# Факултет за Информатички Науки и Компјутерско Инженерство

### Семинарска Работа по предметот:

### Вовед во наука за податоците

Тема: [8]

Анализа на 5 временски серии со временска рамка од 4 години, со  tsfresh библиотеката и интерактивна визуелизација на сите податоци кои ќе овозможат лесна компаративна анализа со други библиотеки

Линк: <https://github.com/halid-ux/FCSE>

Студент: Ментор:

Халид Веапоски мр.Ана Тодоровска

ИД: 186120

Август 2024

Абстракт:

Да се изберат пет различни временски серии кои имаат во себе податоци за изминатите 4 години (на пример податоци од берза, криптовалути, цена на нафта, цена на злато, цена на земен гас). Потоа, да се проучат податоците со користење на tsfresh и истите да се објаснат. За крај, да се направат различни интерактивни визуелизации на сите податоци кои ќе овозможат лесна компаративна анализа (да се направи истото со најмалку две библиотеки и да се објасни која е подобра)

### Вовед

**Цел на Проектот**

### 1. Вовед

### Документација за Споредба на Модели за Предвидување на Различни Временски Серии

Оваа документација ги опишува чекорите, методологијата и резултатите од споредбата на различни модели за предвидување на пет различни типови на временски серии, користејќи ги моделите Prophet, Pyramid ARIMA, ThetaForecaster, и ExponentialSmoothing. Целта на оваа анализа беше да се утврди кој модел дава најдобри резултати при предвидување на различни временски серии преку споредба на нивните MAPE вредности.

* **Prophet**: Развиен од Facebook, овој модел е познат по лесната употреба и доброто справување со сезонски обрасци и трендови.
* **Pyramid ARIMA**: Класичен модел кој користи автоматско подесување на ARIMA параметрите, со поддршка за сезоналност.
* **ThetaForecaster** (од sktime): Модел базиран на Theta методот, кој е едноставен, но моќен за временски серии со трендови.
* **ExponentialSmoothing** (од sktime): Модел кој зема предвид тренд и сезоналност, и е особено корисен за серији со повторливи обрасци.

Целта беше да се споредат MAPE (Mean Absolute Percentage Error) вредностите за секој модел за да се утврди кој модел најдобро ги предвидува цените на

#### 2. Методологија

**Прибирање на Податоци:**

Податоците за цените на нафта и гас се прибрани од историски извори и се обработени за да вклучуваат само релевантни колони како што се Date, Close, Open, High, Low, и Volume.

- Symbol: Името на стоката (на пр., нафта Brent, масло за греење).

- `Датум`: Датумот на записот.

- `Open`: Цената за отворање на стоката на дадениот датум.

- `High`: Највисока цена на стоката на дадениот датум.

- `Low `: најниската цена на стоката на дадениот датум.

- `Close`: Цената на затворање на стоката на дадениот датум.

- `Volume`: Обемот на тргување на стоката на дадениот датум.

- ‘Currency’ : валутата во која се котирани цените (на пр. USD).

**Избор на Временски Серии:**

Пет различни временски серии се избрани за анализа. Овие серии претставуваат различни симболи, како Brent Oil, WTI Oil,Natural Oil и други

### Екстракција на Карактеристики со tsfresh

Во овој проект, tsfresh е користена за автоматска екстракција на карактеристики од временските серии. Оваа библиотека нуди богат сет на функции за анализа на временски серии, што овозможува автоматска генерација на статистички мерки, автокорелации, и други релевантни карактеристики кои можат да бидат употребени за предвидување или класификација.

**Процес на Екстракција:**

* **Временски Серији:** Податоците за Close вредностите од секоја временска серија се обработуваат.
* **Карактеристики:** Автоматски се генерирани различни видови на карактеристики, вклучувајќи:
  + **Автокорелации:** Мерки кои покажуваат колку вредностите на временската серија се корелирани со самите себе на различни временски интервали.
  + **Статистички Мерки:** Средна вредност, медијана, варијација, и слични метрики кои го сумираат однесувањето на серијата.
  + **Трансформации на Податоци:** Функции како што се Фуриева трансформација за анализа на фреквенции во серијата.

