את ההשפעה של רעידת האדמה שהתרחשה ב-16 באפריל 2016 ובאמת ניתן לראות שכמה ימים אחרי רעידת האדמה אכן יש עליה משמעותית ביחידות הנמכרות אבל לאחר שבוע המצב יחסית מתאזן. לאור השוני אותו אנו רואים בין הימים במהלך החודש אנו מבינים שייתכן ואם נתייחס לכך בעת אימון המודל ייתכן וזה עשוי לשפר את תוצאותיו.



בגרף הבא ניסינו לבדוק כמה יחידות נמכרות בשנת 2016 לפי היום בשבוע. ניתן לראות שבסופ"ש בימים שבת וראשון יש עלייה ביחידות הנמכרות לעומת שאר ימי השבוע. זה הגיוני לנוכח העובדה שבסופ"ש אנשים לרוב לא עובדים ויותר פנויים לבצע קניות. מכאן אנו מבינים שיש חשיבות ליום בשבוע והוא יוכל לנו לעזור כפיצ'ר באימון המודל.

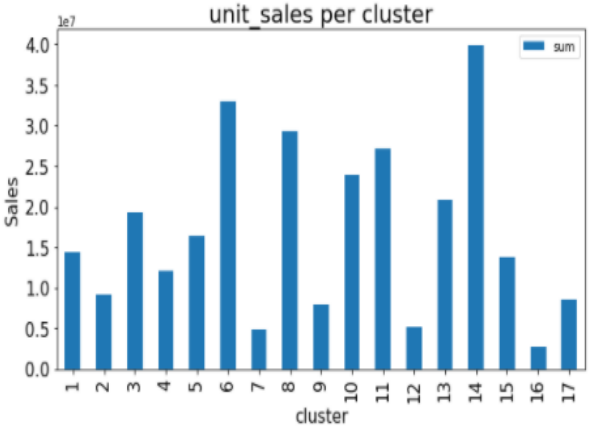


בגרף זה ניסנו להבין מהי כמות היחידות הנמכרות בכל מדינה בשנת 2016 והאם השימוש בפיצ'ר זה יכול לסייע לנו בתהליך האימון. מהגרף ניתן לראות כי המדינה בה נמכרים הכי הרבה יחידות היא Pichincha והמדינה עם הכי מעט מכירות היא Pastaza.

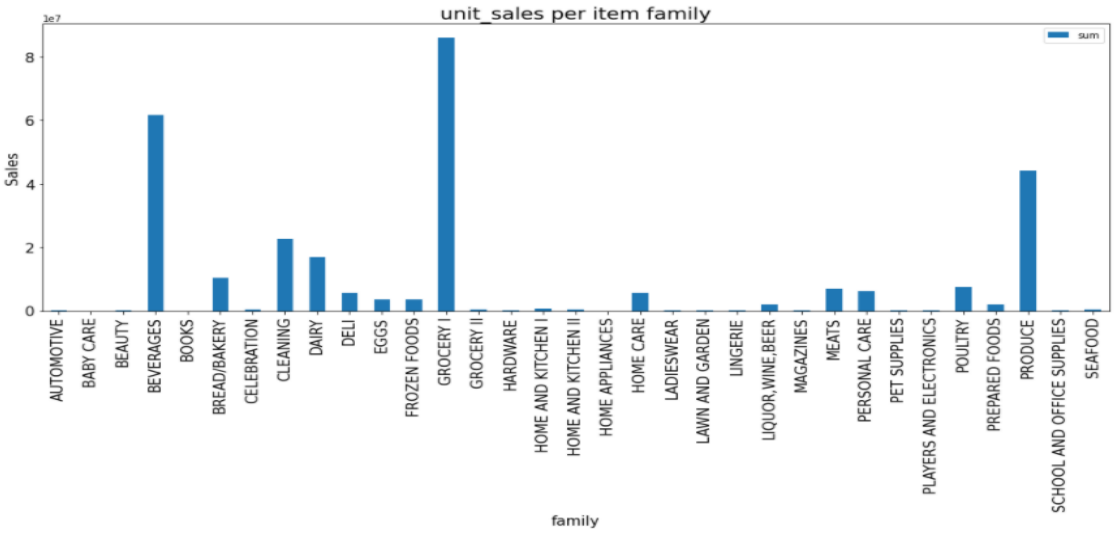


בגרף זה ניסינו את כמות היחידות הנמכרת בכל עיר בשנת 2016 ולנתח אם פיצ'ר זה יכול לסייע לנו בתהליך האימון. ניתן לראות שהעיר שבה נמכרים הכי הרבה יחידות היא קיטו, עיר הבירה של אקוודור. נתון זה מחזק את החשיבות של פיצ'ר זה ומראה שיש השפעה כל כמות היחידות הנמכרת ביחס לעיר הבירה בה נמצאת החנות.

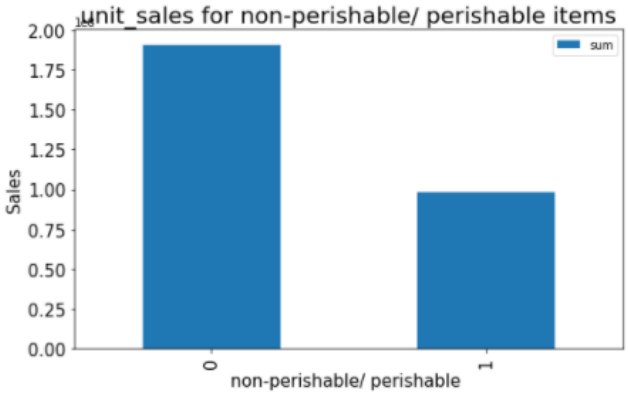
בגרף הבא ניסנו לנתח כמה יחידת מוצרים נקנו סה"כ בשנת 2016 לפי סוג החנות. ניתן לראות שסוג החנות שבה נקנו הכי הרבה יחידות היא D ואילו סוג החנו שנקנו בה הכי מעט מוצרים היא E. גם עבור סוגי החנויות האחרות נקנו לא מעט יחידות ויש שוני בכמות בין כל חנות ולכן ניתן להסיק שייתכן שיש השפעה על סוג החנות על קניית מוצרים, אבל ייתכן גם שפשוט יש מעט חנויות מסוגים שונים ולכן זה גם משפיע על כמויות המכירה.



