МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра

автоматизованих систем обробки інформації та управління

(повна назва кафедри, циклової комісії)

**КУРСОВА РОБОТА**

з баз данних - 1

(назва дисципліни)

на тему: «Прогнозування потоку відвідувачів у ресторані. Кількість заброньованих місць.»

Студента II курсу, групи ІП-96

Бурятова Олексія Олексійовича

Спеціальності 121 «Інженерія програмного забезпечення»

Керівник ст. викладач Олійник Ю.О.

(посада, вчене звання, науковий ступінь, прізвище та ініціали)

Кількість балів: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Національна оцінка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| Члени комісії |  |  |  |
|  | (підпис) |  | (вчене звання, науковий ступінь, прізвище та ініціали) |
|  |  |  |  |
|  | (підпис) |  | (вчене звання, науковий ступінь, прізвище та ініціали) |

Київ-2021

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

(назва вищого навчального закладу)

## Кафедра автоматизованих систем обробки інформації і управління

Дисципліна Аналіз даних в інформаційно-управляючих системах

Напрям "Програмна інженерія"

Курс 2 Група ІП-96 Семестр 4

### **ЗАВДАННЯ**

#### на курсову роботу студента

##### Бурятова Олексія Олексійовича

(прізвище, ім’я, по батькові)

1. Тема роботи

2. Строк здачі студентом закінченої роботи

3. Вихідні дані до роботи

4. Зміст розрахунково-пояснювальної записки (перелік питань, які підлягають розробці)

5. Перелік графічного матеріалу ( з точним зазначенням обов’язкових креслень )

6. Дата видачі завдання

ЗМІСТ

[Кафедра автоматизованих систем обробки інформації і управління 2](#_Toc72749413)

[**ЗАВДАННЯ** 2](#_Toc72749414)

[**Вступ** 4](#_Toc72749415)

[**1** **ПЕРЕДМОВА** 5](#_Toc72749416)

[**2** **Теоретичні відомості** 6](#_Toc72749417)

[**3** **Опис програмного забезпечення** 10](#_Toc72749418)

[3.1. ER Діаграма 10](#_Toc72749419)

[3.2. Опис сутностей 10](#_Toc72749420)

[3.3. Приклади даних 11](#_Toc72749421)

[**4** **Аналіз і узагальнення результатів** 13](#_Toc72749422)

[Додаток А Технічне завдання 22](#_Toc72749423)

[Додаток Б Тексти програмного коду 23](#_Toc72749424)

[Перелік посилань 47](#_Toc72749425)

# **Вступ**

Пояснювальна записка до курсової роботи: 45 сторінок, 13 рисунків, 2 таблиці, 2 посилання.

Об’єкт дослідження: дослідження робити зі сховищами даними, обробкою даних та їх аналіз.

Мета роботи: спрогнозувати потік відвідувачів у ресторані. Визначити кількість заброньованих місць..

Вивчено синтаксис реляційної бази даних MSSQL та мови програмування R. Приведені змістовні приклади даних у кожній з таблиць бази даних.Проаналізовано періодичність відвідування ресторану та спрогнозовано кількість відвідувачів у майбутному. Був проведений аналіз та робота над помилками.

# **ПЕРЕДМОВА**

Дана курсова робота була призначена для поглиблення та закріплення теоретичних знань студентів, що навчаються за дисципліною «Аналіз даних в інформаційно-управляючих системах». Написання курсової роботи дає студентові можливість навчитися самостійно і творчо використовувати наукові джерела та узагальнювати теоретичні положення.

# **Теоретичні відомості**

Різновидом баз даних є сховище даних (Data WarenHouse). Поняття сховищ даних виникло зовсім недавно. Необхідність розробки нової концепції сховищ даних обумовлена такими факторами:

* Розвиток інформаційних технологій привів до систем нового типу, які дістали назву систем підтримки прийняття рішень. Ці системи основані на новій технологіі - OLAP-технології. Основою OLAP-технології є реалізація аналітичних запитів.
* Системи підтримки прийняття рішень, основані на формуванні аналітичних запитів, почали конфліктувати з транзакційними системами оперативної обробки даних (OLTP- системами). Одночасне вирішення оперативних і аналітичних запитів на одній базі даних часто призводить до нестачі ресурсів.
* Формування аналітичних звітів на основі традиційних баз даних, які вміщують оперативну інформацію, займає дуже багато часу. Це призводить до того, що менеджери не встигають готувати відповідні рішення на основі отриманих аналітичних звітів.
* Дуже часто на підприємстві чи в організації функціонує декілька OLTP-систем, кожна з яких має свою окрему базу даних, в яких використовуються різні структури даних, способи кодування, одиниці вимірювання. Побудова зведеного аналітичного запиту на основі декількох баз даних є дуже складною проблемою, яка спочатку потребує вирішення проблеми узгодженності даних, що зберігаються в різних базах даних.

**Сховища даних** характеризуються предметною орієнтацією, інтегрованістю, підтримкою хронології, незмінністю і мінімальною надлишковістю

**Предметна орієнтація.** Дані в сховищі даних організовані у відповідності до основних напрямків діяльності підприємства чи фірми (замовники, продажі, склад і т.п.).

**Інтегрованість.** Первинні дані оперативних баз даних перевіряються, певним чином добираються, приводяться до одного виду, необхідною мірою агрегуються ( тобто обраховуються сумарні показники) і завантажуються у сховище даних.

**Підтримка хронології.**Це дозволяє проводити аналіз зміни показників у часі.

**Незмінність.** Це є суттєвою відмінністю даних, що зберігаються у сховища даних, від оперативних даних. Оперативні дані можуть дуже часто змінюватись, з даними сховища можливі лише операції їх первинного завантаження, пошуку та їх читання.

**Мінімальна надлишковість.** Забезпечується тим, що перш ніж завантажувати дані до сховищ, їх фільтрують і певним чином очищають від таких даних, які не потрібні і не можуть бути використані в OLAP-системах.

* 1. Створення ER діаграми

ER-модель — це мета-модель даних, тобто засіб опису моделей даних. Існує ряд моделей для представлення знань, але одним з найзручніших інструментів уніфікованого представлення даних, незалежного від програмного забезпечення що його реалізує, є модель «сутність-зв'язок». Важливим є той факт, що з моделі «сутність-зв'язок» можуть бути породжені всі існуючі моделі даних (ієрархічна, мережева, реляційна, об'єктна), тому вона є найзагальнішою.

