Семинарска работа по предметот

Вовед во науката за податоци

Тема: **Алгоритам за динамичко поставување на цени на производи од различни категории врз база на побарувачката и конкуренцијата**

Ментор:

асс. Милена Трајаноска

Изработил:

Леонид Давитковски 203212

Линк до проектот на GitHub:

https://github.com/le0nid36/VNP_StandardenProekt/tree/master

1. Вовед

Во современата конкурентна малопродажна индустрија, цените играат клучна улога во профитабилноста и пазарниот удел. Фиксните стратегии често се недоволни за справување со флуктуации во побарувачката, однесувањето на конкурентите и економските услови. Затоа, динамичното одредување цени — прилагодување на цените во реално време според различни фактори — станува сè почеста практика, особено во е-трговијата.

Целта на овој проект е да се истражи примената на техники за машинско учење во градење динамичен систем за одредување цени за производи од различни категории. Проектот се фокусира на развој на модели за предвидување оптимални цени во различни пазарни услови, помагајќи им на трговците на мало да носат подобри одлуки за цени.

2. Дефинирање на проблемот

Определувањето на точна цена е сложен процес што зависи од многубројни фактори. Неправилната стратегија може да предизвика губење на продажба, намалување на профитните маржи, акумулирање залихи или слаб настап за време на промоции.

Традиционалните пристапи како cost-plus често ги игнорираат сложените интеракции меѓу факторите како залихи, побарувачка, конкуренција и сезоналност. Со оглед на тоа што конкурентите ги менуваат цените динамично, потребни се пофлексибилни и адаптивни модели.

Како што напоменав погоре главната цел на овој проект е дизајн и имплементација на систем за динамично одредување цени за малопродажни производи од повеќе категории, базиран на техники од машинско учење.
За тоа да го направам ќе ги прикажам и анализирам факторите што најмногу влијаат врз тоа како се движат цените на производите.
Исто така ќе го прикажам процесот на изработка на модели кои ќе предвидуваат оптимални цени, земајќи предвид внатрешни (залихи, побарувачка) и надворешни (конкуренција, промоции, временски услови) фактори и сето тоа ќе го визуелизирам со различни графици со цел да ја прикажам нивната ефективност.

Овој проект има за цел да придонесе кон развојот на интелигентни системи за оптимизација на цените во малопродажбата и да покаже како машинското учење може да помогне во подобри стратешки одлуки.

3. Опис на податочното множество

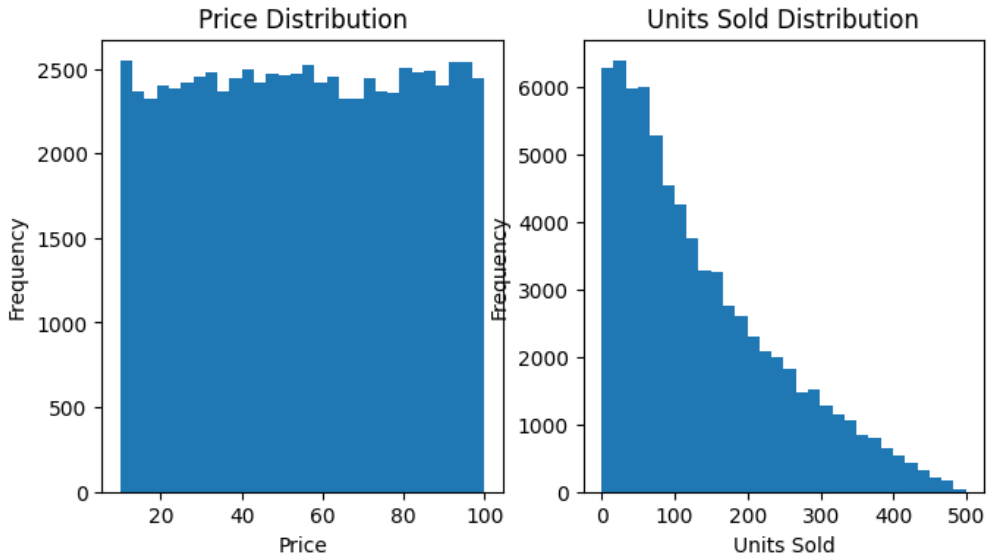
За овој проект, го користев податочното множество „Retail Store Inventory Forecasting Dataset“, јавно достапно на Kaggle, кое обезбедува сеопфатна колекција на податоци за малопродажни трансакции погодни за предвидување на временски серии, управување со залихи и анализа на цените. Датасетот се состои од повеќе од 73000 редици со клучни карактеристики како:

Date: Дневни записи од [почетен_датум] до [краен_датум]
Store ID & Product ID: Уникатни идентификатори за продавници и производи
Category: Категории на производи како електроника, облека, намирници итн
Region: Географски регион на продавницата
Inventory Level: Достапна залиха на почетокот на денот
Units Sold: Продадени единици во текот на денот
Demand Forecast: Предвидена побарувачка врз основа на минатите трендови
Price: Цена на производ
Discount: Попуст на производ
Weather Condition: Време во текот на денот кое влијае на продажбата
Holiday/Promotion: Индикатори за празници или промоции
Competitor Pricing: Цена на производот на конкурентите
Seasonality: Годишно време

4. Истражување и визуелизација на податочното множество

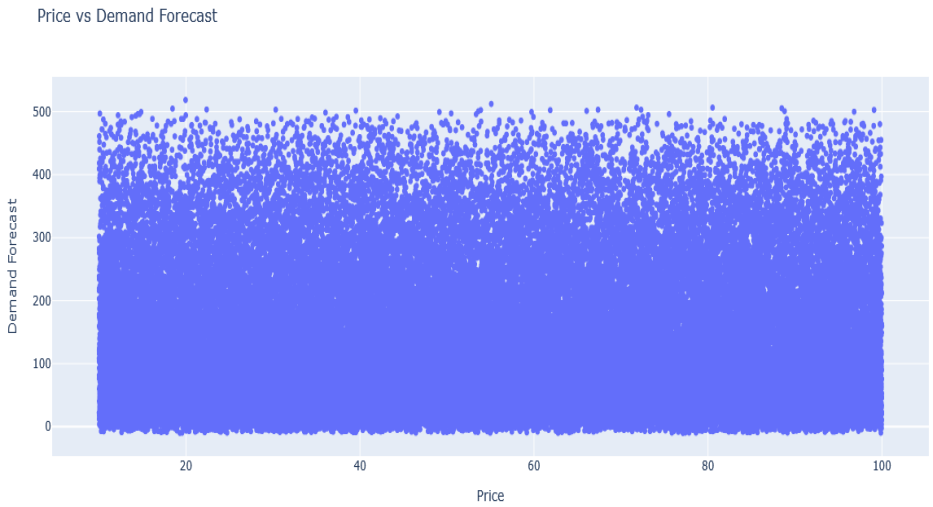
Започнав со разгледување на структурата и воочив дека датасетот се состои од 3 различни типови на податоци: object, int и float.
Податочното множество се состоеше од 8 нумерички колони и 7 категориски колони меѓу кои и Date колоната.
Во датасетот немаше вредности што недостасуваат така што немаше потреба од импутирање на вредности.
Следен чекор сметав дека треба да бидат различни визуелизации на податоците од датасетот и како тие се однесуваат помеѓу себе.

Првично направив визуелизација на таргет колоната Price и Units Sold колоната со која се гледа нивната дистрибуција.

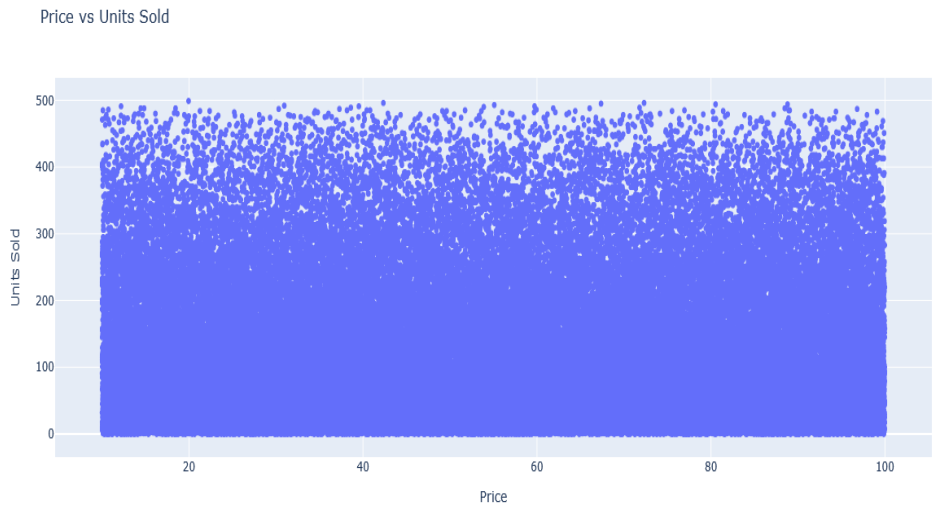


Price има прилично нормална распределба на вредности додека кај Units Sold јасно се гледа right-skewed дистрибуција.