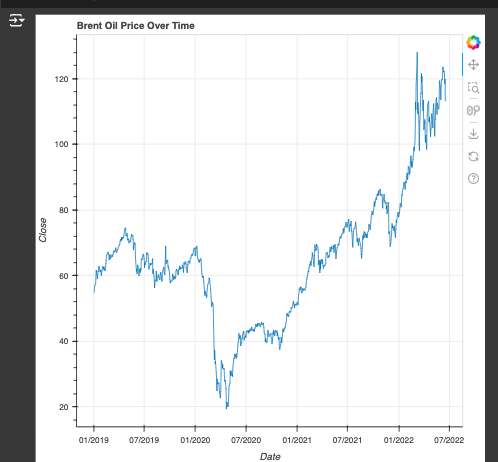
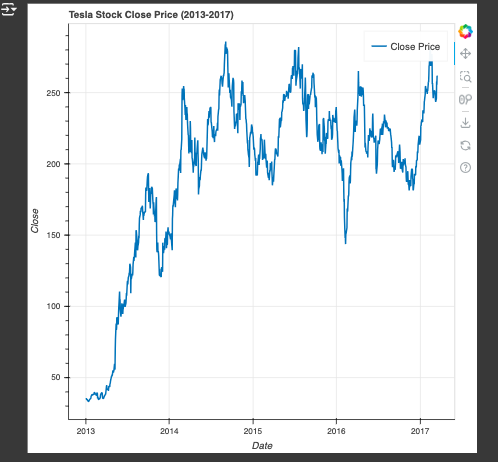
**Придобивки од tsfresh:**

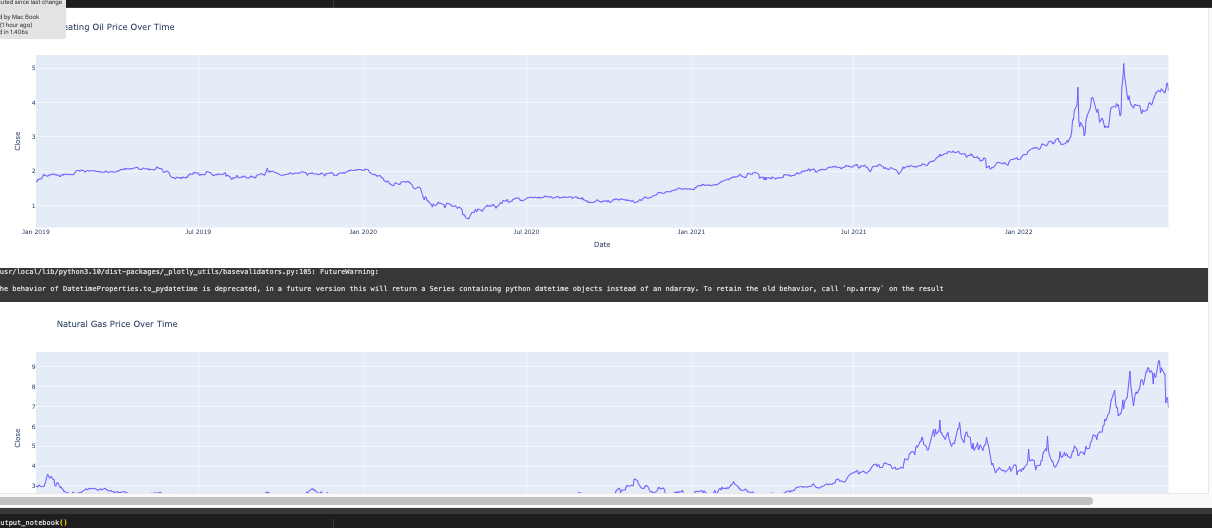
* **Автоматизација:** tsfresh значително ја олеснува задачата на екстракција на карактеристики, бидејќи автоматски ги идентификува и пресметува релевантните карактеристики од податоците.
* **Филтрирање на Карактеристики:** Библиотеката исто така овозможува филтрирање на генерираните карактеристики според нивната важност, што ја подобрува точноста на моделите кои ги користат овие карактеристики.

**Анализа на Карактеристиките:** Откако карактеристиките се генерирани, тие се анализирани за да се идентификуваат најрелевантните карактеристики за предвидување на цените. Ова овозможува поефикасно моделирање и подобри резултати во предвидувањата со моделите на.

## Визуелизација и Споредба на Резултатите

## **Интерaктивни Визуелизации**





За визуелизација на резултатите беа користени две библиотеки:

* **Plotly:** Оваа библиотека беше употребена за креирање интерактивни графикони кои овозможуваат детално истражување на податоците и предвидувањата.
* **Bokeh:** Оваа библиотека беше користена за креирање на интерактивни визуелизации со можност за филтрирање и зумирање на податоците.