Модель сутність-зв'язок є результатом систематичного процесу, який описує та визначає деяку предметну область. Вона не визначає сам процес, а лише візуалізує його. Дані представлені у вигляді компонентів (сутностей), які пов'язані між собою певними зв'язками, які виражають залежності і вимоги між ними, такі як: одна будівля може бути розділена на нуль або більше квартир, але одна квартира може бути розташована лише в одній будівлі. Сутності можуть мати різні властивості (атрибути), які характеризують їх. Діаграми, створені для представлення цих сутностей, атрибутів і зв'язків графічно, називають сутність-зв'язок діаграмами.

ER-модель зазвичай реалізується в вигляді баз даних. У разі реляційної бази даних, в якій зберігаються дані в таблицях, кожен рядок кожної таблиці являє собою один екземпляр сутності. Деякі поля даних в цих таблицях вказують на індекси в інших таблицях. Такі поля є покажчиками фізичної реалізації зв'язків між сутностями.

* 1. Аналіз даних. Прогнозування.

Аналіз даних — це процес виконання послідовних логічних дій з інтерпретації відповіді респондентів та їх перетворення у статистичні форми, необхідні для ухвалення маркетингових та управлінських рішень.

Математичні методи аналізу даних широко використовують при дослідженні різноманітних систем і процесів – природних, технічних, екологічних, економічних, соціальних тощо. З огляду на це формування відповідних знань та навичок є необхідною складовою підготовки фахівців у галузі системних наук і кібернетики, інформатики та багатьох інших галузей знань.

Останнім часом значного поширення набувають нові технології й методи аналізу даних, зокрема методи інтелектуального аналізу даних (data mining), які використовують для виявлення прихованих закономірностей у великих масивах даних, та нейроінформатики, а також методики й засоби статистичного контролю за якістю на виробництві та в управлінні організаціями.

Основні процедури аналізу даних найчастіше реалізують за допомогою сучасних комп’ютерних технологій. При цьому дослідники або самі будують розрахункові алгоритми й пишуть відповідні комп’ютерні програми, або ви- 6 користовують наявне програмне забезпечення – електронні таблиці MS Excel, спеціалізовані пакети SPSS, STATISTICA, математичні пакети загального призначення MatLab, MathCad тощо. Але навіть при застосуванні спеціалізованих пакетів досліднику необхідно володіти теоретичними основами математичних методів аналізу даних, оскільки зазвичай це передбачає необхідність вибору оптимальних алгоритмів та певних параметрів їх реалізації, іноді з декількох сотень можливих варіантів. Це зумовлює необхідність вивчення майбутніми фахівцями основних понять та алгоритмів аналізу даних.

**Прогнозування** (від грецького Prognosis), у широкому розумінні цього слова, визначається як випереджаюче відображення майбутнього.

**Метою** прогнозування є передбачення майбутніх подій.

**Прогнозування** (forecasting) є однією з задач Data Mining і одночасно одним із ключових моментів при прийнятті рішень.

**Прогностика** (prognostics) – теорія й практика прогнозування.

**Прогнозування** спрямоване на визначення тенденцій динаміки конкретного об’єкта або події на основі ретроспективних даних, тобто аналізу його стану колись і тепер. Отже, розв’язок задачі прогнозування вимагає деякої навчальної вибірки даних.

**Прогнозування** – установлення функціональної залежності між залежними й незалежними змінними.

Прогнозування є розповсюдженим і затребуваним завданням у багатьох сферах людської діяльності. У результаті прогнозування зменшується ризик прийняття невірних, необґрунтованих або суб’єктивних рішень.

Приклади задач прогнозування: прогноз руху грошових коштів, прогнозування врожайності агрокультури, прогнозування фінансової стабільності підприємства

Крім економічної й фінансової сфери, задачі прогнозування постають в медицині, фармакології; популярним зараз стає політичне прогнозування.