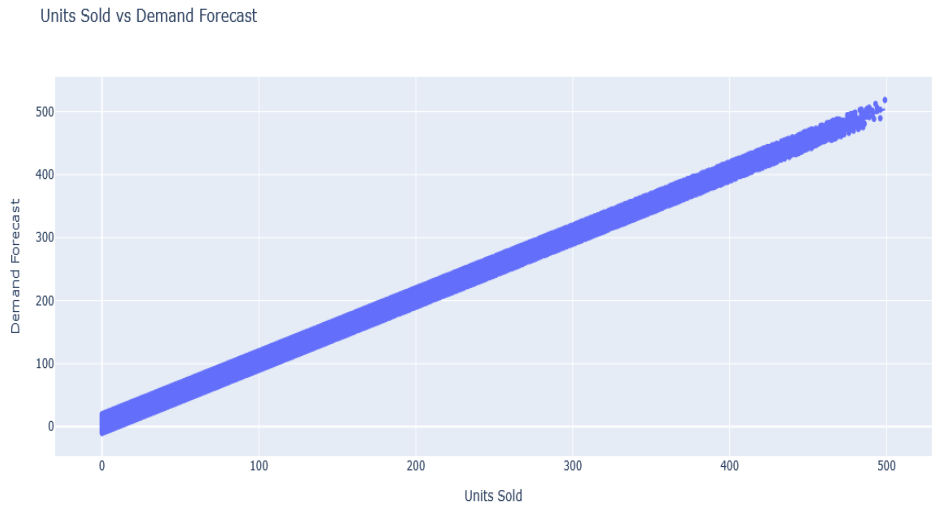
Следно направив повеќе визуелизации каде ги споредував податоците во колоните Price, Units Sold и Demand Forecast меѓусебно



Price vs Demand Forecast: Графиконот открива слаба негативна корелација помеѓу цената и побарувачката. Ова сугерира дека со зголемувањето на цените, побарувачката има тенденција да се намалува. Сепак, врската не е силна, што укажува дека и други фактори влијаат на побарувачката.