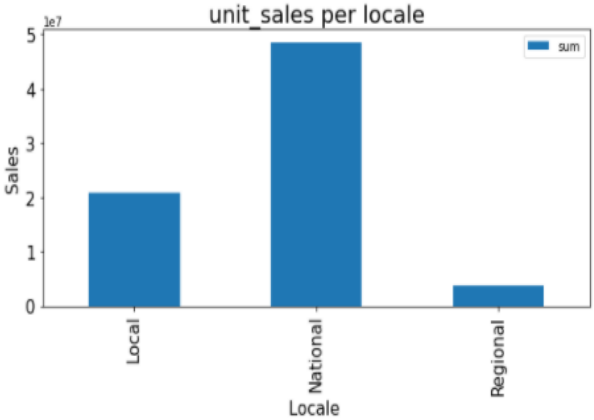
בגרף הבא ניסינו להבין כמה יחידות נמכרות בשנת 2016 לפי כל cluster שחנויות נמצאות בו. ניתן לראות שעבור cluster 14 יש הכי הרבה יחידות שנמכרות. בנוסף ניתן לראות כי ישנה התפלגות שונה של המכירות לפי ה-clusters וייתכן שגם פיצ'ר זה יכול לסייע בתהליך האימון. גם כאן נסתייג ונאמר כי ייתן שיש clusters שבהם יש מעט חנויות וכאלה שיש הרבה חנויות וזה מה שבעצם מעלה או מוריד את המכירות עבור כל cluster.



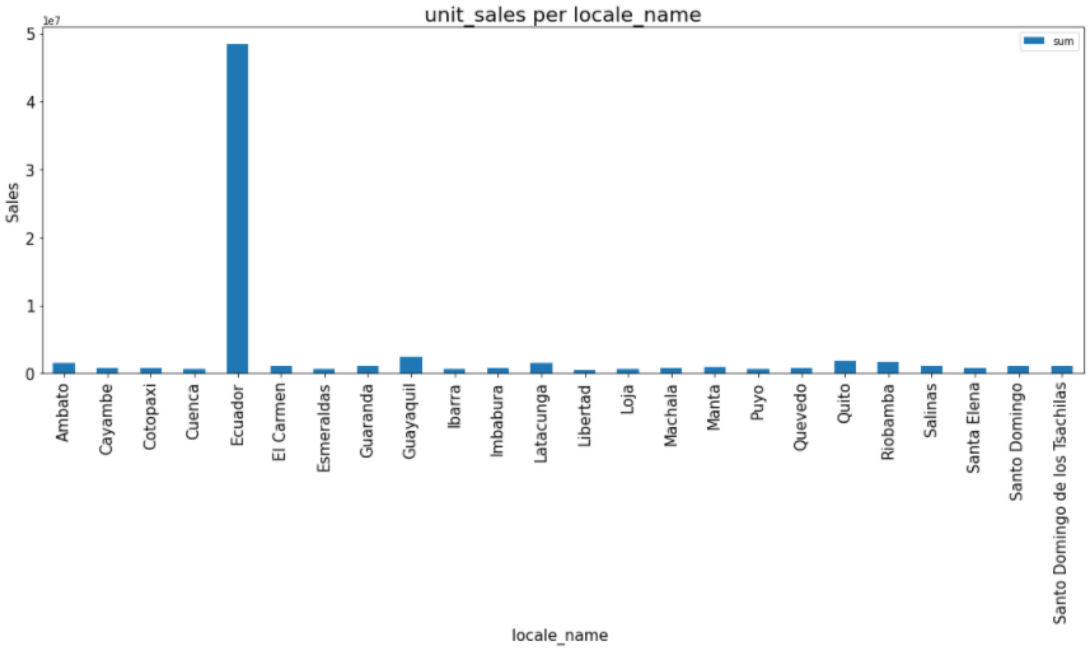
בגרף הבא ניסינו להבין כמות יחידות נמכרות בשנת 2016 לפי המשפחה אלייה שייך כל פריט. ניתן לראות שמשפחת המוצרים שעבורה נקנים הכי הרבה יחידות היא GROCERY I. ניתן גם לראות שיש שונות בין המכירות של כל משפחה, כלומר ייתכן שפיצ'ר זה יכול לסייע בתהליך האימון.



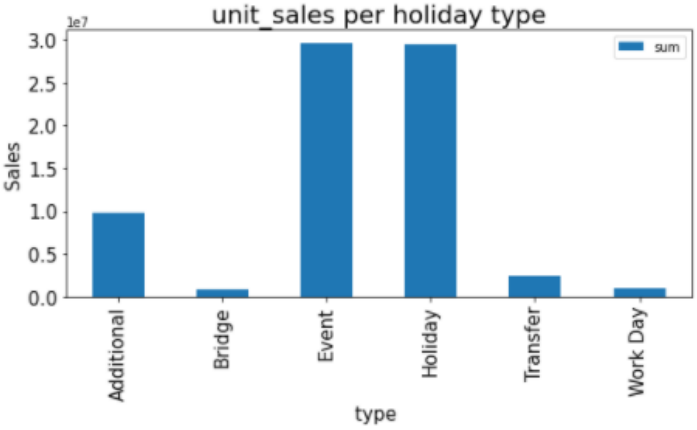
בגרף הבא ניסינו לבדוק כמה יחידות נמכרות עבור מוצרים מתקלקלים ומוצרים שאינם מתקלקלים בשנת 2016. ניתן לראות שמוצרים שאינם מתקלקלים נמכרים בכמות גדולה יותר מאשר לא מתקלקלים דבר שעשוי להשפיע בעת על הכמויות הנמכרות.