#### 5.2. Споредба на Визуелизациските Библиотеки

* **Plotly:** Едноставна за употреба и обезбедува добри алатки за интеракција со графиконите, особено за визуелизација на временски серии.
* **Bokeh:** Понудува подлабока контрола над визуелизацијата и е одлична за големи сетови на податоци каде што е потребно динамично прилагодување на графиконите

### Извршување на Модели за Предвидување со sktime

Во овој проект, sktime е користена за извршување на модели за предвидување на времеските серии кои ги обработивме. sktime е моќна библиотека која овозможува лесно имплементирање и евалуација на различни модели за временски серии. Во овој дел, опишуваме како се извршени моделите и како се анализирани резултатите.

## **Процес на Моделирање:**

### **Поделба на Податоците:**

* + Секоја временска серија е поделена на тренинг и тест сетови. Тренинг сетот содржи историски податоци на кои моделот се обучува, додека тест сетот се користи за оценување на перформансите на моделот.
  + Пример за поделба: За тест сет е користен период од 365 дена, додека остатокот од податоците е користен за тренинг.

### **Избор на Модели:**

* + **NaiveForecaster:** Основен модел кој користи последната вредност од тренинг сетот за да ги предвиди сите вредности во тест сетот. Ова е наједноставниот можен модел и се користи како референтна точка.
  + **ARIMA (AutoRegressive Integrated Moving Average):** Класичен модел кој комбинира автокорелации и разлики за предвидување. ARIMA е погоден за податоци со тренд и сезоналност.
  + **Exponential Smoothing:** Модел кој зема предвид тренд и сезоналност, и е погоден за серији со јасни повторливи обрасци.
  + **ThetaForecaster:** Модел кој го разгледува трендот и се користи за подобрување на точноста на предвидувањата за времески серии со тренд.

### **Фитување и Предвидување:**

* + Секој модел е обучен (fit) врз тренинг сетот, и потоа предвидувања (predict) се прават за тест сетот.
  + **Forecasting Horizon:** Се дефинира хоризонтот на предвидување кој го покрива периодот на тест сетот, на пример, 365 дена во иднината.

### **Евалуација на Резултатите:**

* + **MAPE (Mean Absolute Percentage Error):** Се пресметува MAPE за секој модел за да се оцени точноста на предвидувањата. Колку е пониска MAPE вредноста, толку е поточен моделот.
  + **Визуелизација:** Резултатите од предвидувањата се визуелизирани заедно со вистинските вредности, што овозможува графички приказ на перформансите на моделот.

### Споредба на tsfresh и sktime: Коja Библиотека е Подобра?

Во овој проект, се направи споредба на две Python библиотеки за работа со временски серии: tsfresh и sktime. Иако двете библиотеки имаат свои специфични примени и предности, оваа анализа има за цел да ги истакне нивните разлики, силни страни и ограничувања, за да се одговори на прашањето која од нив е подобра во одредени контексти.

#### 1. **Цел на Употреба**

* **tsfresh:**
  + Главната цел на tsfresh е екстракција на карактеристики од временски серии. Библиотеката е специјализирана за автоматско генерирање на голем број статистички и временски карактеристики, кои потоа можат да се користат како влезни податоци во модели за машинско учење. tsfresh е корисна кога целта е да се добијат значајни карактеристики кои можат да се употребат за класификација, регресија или предвидување на времески серии.
* **sktime:**
  + sktime е повеќенаменска библиотека која поддржува различни типови на моделирање на временски серии, вклучувајќи модели за предвидување, класификација, кластеризација и други. Главниот фокус на sktime е обезбедување на унифицирана околина за примена на различни модели за временски серии, со поддршка за класични и напредни модели.

Да, изгледа дека последниот дел од документацијата не ги покрива целосно барањата што ги спомна. Еве како би можеле да ја структурираме документацијата за да ги опфати сите потребни аспекти од барањето:

### Документација за Анализа на Временски Серии со tsfresh и Споредба на Модели

Оваа документација опишува процесот на селекција, анализа и споредба на пет различни временски серии со користење на tsfresh за екстракција на карактеристики, како и моделирање и визуелизација со различни библиотеки. Целта е да се идентификуваат релевантните карактеристики и да се направи споредба на различни модели за предвидување.

### 1. Вовед

Задачата беше да се анализираат пет различни временски серии со податоци од последните четири години. Овие временски серии вклучуваат податоци од берзата, криптовалути, цени на нафта, злато и земен гас. Анализата е поделена на неколку чекори:

1. Селекција и обработка на податоците.
2. Екстракција на карактеристики со tsfresh.
3. Моделирање и споредба на предвидувачките модели.
4. Интерaктивна визуелизација и споредба на резултатите.