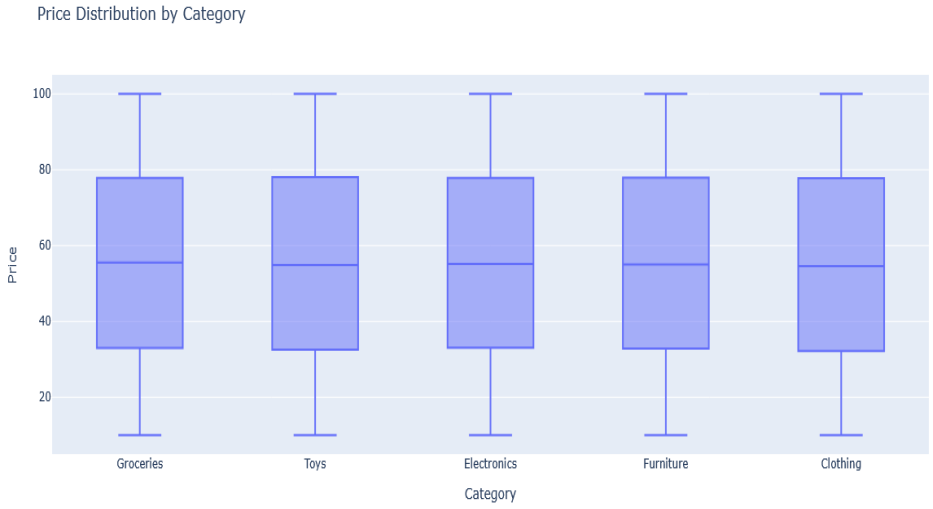


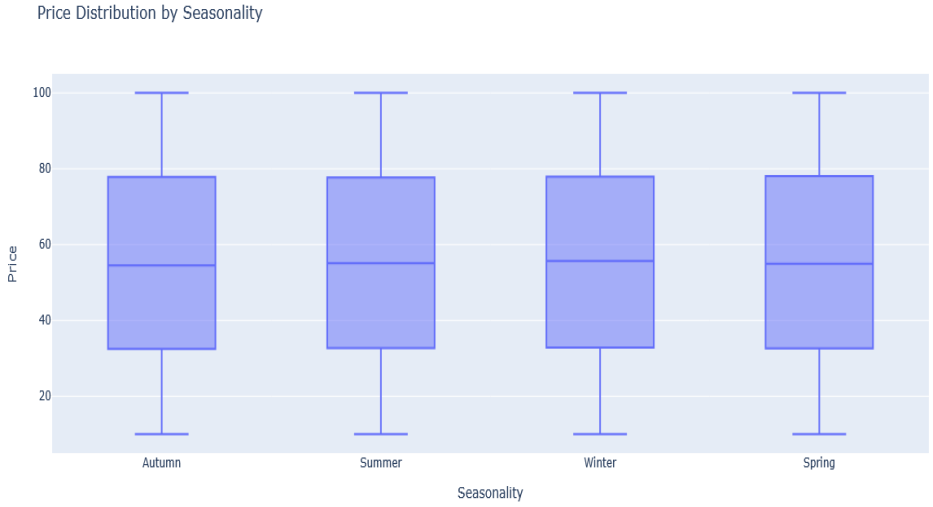
Price vs Units Sold: Покажува дека без разлика на цената бројот на продадени производи се движи рамномерно



Sales vs Demand Forecast: Графиконот открива силна позитивна корелација помеѓу продажбата и побарувачката. Ова е очекувано, бидејќи побарувачката е клучен двигател на продажбата. Графиконот сугерира дека со зголемувањето на побарувачката, продажбата исто така се зголемува пропорционално.

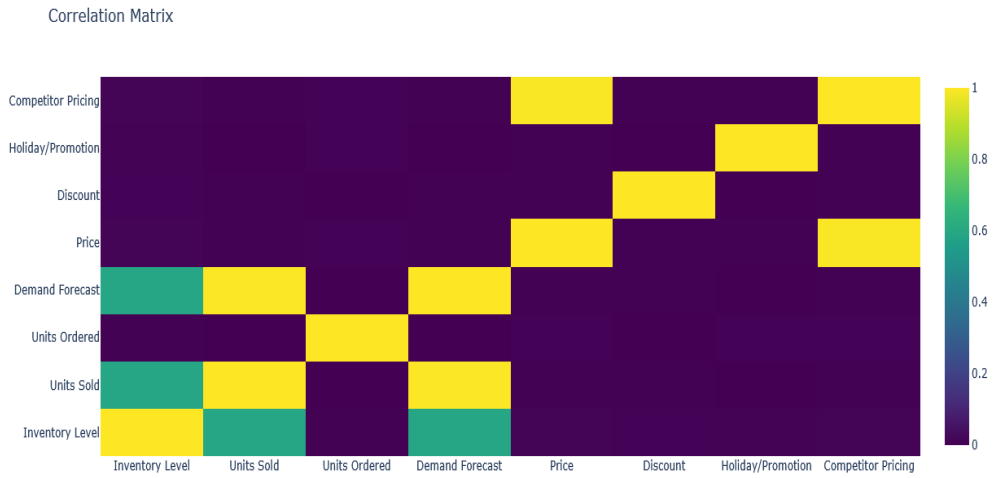
Со следните две графици ќе ја прикажам дистрибуцијата на цената во однос на различните категории и годишните времиња