בגרף הבא ניסינו לבדוק האם כמות היחידות הנמכרות מושפעת לפי אירועים לאומיים/ אזוריים או מקומיים בשנת 2016. ניתן לראות שבאירועים לאומיים יותר יחידות נמכרות ביחס לסוגי החגים האחרים. כלומר ייתכן ויש השפעה על כמות היחידות הנמכרות בעת אירוע שהוא לאומי/ אזורי או מקומי ואפשר להשתמש בפיצ'ר זה כדי לחזות מכירות ביום שהוא נופל באחד מהחגים האלה.



בגרף זה ניסינו לבדוק את כמות המכירות בכל אזור (עיר במדינה או בכל המדינה) בעת תאריך של אירוע מסויים. ניתן לראות שבעת שיש אירוע לאומי באקוודור נקנים הכי הרבה יחידות, כלומר ניתן להסיק או שיש יותר אירועים לאומיים או שהם פרוסים על יותר זמן. לעומת זאת בערים/ אזורים שיש אירועים שהם לא לאומיים יש מעט יותר מכירות שזה אולי נובע מכך שאין הרבה אירועים כאלו בשנה או אולי הם פרוסים על מעט ימים ביחס לאירועים לאומיים.



בגרף הבא ניסינו לבדוק כמו יחידות נמכרות בשנת 2016 לפי סוג האירוע. ניתן לראות שיותר נמכרות בעת אירועים שמוגדרים בחגים או באירועים שאינם מוגדרים כחגים. כלומר בתאריך שמוגדר כחג כמות היחידות נמכרת בצורה גדולה יותר לעומת סוגי אירועים אחרים ולכן זה פיצ'ר זה אכן יכול לסייע לנו בתהליך האימון.

בגרף הבא ניסנו לבדוק האם לעלייה או ירידה במחירי השמן לאורך שנת 2016 יש השפעה על כמות היחידות המוצרים שנמכרים. מהגרף ניתן לראות שבתחילת השנה מחיר השמן היה נמוך ובאמצע השנה המחיר עלה ויחסית התייצב. אבל ניתן לראות שלכל אורך השנה כמות המכירות לא השתנתה בצורה משמעותית ביחס לשינויי המחיר של השמן ולכן אנו חושבים שאין בהכרח השפעה של מחיר השמן על המכירות ואולי נמנע מלהשתמש בנתון זה בנתוני האימון שלנו.

לאור הניתוח שנ"ל שבוצע על הנתונים בחרנו לאמן את המודלים על כל חודשי אוגוסט מכל השנים לאור העובדה שאנו מתבקשים לחזות את השבועיים האחרונים של אוגוסט 2017 ולכן לדעתנו נכון לקחת את אותו חודש מכל השנים כך שלא תהיה הטיה בתוצאות עקב מכירות בחודשים אחרים.

סעיף b- קביעת solid benchmark בעזרת אלגוריתם ML

בסעיף זה התבקשנו להשתמש באלגוריתם קלאסי של ML כדי לבצע פרדיקציה על קובץ המבחן ולקבוע solid benchmark שלפיו נוכל לשפר את המודלים שנבנה בהמשך ולראות את השיפור.

לסעיף זה בחרנו להשתמש ב- Random Forest Regressor.

עבור מודל זה בחרנו להשתמש בפיצ'רים הבסיסים של item, store ו-date כדי לקבל solid benchmark בסיסי ואז בהמשך אנו מתכוונים להוסיף עוד פיצ'רים ולראות עד כמה הם השפיעו ביחס לאותו solid benchmark שקבענו.

את הנתונים לאימון המודל קבענו שיהיו חודשי אוגוסט של השנים 2013-2016 ואילו נתוני הוולידציה קבענו שיהיו השבועיים הראשונים של אוגוסט 2017.

לאחר כמה הרצות שונות קבענו את ה- hyper parameters הבאים ל- Random Forest Regressor:

```
RandomForestRegressor(n_estimators=25, random_state=10, max_depth=15, n_jobs=-1, criterion='mse')
```

בתרגיל נתבקשנו להשתמש במטריקה:

Normalized Weighted Root Mean Squared Logarithmic Error (NWRMSLE)

מטריקה זו מתאימה בעת חיזוי ערכים בטווח גדול של סדרי גודל וזה מונע ענישה על הבדלים גדולים בחיזוי כאשר גם החיזוי וגם המספר האמיתי גדולים. במשימה יש טווח רחב של יחידות מוצרים שנמכרות ולכן משתמשים במטריקה זו.

בנוסף השתמשנו גם במטריקה הידועה: MSE.

בעזרת השימוש בשני מטריקות אלו נקבל אומדן שלפיו נרצה להשתפר במודלים הבאים שניצור בשימוש עם יותר פיצ'רים שנבחר לנכון.

תוצאות שקיבלנו עבור Random Forest Regressor על סט האימון והוולידציה בשימוש המטריקות הנ"ל הם:

```
Mean Squared Error: - train: 0.55
NWRMSLE RF Train 0.7376459986560973
Mean Squared Error: - val: 0.57
NWRMSLE RF Val 0.747039748388704
```

בנוסף ביצענו פרדיקציה על קובץ נתוני המבחן ואתר קובץ התוצאות הגשנו לאתר kaggle על מנת לקבל ציון שיהווה עבורנו solid benchmark עבור נתוני המבחן.

Name	Submitted	Wait time	Execution time	Score
submission_rfr.csv	a few seconds ago	0 seconds	14 seconds	1.24498

Complete

Jump to your position on the leaderboard

סעיף c- הצגת תהליך יצירת מודל Embedding

בסעיף זה נתבקשנו ליצור מודל embedding בסיסי בעזרתו נוכל לחזות את כמו המכירות לכל יום עבור מוצר וחנות בקובץ המבחן. בחרנו לבצע את ה-embedding על שני הפיצ'רים הבסיסיים של מוצר וחנות ובנוסף התייחסנו גם לתאריך וביצענו עליו גם embedding כאשר פרקנו את התאריך ליום, חודש ושנה.