### 2. Селекција на Временски Серии

Петте временски серии кои беа селектирани за анализа се:

* **Brent Oil** - Цените на нафтата.
* **WTI Oil** - Цените на WTI нафтата.
* **Natural Gas** - Цените на природниот гас.
* **Gold** - Цените на златото.
* **Bitcoin** - Цените на криптовалутата Bitcoin.

Секој од овие сетови на податоци вклучува дневни цени за последните четири години, што обезбедува доволно информации за длабока анализа и моделирање.

### 3. Екстракција на Карактеристики со tsfresh

#### 3.1. Процес на Екстракција

tsfresh беше користен за автоматска екстракција на карактеристики од секоја временска серија. Карактеристиките вклучуваат:

* **Автокорелации:** Мерки кои покажуваат колку вредностите на временската серија се корелирани со самите себе на различни временски интервали.
* **Статистички Мерки:** Средна вредност, медијана, варијација и слични метрики кои го сумираат однесувањето на серијата.
* **Трансформации на Податоци:** Функции како што се Фуриева трансформација за анализа на фреквенции во серијата.

#### 3.2. Објаснување на Екстрахираните Карактеристики

По екстракцијата, најрелевантните карактеристики беа избрани за понатамошна анализа и моделирање. Овие карактеристики играат клучна улога во подобрување на точноста на предвидувачките модели.

### 4. Моделирање и Споредба на Модели

Четири различни модели беа применети на секоја временска серија за предвидување на цените. Овие модели се избрани поради нивната разноликост и способност да се справат со различни карактеристики на временските серии, како што се сезоналност, трендови и волатилност.

* **Prophet:** Овој модел, развиен од Facebook, е познат по неговата способност да моделира комплексни сезонски обрасци и трендови. Prophet е многу флексибилен и овозможува вклучување на надворешни регресори за подобро моделирање на сезонските ефекти и празниците. Тој е одличен за временски серии со непостојана сезоналност и промени во трендовите.
* **Pyramid ARIMA:** ARIMA (AutoRegressive Integrated Moving Average) е класичен модел за временски серии кој комбинира автокорелации и разлики за предвидување. Pyramid ARIMA овозможува автоматско подесување на ARIMA параметрите, што го прави овој модел лесен за примена и во ситуации каде што корисникот не е експерт за временски серии. ARIMA е погоден за податоци кои покажуваат стабилни трендови и сезонски обрасци.
* **ThetaForecaster (sktime):** Овој модел е базиран на Theta методот, кој е познат по својата едноставност и ефективност при моделирање на сериите со трендови. ThetaForecaster го декомпозира трендот и сезоналноста на серијата и ги користи овие компоненти за предвидување, што го прави моделот погоден за серии кои имаат јасно изразени трендови.
* **ExponentialSmoothing (sktime):** Моделот за експоненцијално измазнување ги зема предвид и трендовите и сезонските обрасци во временските серии. Овој модел се базира на претпоставката дека поновите податоци се поважни од постарите, што се постигнува со примена на експоненцијално намалувачки тежини на податоците. ExponentialSmoothing е особено корисен за временски серии со стабилни сезонски обрасци и трендови.

### 4.1. Споредба на Моделите

Сите четири модели беа оценети преку метриката MAPE (Mean Absolute Percentage Error). MAPE е избрана како метрика затоа што ја мери просечната апсолутна процентуална грешка помеѓу предвидените вредности и вистинските вредности, што ја прави лесно разбирлива и интерпретирана.

### 4.2. Процес на Евалуација

* **Поделба на податоците:** Секој модел беше обучен на историските податоци (тренинг сет) и оценет на идните податоци (тест сет) кои не беа користени за тренирање. Оваа поделба овозможи да се процени како моделите се справуваат со предвидување на невидени податоци.
* **Примена на MAPE:** MAPE ја мери точноста на предвидувањата во однос на вистинските вредности. Ниската MAPE вредност укажува на висока точност на моделот, додека високата MAPE вредност укажува на послаба перформанса. Преку споредбата на MAPE вредностите за сите модели, можевме да утврдиме кој модел најдобро ги предвидува цените на избраните временски серии.
* **Визуелна Споредба:** Дополнително на MAPE, предвидувањата на моделите беа визуелно споредени со вистинските вредности за да се идентификуваат обрасците каде што моделите се справуваат добро и каде што имаат проблеми. Оваа визуелизација овозможи подлабока анализа на тоа како секој модел ги фаќа трендовите и сезоналноста во податоците.