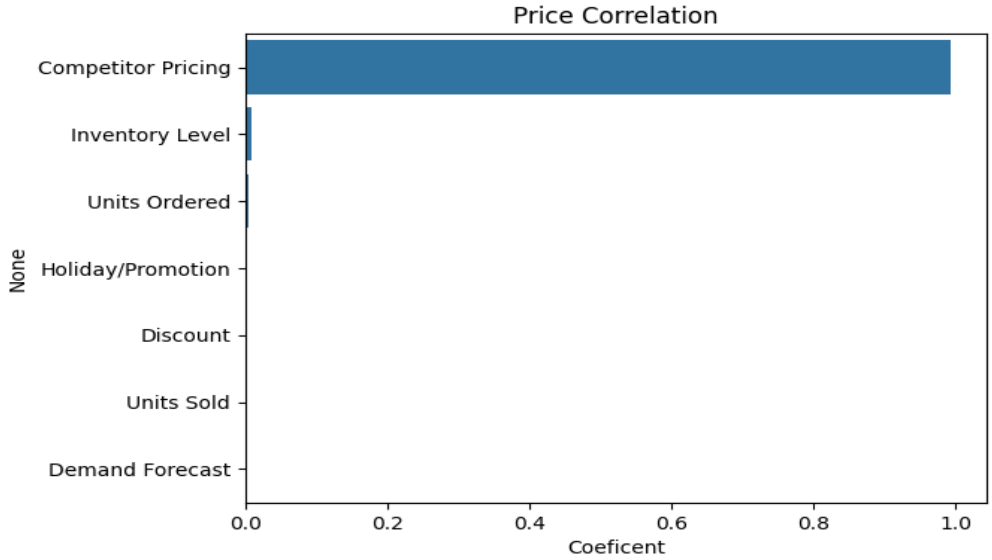
Слична дистрибуција имаат различните променливи и во двата графикона што покажува дека без разлика на тоа во која категорија припаѓа производот(мебел, облека, намирници, играчки, електроника) и кое годишно време е(есен, лето, зима, пролет) просечната цена на производите драстично не се менува.

Ги искористив нумеричките колони за да прикажам модел на корелација



Повеќе податоци покажуваат висока меѓусебна корелација: Units Sold со Inventory Level, Demand Forecast со Inventory Level, Units Sold со Demand Forecast и Price со Competitor Pricing.

-Units Sold и Inventory Level: Поголемите залихи често водат до поголема продажба бидејќи трговците нарачуваат повеќе од очекувано продаваните производи.
-Demand Forecast и Inventory Level: Залихите се прилагодуваат според прогнозите за побарувачка за да се избегнат недостатоци или вишок.
-Units Sold и Demand Forecast: Прогнозите добро се совпаѓаат со реалната продажба, што ја потврдува точноста на предвидувањата.
-Price и Competitor Pricing: Силната корелација покажува дека цените се често усогласени со конкурентите за да се остане конкурентен на пазарот.



Силната корелација помеѓу Price и Competitor Pricing повторно е потврдена овде.

Овие врски се корисни за моделот бидејќи даваат важни сигнали за предвидување на оптимални цени. Особено цените на конкурентите се клучни за динамично ценообразување, бидејќи моделот треба да реагира на пазарните промени.

Сепак, ваквите силни корелации може да укажуваат и на мултиколинеарност, што може да влијае негативно на некои модели како линеарната регресија. Затоа е важно внимателно да се изберат карактеристиките или да се користат модели што добро се справуваат со оваа појава, како Random Forest или XGBoost. Генерално, корелациската анализа потврди дека податоците се погодни за динамично одредување на цени.

5. Подготовка на податоците и енкодирање на категорисите податоци

Прво ја конвертирав колоната Date во pandas datetime за сортирање по датум.
Категориските колони (Category, Region, Weather Condition, Seasonality) ги енкодирав со one-hot encoding за да избегнам лажни нумерички односи.
Store_ID и Product_ID ги енкодирав со Label Encoder, бидејќи може да носат вредна информација за однесувањето на продавниците.

Од Date извлеков Year, Month, Day и Dayofweek за подобро разбирање на временските шеми.

Додадов и временски карактеристики:

-Price_lag1 (вчерашна цена) и Price_roll7 (7-дневен просек на цена)
-UnitsSold_lag1 (вчерашни продажби) и UnitsSold_roll7 (7-дневен просек на продажби)