שלבי התהליך:

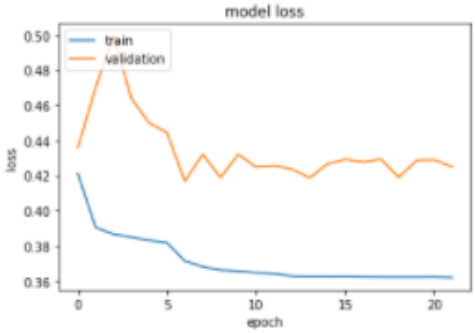
1. עבור נתוני האימון שבחרנו לעבוד עליהם ביצענו merge עם קובץ המוצרים על מנת שיהיה לנו גם את המשקלים שבעזרתם אנו נוכל להשתמש במטריקה NWRMSLE.
2. טענו את קבצי המוצרים והחנויות ובעזרת הערכים היחודיים שנמצאים בקבצים הללו נבצע את שלב ה-enumerate . בחרנו לעשות זאת על ערכים אלה ולא מהערכים שבקובץ האימון מכיוון שיש פריטים שמופיעים בקובץ המבחן ואינם מופיעים בקובץ האימון ולכן אנו רוצים לוודא שיש התאמה ב-לכלל המוצרים והחנויות וכדי שזה יקרה נכון יותר לבצע זאת על ערכים מקבצים.
3. פירקנו את התאריך מהפורמט המקורי שלו ליום, חודש ושנה כי רק כך נוכל לבצע embedding עבור כל אחד מהם.
4. ביצענו enumerate עבור כל אחד מהפיצ'רים שבחרנו (מוצר, חנות, יום, חודש ושנה) כך שכל פיצ'ר יהיה מסופרר מ-0 עד n-1.
5. ביצענו התאמה בין הערכים משלב ה-enumerate שיצרנו לפיצ'רים של נתוני האימון שבחרנו וקיבלנו נתוני אימון עם הפיצ'רים בהתאם לערכים שהתקבלו ה- enumerate עבור כל פיצ'ר. בנוסף ביצענו לוג על נתוני המכירות כדי שנוכל להשתמש ב-.
6. בשלב זה כשהיה לנו את נתוני האימון מוכנים יצרנו שכבות Input ו- embedding עבור כל אחד מהפיצ'רים.

לאחר כל השלבים האלו בנינו את המודל הבסיסי הראשון אליו נכניס את ה-Inputs ושכבות ה-embedding שיצרנו.

```
x = concatenate([year_emb, month_emb, day_emb, store_emb, item_emb], name='embedding_model')
x = Flatten()(x)
x = BatchNormalization()(x)
x = Dense(32, activation='relu')(x)
x = Dense(16, activation='relu')(x)
x = Dropout(0.4)(x)
x = BatchNormalization()(x)
x = Dense(8, activation='relu')(x)
x = Dense(8, activation='relu')(x)
x = Dropout(0.25)(x)
x = Dense(1, activation='relu')(x)
embedding_model = Model([year_inp, month_inp, day_inp, store_inp, item_inp], x)
```

סעיף d- תוצאות המטריקות על נתוני הווליידציה והמבחן

בשלב זה פיצלנו את נתוני האימון ל ו- ונתנו למודל להתאמן על 30 אפוקים כאשר הוא ישמור במהלך האימון את המשקלים הטובים ביותר ובהם נשתמש לחיזוי על נתוני המבחן. לאורך האימון השתמשנו במטריקה כדי לראות את השיפור בין כל אפוק.



ניתן לראות שבהתחלה המודל לא למד בצורה טובה והיה ב-overfitting גבוהה אבל לאחר מכן התחיל ללמוד עד אפוק 8 ולאחר מכן התייצב ולא הייתה למידה משמעותית.

לפי המטריקה MSE שלפיה הערכנו את המודל הערך הכי נמוך היה באפוק 7 והוא היה: `val_mse: 0.4067`

בנוסף ביצענו הערכה על נתוני הווליידציה לפי המטריקה NWRMSLE וקיבלנו את התוצאה:

```
NWRMSLE Embedding - Validation: 0.6398163383110098
```

לאחר שסיימנו לאמן את המודל טענו את המודל עם המשקלים הכי טובים וביצענו פרדיקציה על קובץ המבחן. נציין לפני זה ביצענו גם עיבוד לקובץ המבחן כדי שיהיה תואם לפי ערכי ה-enumerate שביצענו בעזרת קבצי הפריטים והחנויות בשלבים הקודמים.

את התוצאות החיזוי של היחידות שנמכרו העלנו בלוג והתאמנו ל-id של הפריטים בקובץ המבחן (לפי פורמט ההגשה) והגשנו את הקובץ ל-Kaggle למתן ציון.

Name	Submitted	Wait time	Execution time	Score
submission_emb.csv	a few seconds ago	0 seconds	16 seconds	1.04010

Complete

Jump to your position on the leaderboard

ניתן לראות שביחס ל-solid benchmark שקיבלנו עבור ה- Random Forest Regressor ישנו שיפור ניכר גם בתוצאות המטריקות שביצענו על נתוני הווליידציה וגם בתוצאה שקיבלנו עבור ההגשה ל-Kaggle של תוצאות הפרדיקציה על נתוני המבחן. אנו מעריכים שדבר זה נובע מכך שמודל ה-Embedding בנוי בצורה יותר מותאמת מבחינת השכבות שבהן בחרנו להשתמש. בנוסף אימנו את המודל יותר זמן, שמרנו את המודל עם המשקלים הטובים ביותר ואותו טענו לשימוש לצורך הערכה בעזרת המטריקות על נתוני הווליידציה וחיזוי קובץ המבחן.