# **Опис програмного забезпечення**

## ER Діаграма

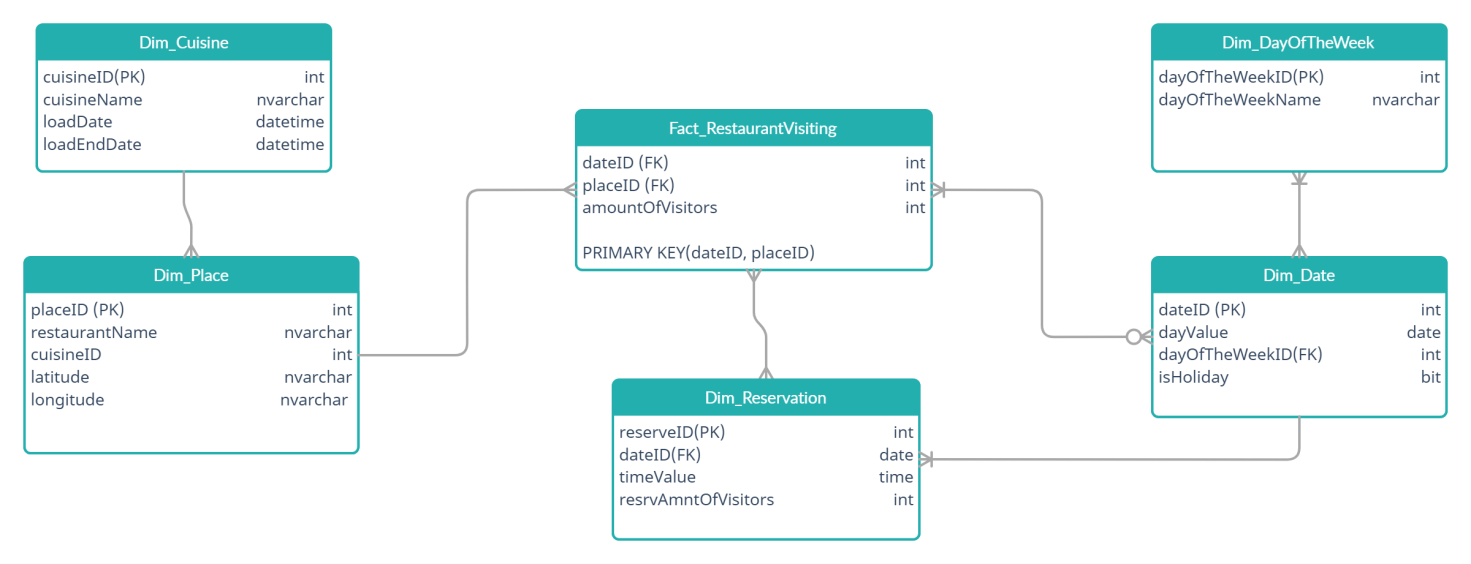


Рисунок 3.1 – ER Діаграма

## Опис сутностей

1. Ресторан
2. Кухня
3. Резерв

## Приклади даних

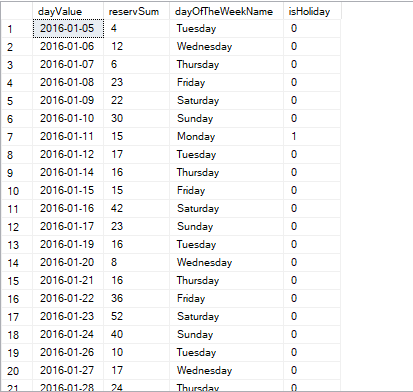


Рисунок 3.1 – Фрагмент даних: кількість зарезервованих місць у конкретний день

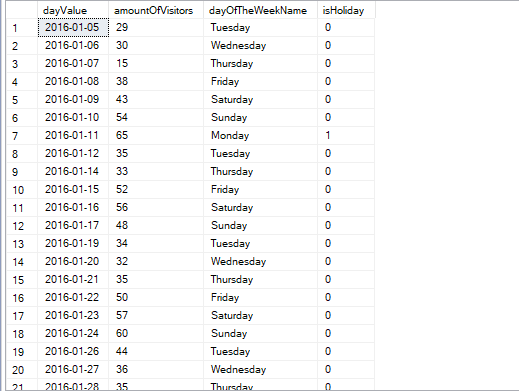


Рисунок 3.2 – Фрагмент даних: кількість відвідувачів у конкретний день

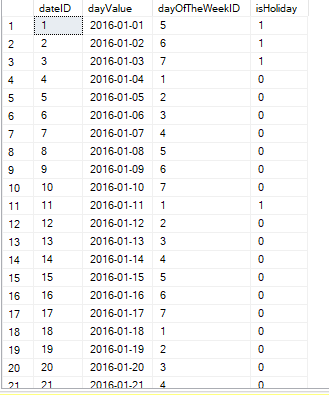


Рисунок 3.3 – Фрагмент даних: характеристики дати

# **Аналіз і узагальнення результатів**

* 1. Графічне зображення даних.