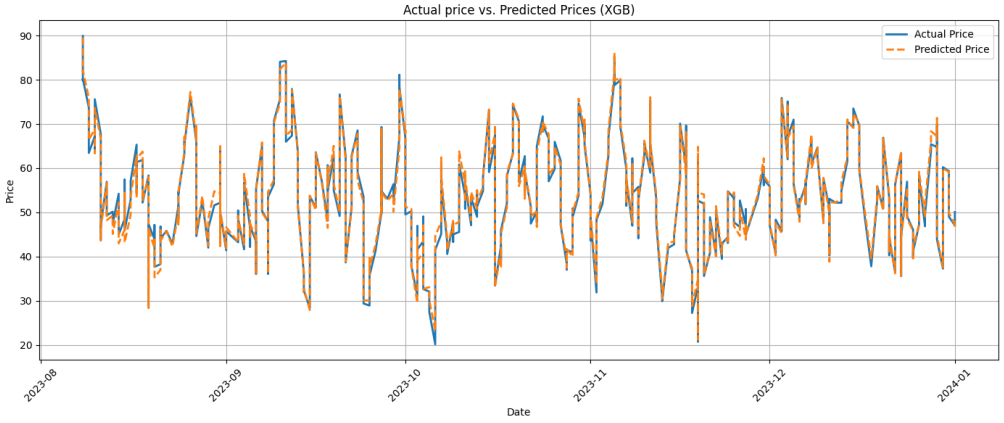
Со вклучување на овие карактеристики му овозможуваме на моделот да ги фати трендовите во цените и побарувачката. Поради новите карактеристики се појавија null вредности во неколку редови, па ги отстранив.

6. XGBoost модел

Избрав XGBoost поради неговата способност да моделира комплексни и нелинеарни односи, како и поради вградената регуларизација.

Датасетот го поделив со train_test_split (shuffle=False) за да се задржи временскиот редослед и да се избегне истекување на информации. Користев XGBRegressor, бидејќи се работи за регресивен проблем.

Резултати:
MAE = 2.4064 → просечно отстапување од реалните цени
RMSE = 2.8063 → ретки големи отстапувања
R^2 = 0.9883 → 98,83% од варијансата објаснета
Моделот покажа висока точност во предвидување на цените врз основа на избраните карактеристики.



За да прикажам уште појасна слика за перформансите на моделот со текот на времето, на реалните и на предвидените цени применив подвижна средна вредност со прозорец од 5. Добиениот график го илустрира усогласувањето на предвидените трендови со вистинските варијации на цените со текот на времето низ примероците на датуми. Оваа визуелизација нагласува дека моделот не само што предвидува точни вредности на цените туку и успешно ги доловува временските трендови и шеми својствени за податоците.

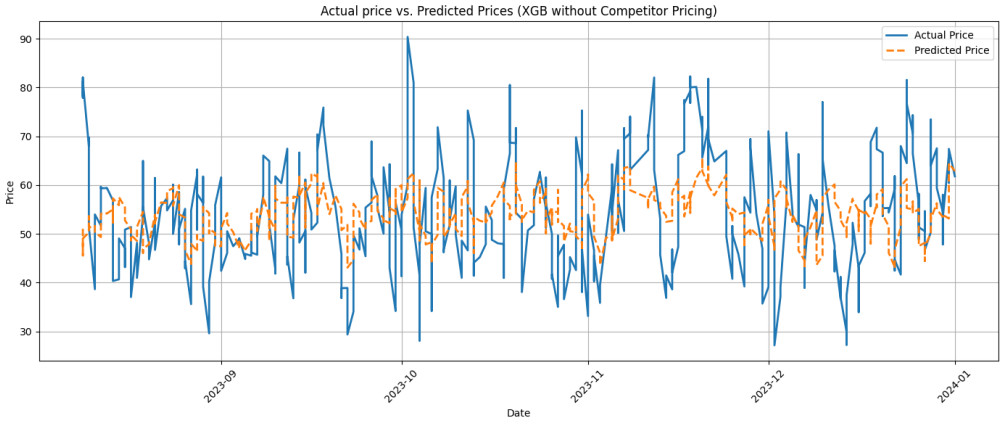
Анализа на моделот со и без Competitor Pricing

Поради големата корелација на Price со Competitor Pricing се решив да следниот мој чекор биде тренирање и евалуација на моделот без Competitor Pricing со цел да визуелно прикажам колкав удел има тој податок во обликувањето на предвидените цени. Го искористив истиот начин на поделување и тренирање на податочното множество како и во претходниот модел. Овој експериментален пристап овозможи директна споредба на предвидливата точност и однесувањето на моделот со и без оваа клучна карактеристика.