יש לציין כי מודל ה-Embedding שיצרנו וגם המודל של Random Forest Regressor הם מודלים בסיסיים שביצענו בעזרת הפיצ'רים הכי בסיסיים כדי לקבל solid benchmark בסיסי ולכן לדעתנו התוצאות יחסית נמוכות. בסעיפים הבאים אנו נשתמש בפיצ'רים נוספים שלדעתנו ישפרו את תהליך הלמידה של המודל ובנוסף נשפר גם את מבנה המודל כאשר המטרה היא להתעלות מעל ה- solid benchmarks שקיבלנו בסעיפים אלו.

סעיף e- יצירת מודל Embedding מורכב יותר בשימוש עם יותר פיצ'רים

מאפיינים נוספים למודל

- מאפיינים קטגוריאליים – בחרנו להוסיף מאפיינים קטגוריאליים מגוונים כך שניתן לחלק את המאפיינים לשלושה קטגוריות מרכזיות –

1. מאפייני פריט – המאפיינים שהשתמשנו הם מזהה פריט, האם הפריט בהנחה, המשפחה של הפריט והאם הפריט מתקלקל.
2. מאפייני חנות – המאפיינים שהשתמשנו הם מזהה חנות, עיר , מחוז, סוג החנות (type) ו- cluster של החנות.
3. מאפייני תאריך- המאפיינים שהשתמשנו בהם הם יום בחודש, יום בשבוע, חודש, שנה, תשלום (משתנה בוליאני שמקבל ערך 1 בין התאריכים 1-4,15-18 שמסמל קבלת משכורות במגזר הציבורי), האם התאריך נמצא לפני או אחרי חג (עד 3 ימים) , האם התאריך הוא חג, סוג החג (יום רגיל מקבל ערך משלו) והשם של החג.

- מאפיינים נומריים – מקובץ המכירות של החנויות סכמנו לכל חנות את המכירות בתקופת זמן של חודש ובתקופות זמן של חודש עבור כל יום בשבוע. לכל תקופת זמן הוצאנו 3 מאפיינים שהם הממוצע, החציון והשונות באותה תקופה. עבור כל רשומה חישבנו את התקופה הקודמת (כלומר עבור חודש את החודש הקודם לו) ובנוסף גם הסתכלנו שנה אחורה והוספנו את המאפיינים של אותה תקופת זמן שנה קודמת. סה"כ הוספנו 12 מאפיינים נומריים המורכבים מסכימה לאורך 2 תקופת זמן שונות עם הסתכלות לעבר הקרוב (חודש אחורה) והסתכלות על תקופה זהה בשנה שעברה.
- בחרנו לנרמל את המשתנים הנומריים לטווח של [0,1] מכיוון שרשתות צריכות את הערכים מנורמלים כדי להתכנס מהר יותר וערכים לא מנורמלים יכולים להוביל לנגזרות גבוהות מדי ואז שינויים קיצוניים במודל ימנעו ממנו להתכנס. את הערכים נרמלנו לפי עמודות כלומר העדפנו לראות את ה"הבדל" בין החנויות השונות עבור כל תקופת זמן, אך אופציה נוספת שניתן לעשות היא לבצע את הנרמול עבור השורות ואז נוכל לראות את ה"הבדל" בין אותה חנות בתקופות זמן שונות. את אותו תהליך ניתן לעשות עבור פריט ואז לראות איזה פריטים הם יציבים לאורך השנה כולה ואיזה פריטים הם עונתיים. בחרנו לא להוסיף את מאפיינים האלה בגלל זמן החישוב הארוך ובגלל שגם ככה הנתונים שלנו היו כבדים מאוד.

המודל

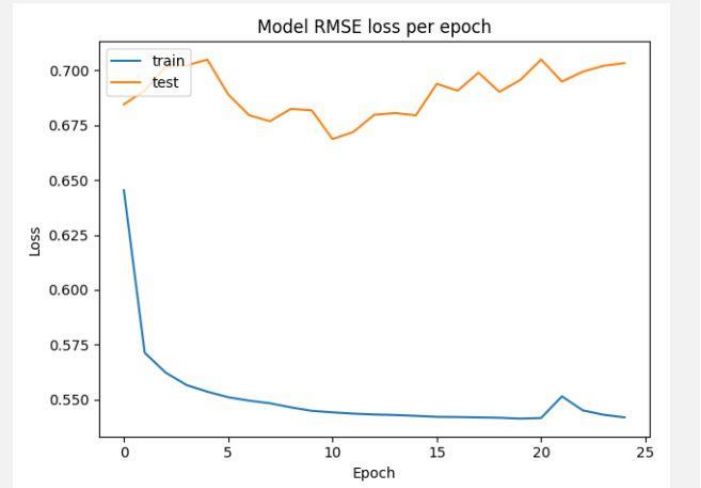
- המודל בו בחרנו להשתמש הוא מודל של רשת עמוקה שמורכב ממשתנים נומריים וקטגוריאליים יחדיו.

סה"כ השתמשנו ב-38 קלטים – 12 נומריים, 26 קטגוריאליים.
עבור שכבות ה- embedding לקחנו גודל פלט של המקסימום בין חמש ל-log של מספר הערכים השונים באותה קטגוריה, השתמשנו בפונקציית אקטיבזציה מסוג relu כדי לתמוך בבעיות רגרסיה למרות שיש ערכים שליליים למעט מהערכים, בגלל שזה ממש imbalance.
שכבות נוספות שהשתמשנו הן dropout, batch normalization, concacante .