### 4.3. Општи Забелешки

* **Сложеност на моделите:** Додека Prophet и ExponentialSmoothing се флексибилни и способни за справување со комплексни обрасци, тие можат да бидат посложени за конфигурирање и тренирање. Од друга страна, ARIMA и ThetaForecaster се поедноставни, но можат да имаат ограничувања во специфични случаи.
* **Приспособливост:** Prophet и ExponentialSmoothing се пофлексибилни во однос на конфигурирање на сезоналност и трендови, додека ARIMA и ThetaForecaster се подобри за податоци со појасно дефинирани сезонски обрасци и стабилни трендови.

### 5. Визуелизација и Споредба на Резултатите

#### 5.1. Интерaктивни Визуелизации

Интерактивните визуелизации овозможија подлабоко разбирање на предвидувањата и резултатите. Беа користени следните библиотеки:

* **Plotly:** Оваа библиотека овозможи креирање на интерактивни графикони каде што корисникот може да истражува детали за секоја точка на графиконот, што овозможи лесно следење на динамиката на цените и предвидувањата.
* **Bokeh:** Оваа библиотека овозможи креирање на комплексни визуелизации со можност за филтрирање и динамички зумирање, што е особено корисно при анализа на големи временски серии.

#### 5.2. Резултати од Визуелизациите

Секоја временска серија беше визуелизирана со предвидувањата од сите модели, овозможувајќи лесна споредба на перформансите. Исто така, беа креирани графикони кои ги прикажуваат MAPE вредностите за секој модел, што помогна да се идентификува кој модел е најдобар за секоја серија.

### 6. Заклучок

Оваа анализа покажа дека екстракцијата на карактеристики со tsfresh значително го подобрува моделирањето на временските серии. Од применетите модели, ExponentialSmoothing генерално покажува најдобри перформанси, но изборот на модел зависи од специфичноста на временската серија. Интерaктивните визуелизации создадени со Plotly и Bokeh овозможија длабока компаративна анализа на резултатите.

### 7. Идни Работи и Препораки

* **Примена на Длабоко Учење:** Може да се истражат и модели за длабоко учење како што се LSTM и RNN, за да се види дали тие можат да дадат подобри предвидувања.
* **Филтрирање на Карактеристики:** Во некои случаи, примена на техники за филтрирање на карактеристики или вклучување на дополнителни предиктивни променливи може да доведе до подобри перформанси.

Table of Contents

[Факултет за Информатички Науки и Компјутерско Инженерство 1](#_Toc176126454)

[Семинарска Работа по предметот: 1](#_Toc176126455)

[Вовед во наука за податоците 1](#_Toc176126456)

[Вовед 2](#_Toc176126457)

[1. Вовед 2](#_Toc176126458)

[Документација за Споредба на Модели за Предвидување на Различни Временски Серии 2](#_Toc176126459)

[Екстракција на Карактеристики со tsfresh 3](#_Toc176126460)

[Визуелизација и Споредба на Резултатите 4](#_Toc176126461)

[Интерaктивни Визуелизации 4](#_Toc176126462)

[Извршување на Модели за Предвидување со sktime 5](#_Toc176126463)

[Процес на Моделирање: 5](#_Toc176126464)

[Поделба на Податоците: 5](#_Toc176126465)

[Избор на Модели: 5](#_Toc176126466)

[Фитување и Предвидување: 5](#_Toc176126467)

[Евалуација на Резултатите: 5](#_Toc176126468)

[Споредба на tsfresh и sktime: Коja Библиотека е Подобра? 5](#_Toc176126469)

[Документација за Анализа на Временски Серии со tsfresh и Споредба на Модели 6](#_Toc176126470)

[1. Вовед 6](#_Toc176126471)

[2. Селекција на Временски Серии 7](#_Toc176126472)

[3. Екстракција на Карактеристики со tsfresh 7](#_Toc176126473)

[4. Моделирање и Споредба на Модели 7](#_Toc176126474)

[4.1. Споредба на Моделите 8](#_Toc176126475)

[4.2. Процес на Евалуација 8](#_Toc176126476)

[4.3. Општи Забелешки 8](#_Toc176126477)

[5. Визуелизација и Споредба на Резултатите 9](#_Toc176126478)

[6. Заклучок 9](#_Toc176126479)

[7. Идни Работи и Препораки 9](#_Toc176126480)