Како резултат добив:
-Mean Absolute Error (MAE): 20.39
-Root Mean Squared Error (RMSE): 24.11
-R^2 Score: 0.14

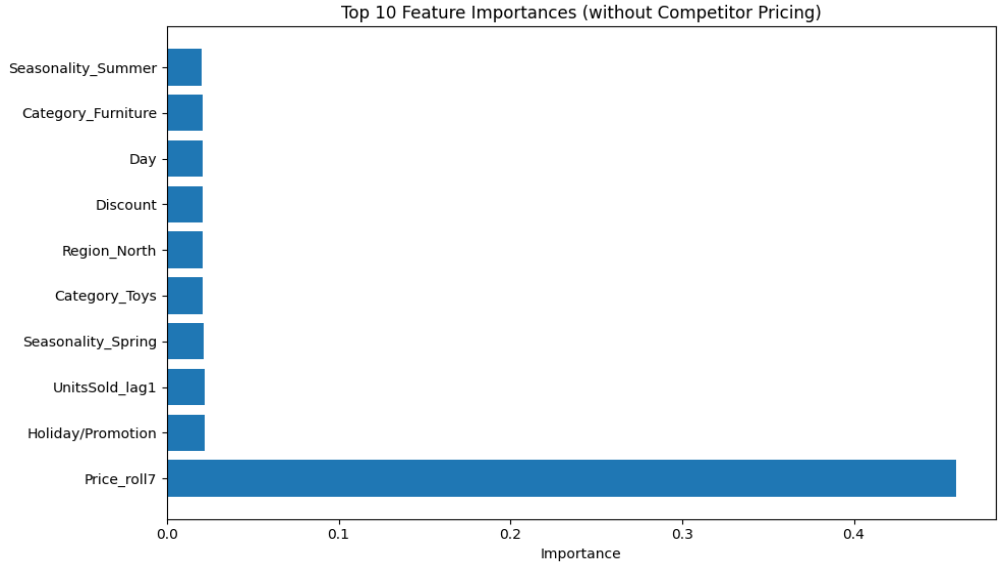
Анализата на важноста на карактеристиките покажа дека Competitor Pricing е една од највлијателните карактеристики во моделот. Резултатите покажаа значителен пад на перформансите, при што MAE се зголеми за повеќе од 8 пати, а R^2 падна од 0,98 на 0,14.

Дијаграм на подвижна средна линија беше искористен за да се илустрира како отсуството на карактеристика влијаеше врз трендовите на предвидување.



Визуелизацијата го покажа истото односно потврди дека Competitor Pricing е клучна за точно предвидување на цените на производите.
Моделот без оваа карактеристика не успеа да ја опфати основната динамика на цените, истакнувајќи дека цените на конкурентите играат доминантна улога во одредувањето на цените на производите во ова податочно множество.

Исто така направив уште една анализа на важноста на карактеристиките на моделот без Competitor Pricing со која открив дека Price_roll7 стана доминантна карактеристика придонесувајќи значително во тежината на важноста. Другите карактеристики, како што се Holiday/Promotion, UnitsSold_lag1 и други покажаа минимална важност, што укажува дека не можат да компензираат за отсуството на информации за цените на конкурентите.



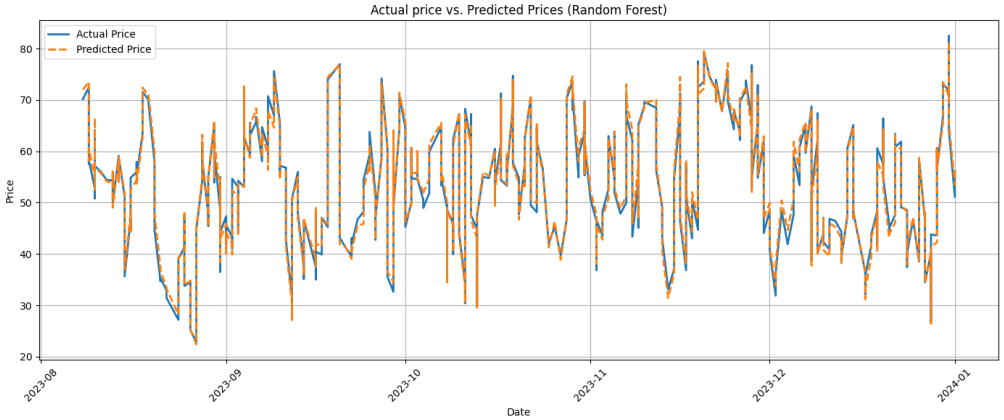
Ова исто така го нагласува силното влијание на пазарната конкуренција врз динамиката на цените на производите. Отстранувањето на тој надворешен пазарен сигнал го принуди моделот да се потпира на послаби внатрешни шеми, што доведе до значително полоши перформанси.