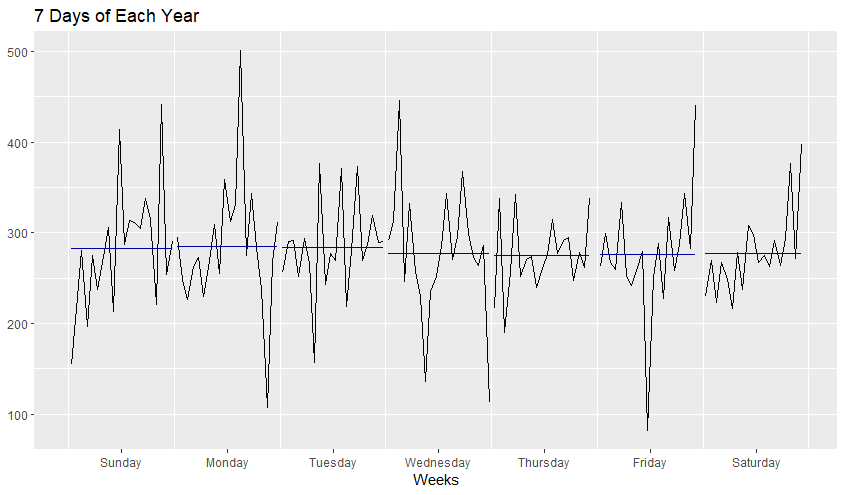


Рисунок 4.1.1 – Графік відображення відвідуваності ресторану по дням тижня

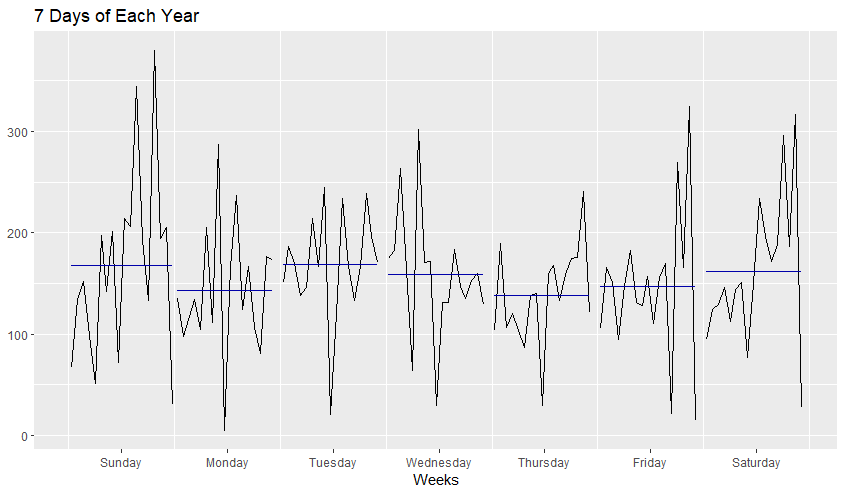


Рисунок 4.1.2 – Графік відображення резервування місць у ресторані по дням тижня

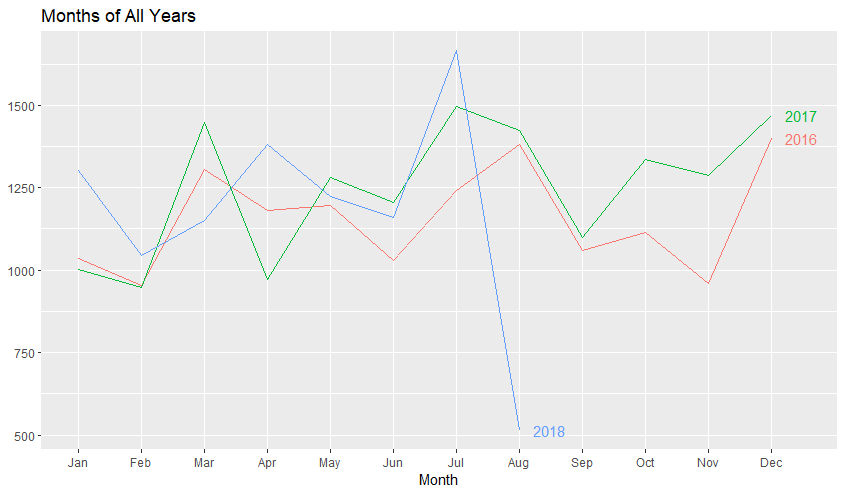


Рисунок 4.1.3 – Графік відображення відвідуваності ресторану за кожен місяць

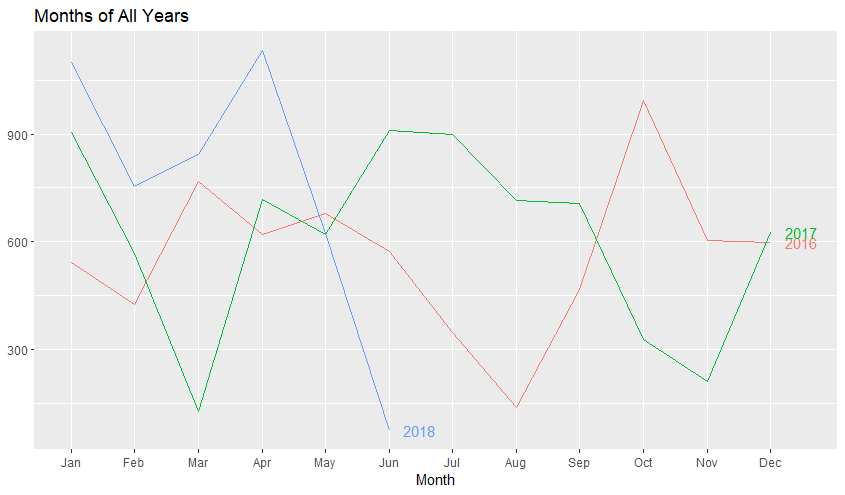


Рисунок 4.1.4 – Графік відображення резервування місць у ресторані за кожен місяць

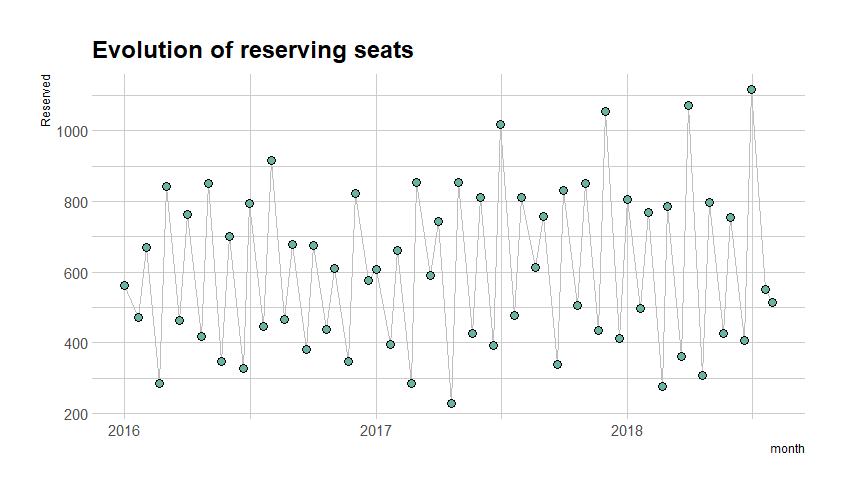


Рисунок 4.1.5 – Графік відображення відвідуваності у ресторані за кожні 20 днів

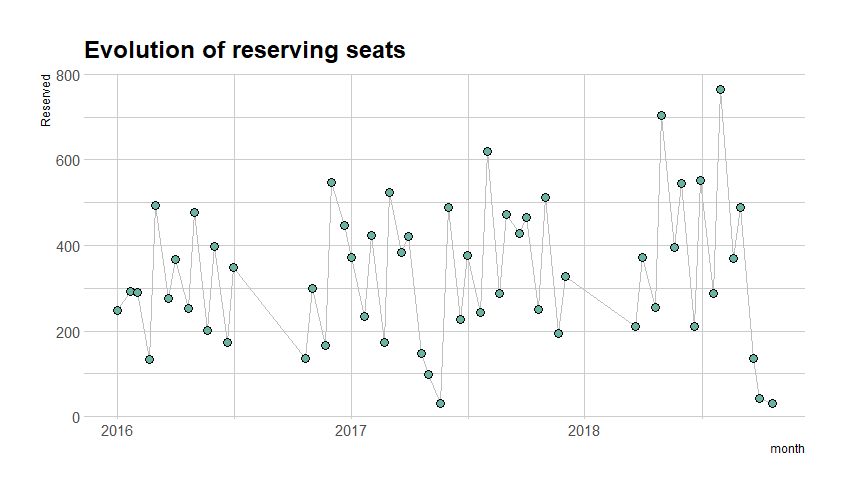


Рисунок 4.1.6 – Графік відображення резервування місць у ресторані за кожні 20 днів

Кожен із цих графіків показує початкові дані. Можна простежити відвідуваність (резервування місць) відвідувачів. Усі графіки побудовані на основі зібраних даних протягом двох із половиною років. Очевидно, що для більш точної та глубокої побудови моделі прогнозування потрібно більший відрізок часу, але для початку та попередніх результатів, цього досить.

Отже, програма починає аналіз даних для резервування.

* 1. Модель прогнозу для датасету резервування.

Здекомпозуємо наш датасет.

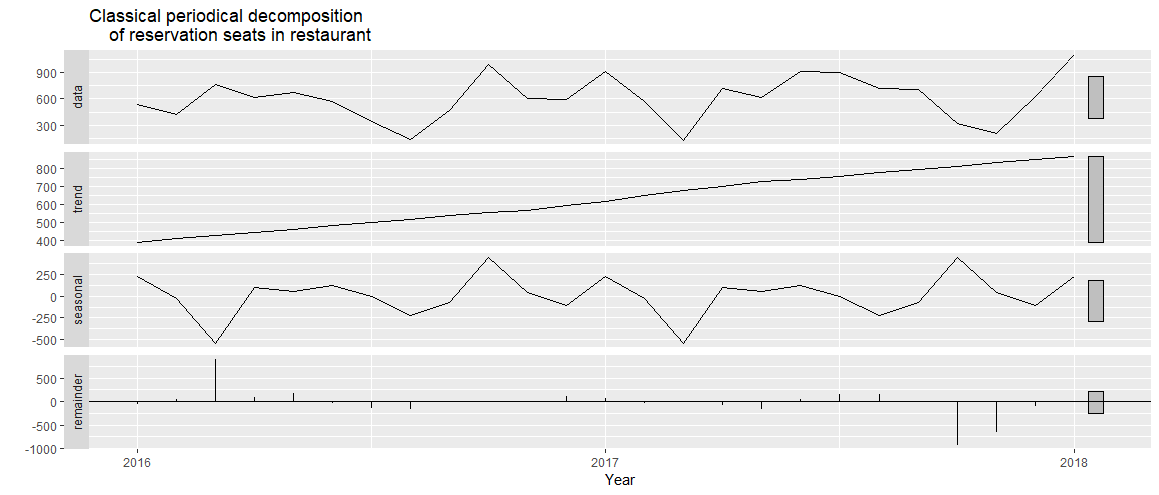


Рисунок 4.2.1 – Графік декомпозиції резервування місць у ресторані

На даному графічному зображенні є 4 графіки.