המודל הראשון בו השתמשנו הוא –

```
[12] x = concatenate([item_emb,item_family_emb,item_class_emb,item_onpromotion_emb,item_perishable_emb,store_emb,store_cluster_emb,
                      store_type_emb,store_city_emb,store_state_emb,before_holiday_week_emb,is_after_holiday_week_emb,
                      is_holiday_day_emb,holiday_type_emb,holiday_locale_emb,holiday_locale_name_emb,payment_week_emb,
                      month_emb,year_emb,weekday_emb,day_emb])
     x = Flatten()(x)
     x = BatchNormalization()(x)
     x = Dense(64,activation='relu',kernel_regularizer=l2(1e-4))(x)
     x = Dense(32,activation='relu',kernel_regularizer=l2(1e-4))(x)
     x = Dropout(0.5)(x)
     x = BatchNormalization()(x)
     x = Dense(16,activation='relu',kernel_regularizer=l2(1e-5))(x)
     x = Dense(16,activation='relu',kernel_regularizer=l2(1e-5))(x)
     x = Dropout(0.5)(x)
     x = BatchNormalization()(x)
     x = Dense(8,activation='relu',kernel_regularizer=l2(1e-5))(x)
     x = Dense(1,activation='relu',kernel_regularizer=l2(1e-5))(x)
```

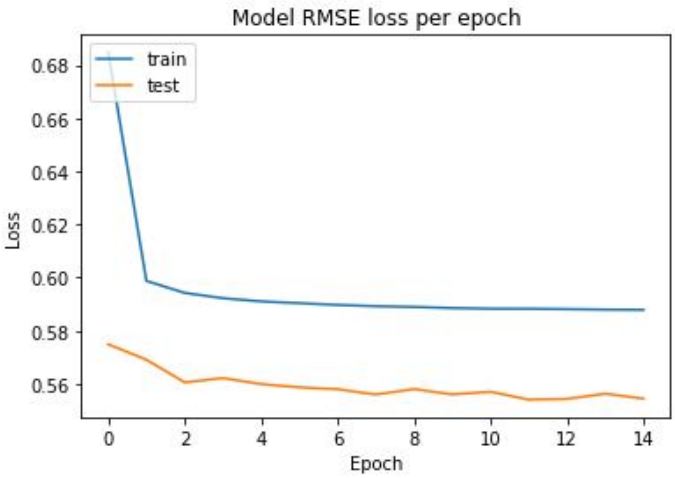
תוצאות המודל –



תוצאת Kaggle – 1.061

מכיוון שהמודל נמצא בהתאמת יתר וראינו שהוא מתכנס מהר מאוד לפעמים תוך שני אפוקים וכל שיפור קטן בתוצאות האימון מוביל להגדלת התאמת היתר החלטנו להוסיף עוד שכבת dropout בין השכבות הצפופות, ולהוריד את השכבה האחרונה ברשת.

תוצאות המודל החדש –

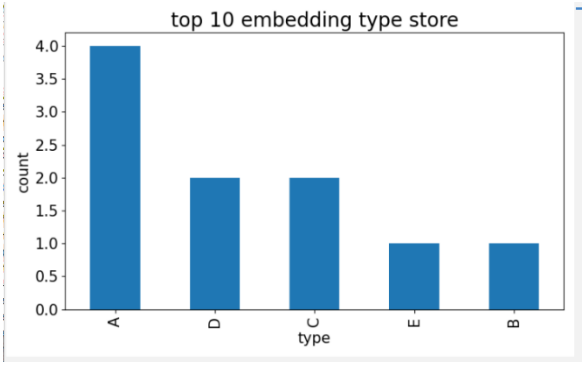


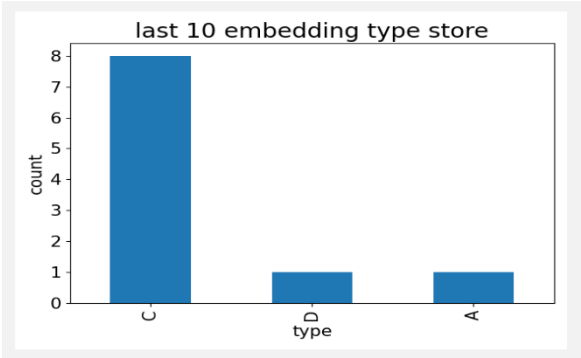
תוצאת Kaggle – 1.05985

השינוי שביצענו גרם לירידה חדה בהתאמת יתר אבל גרם למודל להיכנס למצב של underfitting למרות זאת הצלחנו לשפר קצת תוצאות המודל. ביצענו כמה ניסיונות נוספים לשפר את תוצאות המודל על ידי שינויים בגודל ה-batch וה-dropout אך ניסיונות אלה לא הובילו לשיפור משמעותי במודל.

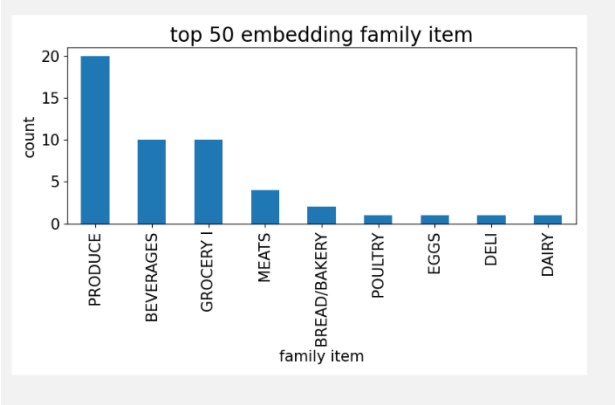
סעיף f- ניתוח ותובנות של שכבות ה-Embedding