7. RandomForest модел

Првичниот XGBoost модел покажа одлични резултати, но се решив да го истренирам податочното множество и со RandomForest модел со цел да ги споредам нивните резултати. И овој модел покажа одлични перформанси.

-Mean Absolute Error (MAE): 2.4315
-Root Mean Squared Error (RMSE): 2.8452
-R^2 Score: 0.9880

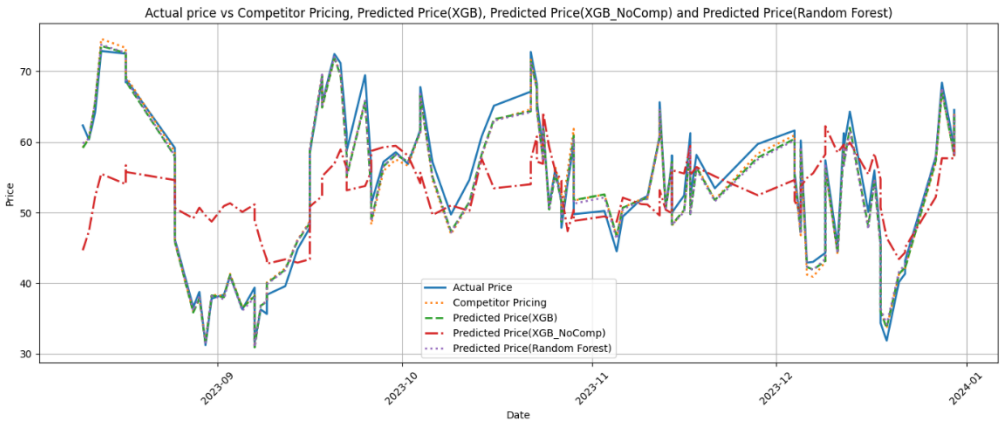
Визуелизацијата го потврдува тоа



И двата моделите постигнаа одлични перформанси, при што XGBoost даде минимално подобри резултати од RandomForest моделот во сите клучни метрики. Овие резултати се во согласност со очекувањата, со оглед на тоа што XGBoost обично се истакнува во фаќањето на сложените интеракции на карактеристиките и поефикасното справување со регуларизацијата.

8. Заклучок и крајни визуелизации

За да ги дополнам метриките за квантитативни евалуации, генерирав визуелизација со 100 случајни примероци каде со подвижен просек ги споредив реалните цени, цените на конкурентите и предвидувањата на трите модели.



Оваа визуелизација потврдува дека вклучувањето на цените на конкурентите значително го подобрува предвидливото усогласување со реалните цени, додека нивното изоставување доведе до видливо поголеми отстапувања. Моделот XGBoost се покажа како најдобар, внимателно следејќи ги реалните цени во повеќето периоди, додека Random Forest покажа споредливо, но за малку послабо следење.

Крајни размислувања

Овој проект го истражуваше потенцијалот на динамичното одредување на цени базирано на машинско учење за да им помогне на трговците на мало да донесуваат подобри одлуки за ценообразување кои се базирани на податоци. Статичките стратегии често не се доволни на динамичните пазари каде што фактори како залихи и конкуренција имаат големо влијание.

Со примена на инженерство на карактеристики, регресивно моделирање и евалуација, развив предиктивни модели за оптимални цени. Анализата покажа дека ценообразувањето на конкурентите е клучен надворешен фактор што значително ја подобрува точноста на моделот. Без тие податоци, перформансите значајно се намалуваат.

XGBoost постигна најдобри резултати, а Random Forest исто така покажа солидни перформанси. Сепак, без целосни пазарни податоци, дури и најдобрите алгоритми имаат ограничувања.

Визуелизациите на реалните и предвидените цени ја покажаа практичната вредност на моделите, особено кога се обезбедени со релевантни податоци.

Севкупно, овој проект успешно демонстрираше дека машинското учење може да биде моќна алатка за динамично одредување на цени, но успехот зависи од квалитетот и комплетноста на податоците. Со таков пристап, трговците можат да градат поприлагодливи и попрофитабилни стратегии.