**Data**- дублює початкові дані (аналогічна діаграма наявна вище).

**Trend**- оцінка трендового циклу має тенденцію до згладжування швидких зростання та падіння даних (як видно з наведеного вище прикладу).

**Seasonal**- щоб оцінити сезонну складову для кожного сезону, потрібно усереднити відхилені значення цього сезону.

**Remainder**- залишок обчислюється відніманням розрахункових сезонних компонентів і компонентів циклу тренду:

Датасет був розподілений на 3 частини: для тестування, валідації та тренування.

У першому випадку увійшло 70% даних, у другому та в третьому по 30% відповідно.

Спрогнозуємо дані.

Був використаний «наївний» метод прогнозування.

При створенні "наївних" моделей передбачається, що деякий основний період прогнозованого тимчасового ряду краще всього описує майбутнє цього прогнозованого ряду, тому в цих моделях прогноз, як правило, є дуже простою функцією від значень прогнозованої змінної в недалекому минулому.

Найпростішою моделлю є

що відповідає припущенню, що "завтра буде як сьогодні".

Поза всяким сумнівом, від такої примітивної моделі не варто чекати великої точності. Вона не тільки не враховує механізми, що визначають прогнозовані дані (цей серйозний недолік взагалі притаменний багатьом статистичним методам прогнозування), але і не захищена від випадкових коливань, вона не враховує сезонні коливання і тенденції. Втім, можна будувати "наївні" моделі дещо по-іншому

такими способами ми намагаємося пристосувати модель до можливих тенденцій

це спроба врахувати сезонні коливання.

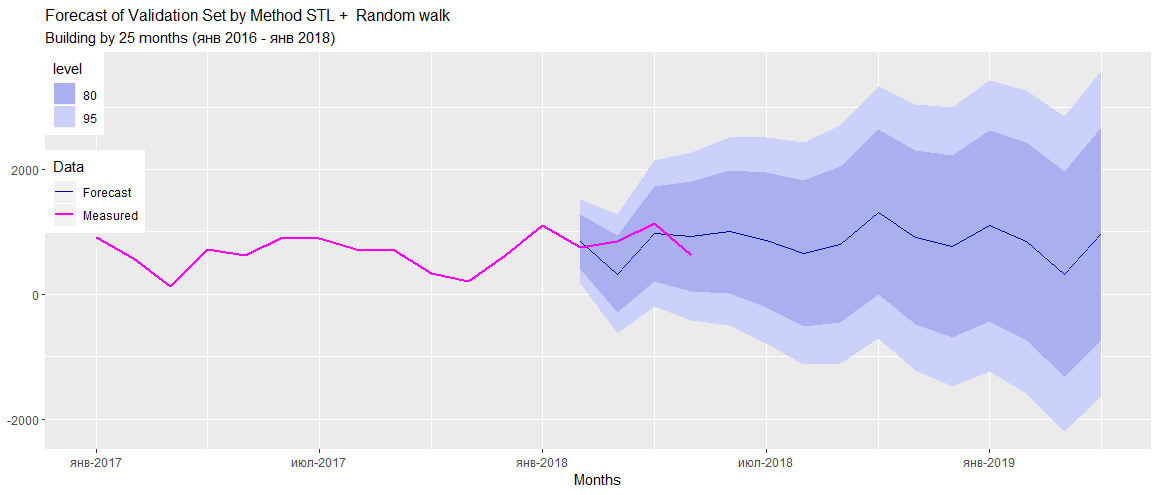


Рисунок 4.2.2 – Прогноз резервування в ресторані

Точність:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ME** | **RMSE** | **MAE** | **MPE** | **MAPE** | **MASE** | **ACF1** | **Theil’s U** |
| **Training set** | 23.2 | 344 | 218.1 | 2.3 | 45.4 | 0.6 | -0.1 | NA |
| **Validation set** | 71.2 | 3.2 | 269.2 | 3.7 | 34.2 | 0.8 | -0.1 | 1.3 |

Таблиця 4.2.1 – Прогноз резервування в ресторані

**MAE** – Mean Absolute Error

Це надає меншу вагу викидам, які не чутливі до викидів.

**MAPE** – Mean Absolute Percentage Error

Подібний до MAE, але нормований за допомогою істинних спостережень.

**MSE** – Mean Squared Error

MSE - це як комбіноване вимірювання упередженості та дисперсії прогнозу.

**RMSE** – Root MSE.

Аналізуючи дану таблицю, можна сказати, що прогноз відбувся у рамках норми. Тобто відхилення не критичні (MAPE). У деяких випадках він міг би досягатися до 70% погрішності, якщо взяти менший розмір датасету для валідації.

* 1. Модель прогнозу для датасету відвідування.

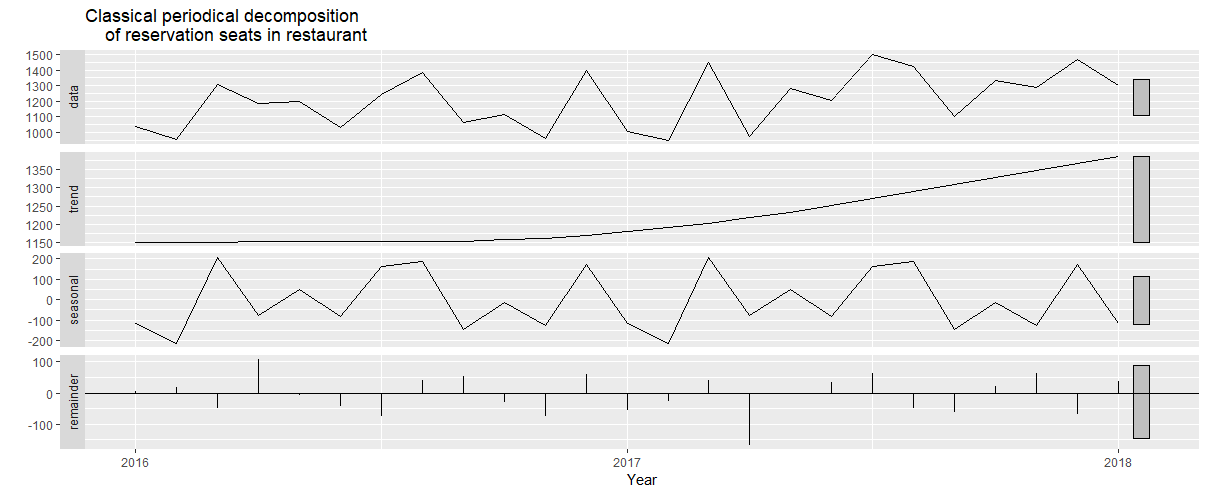


Рисунок 4.3.1 – Графік декомпозиції відвідування у ресторані

Спрогнозуємо дані.