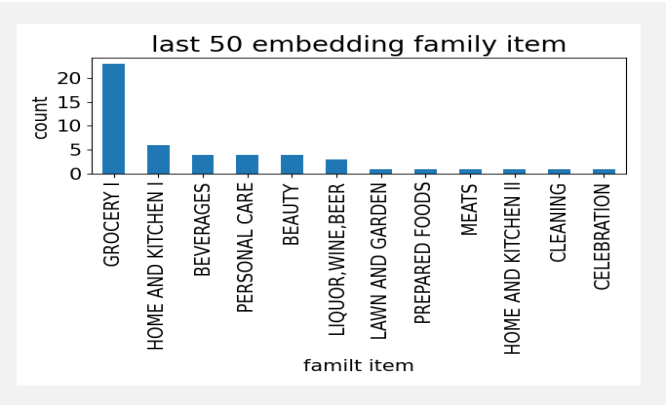
ניתוח וקטורי החנות – אנחנו רואים שהחנויות מסוג C נמצאות בתחתית הרשימה וחנויות מסוג A נמצאות בראש הרשימה, אולם בהמשך הרשימה יש הרבה יותר שוני מאשר בפריטים או בווקטורים האחרים שבדקנו. כאשר אנחנו מסתכלים על המכירות לפי סוג החנות (בסעיף בקודם) ניתן לראות כי סוג A נמצא גבוה ביחס ל-C אבל יש גם את סוג E שנמצא אצלנו במרכז הטבלה ולכן זה לא חד משמעי.





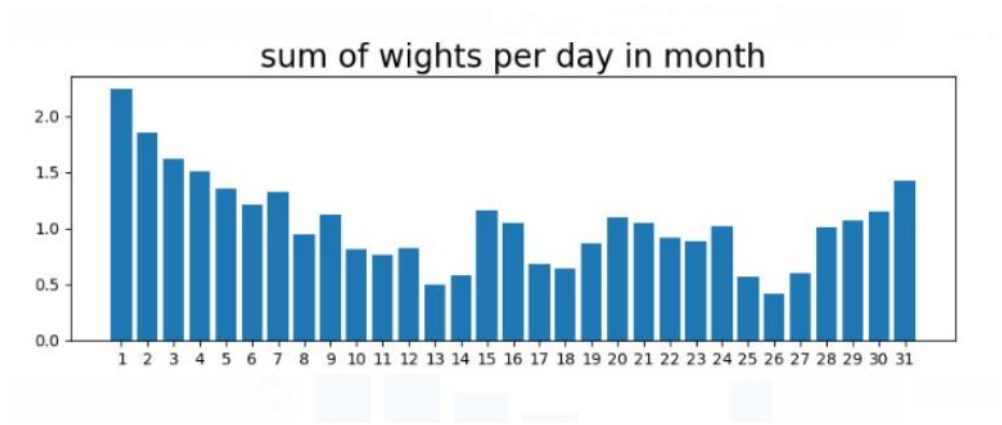
ניתוחי וקטורי הפריטים – אנחנו רואים שמוצאי תבואה, ביצים , בשר ומה שאפשר להגדיר כמוצרי צריכה בסיסיים הם בעלי השקלול הגבוה ביותר בשקלול שכבות ה-embedding, לעומת זאת אנחנו רואים יותר מוצרי מותרות בתחתית הרשימה.



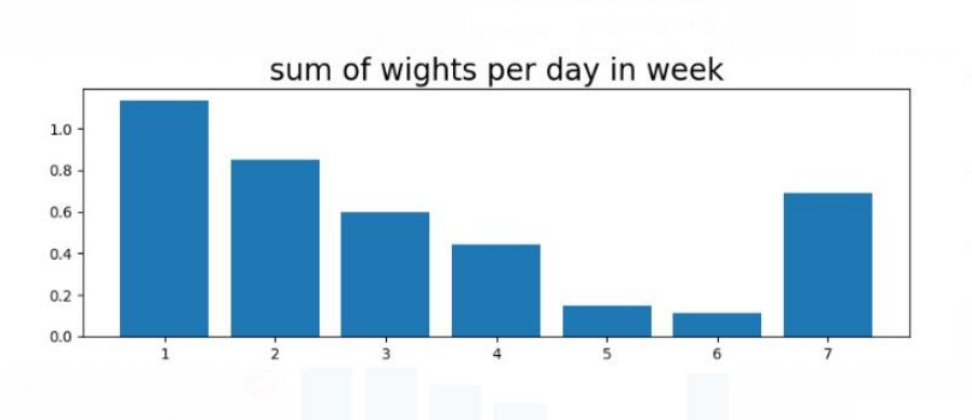


ניתוחים נוספים -

- וקטור הימים בחודש ניתן לראות כי הימים הראשונים בחודש והיום האחרון בחודש הם בעלי המשקל הגבוה ביותר וזה מתאים לגרפים שראינו בחקירת הנתונים שבימים אלה אכן כמות הקניות גבוה יותר.



- יום בשבוע – אנחנו רואים כי הימים שיש להם את המשקל הכולל הגבוה ביותר הם ימי סופש שבת-שני, זה נראה לנו הגיוני כי זה סופ"ש במדינת אקוודור.



- בשאר שכבות ה-emb לא הצלחנו למצוא מגמה ברורה אולי מכיוון שבסופו של דבר על מנת לקצר את תהליך האימון בחרנו לאמן על חודשי יולי-אוגוסט בלבד.

סעיף g– ביצוע feature extractor על ה- Embedding לשימוש באלגוריתם ML

בחרנו להשתמש במודל של catboost מכיוון שיצא לנו לעבוד איתו וראינו שהוא יכול לספק ביצועים טובים והכי חשוב הוא מאפשר טעינה של מודל קודם והמשך לימוד כך שנוכל לאמן את המודל למרות המידע העצום.

החלטנו להעביר למודל את השכבות הנומריות בהן ראינו שונות שניתנת להסבר בנתונים והן מזהה חנות, פריט יום בשבוע ויום בחודש בנוסף למשתנים הנומריים שהוצענו בסעיף הקודם.