Був використаний «наївний» метод прогнозування.

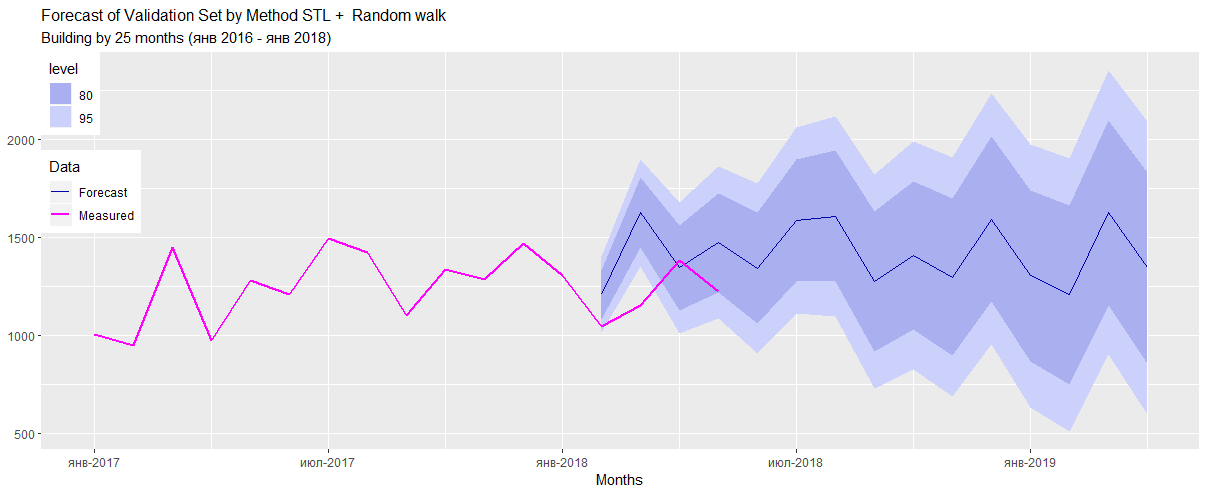


Рисунок 4.3.2 – Прогноз відвідування в ресторані

Точність:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **ME** | **RMSE** | **MAE** | **MPE** | **MAPE** | **MASE** | **ACF1** | **Theil’s U** |
| **Training set** | 11.1 | 91 | 84 | 0.4 | 6.9 | 0.57 | -0.4 | NA |
| **Validation set** | -212 | 281 | 231 | -18 | 20 | 1.57 | -0.65 | 1.9 |

Таблиця 4.2.1 – Прогноз резервування в ресторані

Аналізуючи дану таблицю, можна сказати, що прогноз відбувся успішно. Основною оцінкою «успішності» є MAPE. Навіть для Validation set погрішність всього-на-всього 20%. Це дуже добре.

Тобто відхилення не критичні (MAPE). У деяких випадках він міг би досягатися до 70% погрішності, якщо взяти менший розмір датасету для валідації. Дивлячись на MAE, припущення, що даний датасет є чутливий до викидів. Тобто сильно реагує на аномальні перепади в самому датасеті. Адже, дійсно, у вибраному закладі були «прогалини», тобто мало відвідувачів.

Отже, у рамках даної курсової роботи було спрогнозовано кількість відвідувачів у ресторані у двох категоріях: резервація місць та відвідання самого ресторану. Було побудовано безліч різних інтерактивних графіків аналізу датасету. Було використано «наївний» метод прогнозування. При створенні "наївних" моделей передбачається, що деякий основний період прогнозованого тимчасового ряду краще всього описує майбутнє цього прогнозованого ряду, тому в цих моделях прогноз, як правило, є дуже простою функцією від значень прогнозованої змінної в недалекому минулому.

Через те, що був знайдений не дуже великий датасет, можливі погрішності. У першому випадку, коли відбувався аналіз резерву, погрішність досягла до 30-35 відсотків. Такий відсоток спостерігається через те, що резерв у ресторані відбувається наперед та на конкретну годину. Тобто є непердбачуваність та невідома кількість заповненості ресторану в реальний момент часу. Також датасет не вказує максимальну кількість посадкових місць, що ускладнює прогнозування.

Підбиваючи підсумки, можна стверджувати, що аналіз відбувся успішно, незважаючи на незначні погрішності при прогнозуванні.

Додаток А Технічне завдання

КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ ім. І. Сікорського

Кафедра

автоматизованих систем обробки інформації та управління

Затвердив

Керівник \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\_ р.

Виконавець:

Студент Бурятов Олексій Олексійович

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_202\_ р.

ТЕХНІЧНЕ ЗАВДАННЯ

на виконання курсової роботи

на тему: ІС «Прогнозування потоку відвідувачів у ресторані. Кількість заброньованих місць»

з дисципліни:

«Аналіз даних в інформаційно-управляючих системах»

Київ 2021

Додаток Б Тексти програмного коду

**SQL Server**

CREATE DATABASE RestaurantVisitingDB

USE RestaurantVisitingDB

--DROP TABLE IF EXISTS Dim\_Cuisine

CREATE TABLE Dim\_Cuisine(

cuisineID INT PRIMARY KEY NOT NULL,

cuisineName NVARCHAR(255 ) NOT NULL,

loadDate DATETIME DEFAULT(GETDATE()) NOT NULL,

loadEndDate DATETIME DEFAULT('9999-01-01 23:59:59') NOT NULL

)

--DROP TABLE IF EXISTS Dim\_Place

CREATE TABLE Dim\_Place(

placeID INT PRIMARY KEY NOT NULL,

restaurantName NVARCHAR(255) NOT NULL,

cuisineID INT NOT NULL,

latitude NVARCHAR(255) NOT NULL,

longitude NVARCHAR(255) NOT NULL,

CONSTRAINT FK\_Dim\_Cuisine\_Dim\_Place FOREIGN KEY (cuisineID)