אימנו את המודל בצורה איטרטיבית.

```python
def create_catBoost_model(data_path,test_path,test_output):
    model = CatBoostRegressor(iterations=2000,
                              learning_rate=1e-2,random_state=42,
                              task_type='CPU',
                              l2_leaf_reg=1e-4, bootstrap_type='MVS',subsample=0.35)
    flag=True
    for chunk in pd.read_csv(data_path, chunksize=1000000):
        target=chunk.loc[:,['unit_sales']]
        chunk.drop(columns=['unit_sales'],inplace=True)
        chunk['day']=chunk['day'].astype(int)
        chunk['weekday'] = chunk['weekday'].astype(int)
        chunk['store_nbr'] = chunk['store_nbr'].astype(int)
        chunk['item_nbr'] = chunk['item_nbr'].astype(int)
        chunk['onpromotion'] = chunk['onpromotion'].astype(int)
        chunk['type'] = chunk['type'].astype(int)
        chunk['cluster'] = chunk['cluster'].astype(int)
        X_train, X_test, y_train, y_test = train_test_split(chunk, target, test_size=0.25, random_state=42,shuffle=True)
        y_train_normalize = np.log1p(y_train)
        y_test_normalize = np.log1p(y_test)
        if flag:
            model.fit(X_train,y_train_normalize,eval_set=(X_test,y_test_normalize),
                      use_best_model=True,cat_features=['day','weekday','store_nbr','item_nbr','onpromotion','cluster','type'])
            flag=False
        else:
            model.fit(X_train,y_train_normalize,
                      init_model='model.cbm',
                      eval_set=(X_test,y_test_normalize),
                      use_best_model=True,
                      cat_features=['day','weekday','store_nbr','item_nbr','onpromotion','cluster','type'])
        model.save_model('model.cbm')
```

תוצאה ראשונה מ- Kaggle:

Name	Submitted	Wait time	Execution time	Score
sample_submission.zip	a few seconds ago	0 seconds	18 seconds	0.97443

Complete

Jump to your position on the leaderboard

אנחנו רואים שיפור קל בתוצאות, שמנו לב שהמודל ממשיך להשתפר גם באיטרציות האחרונות ועוד לא התכנס ולכן אנו מעריכים כי אימון ארוך יותר יוכל בשאיפה להוריד את השגיאה עוד יותר.

לפי הנחה זו בחרנו לאמן שוב את המודל עם אפוק אחד יותר ולטעון שוב את המשקלים הטובים ביותר ולאמת את ההנחה אם אימון ארוך יותר יוכל להוריד את השגיאה עוד יותר. לאחר מכן הגשנו שוב את תוצאת הפרדיקציה ל-kaggle והתוצאה אכן השתפרה, כלומר ככל שנאמן את המודל זמן רב יותר ייתכן והתוצאות ישתפרו, בחרנו לעצור בנקודה זו מכיוון שתהליך זה מאוד ארוך ועשוי לקחת זמן רב לאור והמשאבים העומדים לרשותנו.

תוצאה שנייה מ- Kaggle:

Name	Submitted	Wait time	Execution time	Score
sample_submission.csv	a few seconds ago	0 seconds	20 seconds	0.94287

Complete

Jump to your position on the leaderboard

סיכום המשימה

לכל אורך המשימה אנו רואים כי שיפור המודל והשימוש בפיצ'רים נוספים קטגוריאליים וגם נומריים מסעיף לסעיף הביא בסופו של דבר לשיפור בתוצאות.

האתגר הגדול בתרגיל הזה מבחינתנו היה התמודדות עם הכמות העצומה של הנתונים, ישנם מיליוני רשומות עבור כל חודש ולכן יש צורך לבחור את הדאטה שיתאים הכי טוב לתקופת החיזוי. בחרנו בהתחלה לקחת שני סוגי מאגרי מידע הראשון של שנת 2017 כולה והשני של חודשי יולי-אוגוסט של שנים קודמות, כך נוכל לראות את ההבדל בין מאגרי המידע אך מכיוון שתהליך האימון היה ארוך מאוד (למעלה מ-3 שעות לאפוק) החלטנו לוותר על נתוני 2017 ולהשתמש רק בנתוני יולי-אוגוסט. בעיה נוספת שנתקלנו בה היא שלא יכולנו לטעון את כל הדאטה לזיכרון ולכן השתמשנו בפונקציה שמקטינה את גודל הערכים של dataframe בזכרון ובנוסף בחלקים מתקדמים ממשנו data generator שיטען לנו רשומות ב-batch. בהתחלה בחרנו גודל

סטנדרטי של 64 רשומות אך בגלל שזה הוביל לזמן אימון ארוך מאוד החלטנו להגדיל את גודל ה-batch ל - 1024\512 רשומות, את ההשפעה של זה ראינו בכך שעקומת הלמידה של המודל יותר נמוכה אך הוא המשיך ללמוד לאורך מספר רב של אפוקים לעומת גודל -batch קטן שבו ראינו התכנסות כבר תוך מספר אפוקים בודד. גודל הפלט בשכבת ה-embedding קבענו ל-log מספר הערכים הייחודיים או 5 עבור רוב הרשומות אך קטגוריות שראינו שיש להם חשיבות גבוה יותר נתנו 7 (כמו יום בחודש). כאשר ניסינו להגדיל את גודל שכבת הפלט של וקטורי ה-embedding להיות פי 2 ממה שקבענו בהתחלה ראינו כי ההתאמת יתר היא גבוה יותר.

אנחנו חושבים שהסיבה שלא הצלחנו להגיע לתוצאות טובות יותר היא שזמן האימון שעשינו לא היה מספיק על מנת להביא את שכבות ה-embedding לרמה מספקת, אימון על יותר נתונים לאורך תקופות ארוכות יותר היה עוזר לנו למצוא תובנות גם בשכבות נוספות של ה-embedding וכך עוזר לנו לשפר את התוצאה.

בסופו של דבר אנו מרוצים מהתוצאה הסופית ומהשיפור שחווינו לכל אורך הסעיפים של המשימה למרות שם היו לנו יותר משאבים חזקים להתמודד עם כמות כזו של נתונים ייתכן והיינו מגיעים לתוצאות טובות יותק.