REFERENCES Dim\_Cuisine (cuisineID)

ON DELETE CASCADE

ON UPDATE CASCADE

)

--DROP TABLE IF EXISTS Dim\_Reservation

CREATE TABLE Dim\_Reservation(

reserveID INT PRIMARY KEY NOT NULL,

dateID INT NOT NULL,

timeValue TIME NOT NULL,

resrvAmntOfVisitors INT NOT NULL,

placeID INT NOT NULL,

CONSTRAINT FK\_Dim\_Date\_Dim\_Reservation FOREIGN KEY (dateID)

REFERENCES Dim\_Date (dateID)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_Dim\_Place\_Dim\_Reservation FOREIGN KEY (placeID)

REFERENCES Dim\_Place (placeID)

ON DELETE CASCADE

ON UPDATE CASCADE

)

CREATE TABLE Dim\_DayOfTheWeek(

dayOfTheWeekID INT PRIMARY KEY NOT NULL,

dayOfTheWeekName NVARCHAR(255) NOT NULL

)

--DROP TABLE IF EXISTS Dim\_Date

CREATE TABLE Dim\_Date(

dateID INT PRIMARY KEY NOT NULL,

dayValue DATE NOT NULL,

dayOfTheWeekID INT NOT NULL,

isHoliday BIT NOT NULL,

CONSTRAINT FK\_Dim\_DayOfTheWeek\_Dim\_Date FOREIGN KEY (dayOfTheWeekID)

REFERENCES Dim\_DayOfTheWeek (dayOfTheWeekID)

ON DELETE CASCADE

ON UPDATE CASCADE

)

DROP TABLE Fact\_RestaurantVisiting

CREATE TABLE Fact\_RestaurantVisiting(

dateID INT NOT NULL,

placeID INT NOT NULL,

amountOfVisitors INT NOT NULL,

CONSTRAINT FK\_Dim\_Date\_Fact\_RestaurantVisiting FOREIGN KEY (dateID)

REFERENCES Dim\_Date (dateID)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT FK\_Dim\_Place\_Fact\_RestaurantVisiting FOREIGN KEY (placeID)

REFERENCES Dim\_Place (placeID)

ON DELETE CASCADE

ON UPDATE CASCADE,

CONSTRAINT PK\_dateID\_placeID PRIMARY KEY (dateID, placeID)

CREATE OR ALTER FUNCTION dbo.getFactAmountVisitors(

@placeName NVARCHAR(255)

)

RETURNS TABLE

AS RETURN (

SELECT dd.dayValue

,frv.amountOfVisitors

,wd.dayOfTheWeekName

,dd.isHoliday

FROM dbo.Fact\_RestaurantVisiting frv

JOIN dbo.Dim\_Date dd ON dd.dateID=frv.dateID

JOIN dbo.Dim\_DayOfTheWeek wd ON wd.dayOfTheWeekID=dd.dayOfTheWeekID

WHERE frv.placeID=(

SELECT dp.placeID FROM dbo.Dim\_Place dp

WHERE dp.restaurantName=@placeName

)

);

GO

CREATE OR ALTER FUNCTION dbo.getFactAmountResrvVisitors(

@placeName NVARCHAR(255)

)

RETURNS TABLE

AS RETURN(

SELECT DISTINCT dd.dayValue

,SUM(dr.resrvAmntOfVisitors) OVER(PARTITION BY dd.dayValue) AS reservSum

,wd.dayOfTheWeekName

,dd.isHoliday

FROM dbo.Dim\_Reservation dr

JOIN dbo.Dim\_Date dd ON dd.dateID=dr.dateID

JOIN dbo.Dim\_DayOfTheWeek wd ON wd.dayOfTheWeekID=dd.dayOfTheWeekID

WHERE dr.placeID =(

SELECT dp.placeID FROM dbo.Dim\_Place dp

WHERE dp.restaurantName=@placeName

)

);

GO

**RStudio**

remove(list=ls())

library(RODBC)

library(dplyr)

library(forecast)

library(lubridate)

library("xts")

library(ggplot2)

library(hrbrthemes)

library(tseries)

library(fpp2)

library(forecast)

setwd("d:/Studying/4 семестр/Аналіз даних/Курсова робота")

connStr<- "Driver={SQL Server};Server=LT-ALEXEI\\MSSQLSERVER01;Database=RestaurantVisitingDB"

db <- odbcDriverConnect(connStr, case="nochange")

import.to.Dim\_DayOfTheWeek<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("dayOfTheWeekID"= rawData$dateID,

"dayOfTheWeekName"= rawData$value,stringsAsFactors = F )

sqlSave(connection, rawData, tablename = "dbo.Dim\_DayOfTheWeek", rownames = F, append=T)

}

import.to.Dim\_DayOfTheWeek("day\_ofTheWeek\_info.csv", db)

import.to.Dim\_Date<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("dateID"=rawData$dateID,

"dayValue"= rawData$calendar\_date,

"dayOfTheWeekID"=rawData$day\_of\_the\_week\_id,

"isHoliday"=rawData$holiday\_flg,

stringsAsFactors = F )

sqlSave(connection, rawData, tablename = "dbo.Dim\_Date", rownames = F, append=T)

}

update.to.Dim\_Date<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("dateID"=rawData$dateID,

"dayValue"= rawData$calendar\_date,

"dayOfTheWeekID"=rawData$day\_of\_week\_id,

"isHoliday"=rawData$holiday\_flg,

stringsAsFactors = F )

sqlUpdate(connection, rawData, tablename = "dbo.Dim\_Date")

}

import.to.Dim\_Date("date\_info\_1.csv", db)

update.to.Dim\_Date("UPD\_date\_info.csv", db)

import.to.Dim\_Reservation<- function(path="air\_reserve\_1.csv", connection=db){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("reserveID"=rawData$reserveID,

"dateID"= rawData$dateID,

"timeValue"=rawData$time\_value,

"resrvAmntOfVisitors"=rawData$reserve\_visitors,

"placeID"=rawData$placeID,

stringsAsFactors = F )

sqlSave(connection, rawData, tablename = "dbo.Dim\_Reservation", rownames = F, append=T)

}

update.to.Dim\_Reservation<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("reserveID"=rawData$reserveID,

"dateID"= rawData$dateID,

"timeValue"=rawData$time\_value,

"resrvAmntOfVisitors"=rawData$reserve\_visitors,

"placeID"=rawData$placeID,

stringsAsFactors = F )

sqlUpdate(connection, rawData, tablename = "dbo.Dim\_Reservation")

}

import.to.Dim\_Reservation("air\_reserve\_1.csv", db)

update.to.Dim\_Reservation("UPD\_air\_reserve.csv", db)

import.to.Dim\_Cuisine<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("cuisineID"=rawData$cuisine\_id,

"cuisineName"= rawData$air\_genre\_name,

"loadDate"=rep(Sys.time(), times= nrow(rawData)),

"loadEndDate"=rep(as.Date('9999-01-01'), times= nrow(rawData)),

stringsAsFactors = F )

sqlSave(connection, rawData, tablename = "dbo.Dim\_Cuisine", rownames = F, append=T)

}

update.to.dim\_Cuisine <-function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("cuisineID"= rawData$cuisine\_id,

"loadEndDate"=rep((Sys.time()), times= nrow(rawData)),

stringsAsFactors = F )

sqlUpdate(connection, rawData, tablename = "dbo.Dim\_Cuisine")

x<-paste("SELECT TOP 1 cuisineID FROM dbo.Dim\_Cuisine ORDER BY cuisineID DESC",

sep = "", collapse=NULL)

x1<-sqlQuery(connection, x)

x1<- as.integer(x1$cuisineID)

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("cuisineID"= rawData$cuisine\_id,

"cuisineName"= rawData$air\_genre\_name,

"loadDate"= rep(Sys.time(), times= nrow(rawData)),

"loadEndDate"=rep(as.Date('9999-01-01'), times= nrow(rawData)),

stringsAsFactors = F )

for (n in 1:nrow(rawData)) {

x1<-x1+1

rawData$cuisineID[n]<-x1

}

sqlSave(connection, rawData, tablename = "dbo.Dim\_Cuisine", rownames = F, append=T)

}

import.to.Dim\_Cuisine("air\_cuisine\_info.csv", db)

update.to.dim\_Cuisine("UPD\_air\_cuisine\_info.csv", db)

import.to.Dim\_Place<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("placeID"=rawData$place\_id,

"restaurantName"= rawData$air\_area\_name,

"cuisineID"=rawData$cuisine\_id,

"latitude"=rawData$latitude,

"longitude"=rawData$longitude,

stringsAsFactors = F )

sqlSave(connection, rawData, tablename = "dbo.Dim\_Place", rownames = F, append=T)

}

update.to.Dim\_Place<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("placeID"=rawData$place\_id,

"restaurantName"= rawData$air\_area\_name,

"cuisineID"=rawData$cuisine\_id,

"latitude"=rawData$latitude,

"longitude"=rawData$longitude,

stringsAsFactors = F )

sqlUpdate(connection, rawData, tablename = "dbo.Dim\_Place")

}

import.to.Dim\_Place("air\_place\_info.csv", db)

update.to.Dim\_Place("UPD\_air\_place\_info.csv", db)

import.to.Fact\_RestaurantVisiting<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("dateID"=rawData$dateID,

"placeID"= rawData$placeID,

"amountOfVisitors"=rawData$amountOfVisitors,

stringsAsFactors = F )

sqlSave(connection, rawData, tablename = "dbo.Fact\_RestaurantVisiting", rownames = F, append=T)

}

update.to.Fact\_RestaurantVisiting<- function(path, connection){

rawData <- read.csv(file= path, stringsAsFactors = F)

rawData<- data.frame("dateID"=rawData$dateID,

"placeID"= rawData$placeID,

"amountOfVisitors"=rawData$amountOfVisitors,

stringsAsFactors = F )

sqlUpdate(connection, rawData, tablename = "dbo.Fact\_RestaurantVisiting")

}

import.to.Fact\_RestaurantVisiting("air\_visit\_data\_1.csv", db)

update.to.Fact\_RestaurantVisiting("UPD\_air\_visit\_data.csv", db)

#import from SQL SERVER

dataReserve<-sqlQuery(channel=db, query = "SELECT \* FROM dbo.getFactAmountResrvVisitors('Hyogo-ken Kakogawa Kakogawacho Kitazaike')")

dataWholeAmount<-sqlQuery(channel=db, query = "SELECT \* FROM dbo.getFactAmountVisitors('Hyogo-ken Kakogawa Kakogawacho Kitazaike')")

#Show By Week

getPlotByAllWeeks<-function(data,dayValue,sumValue){

x<-data.frame(Date=as.Date(dayValue), Reserved=sumValue)

x<- x %>% group\_by(month=floor\_date(Date, "week")) %>%

summarize(Reserved=sum(Reserved))

dfReserve<-data.frame(month=as.Date(x$month,

origin = "1899-12-30"), Reserved=x$Reserved, stringsAsFactors = T)

dfReserve$month<-as.factor( format(dfReserve$month,"%d/%m/%Y" ))

dfReserve.xts<-as.xts(dfReserve[,-1], order.by = as.yearmon(as.character(dfReserve[,1]),

"%d/%m/%Y"))

colnames(dfReserve.xts)<-"Table Reserve Visitors, people"

attr(dfReserve.xts, 'frequency') <- 7

df.ts <- as.ts(dfReserve.xts, start=1900 + .indexyear(dfReserve.xts)[1] +

.indexmon(dfReserve.xts)[1] / frequency(dfReserve.xts), names=colnames(dfReserve.xts))

forecast::ggmonthplot(df.ts, xlab="Weeks", ylab=colnames(as.vector(dfReserve.xts)), main="7 Days of Each Year")

}

getPlotByAllWeeks(dataWholeAmount,dataWholeAmount$dayValue,dataWholeAmount$amountOfVisitors )

getPlotByAllWeeks(dataReserve,dataReserve$dayValue,dataReserve$reservSum)

#Show By All Years

getPlotByAllYears<-function(data, dayValue, sumValue){

x<-data.frame(Date=as.Date(dayValue), Reserved=sumValue)

x<- x %>% group\_by(month=floor\_date(Date, "month")) %>%

summarize(Reserved=sum(Reserved))

dfReserve<-data.frame(month=as.Date(x$month,

origin = "1899-12-30"), Reserved=x$Reserved, stringsAsFactors = T)

dfReserve$month<-as.factor(format(dfReserve$month,"%d/%m/%Y" ))

dfReserve.xts<<-as.xts(dfReserve[,-1], order.by = as.yearmon(as.character(dfReserve[,1]),

"%d/%m/%Y"))

df.xts<<-dfReserve.xts

colnames(dfReserve.xts)<-"Table Reserve Visitors, people"

df.ts <<- as.ts(dfReserve.xts, start=1900 + .indexyear(dfReserve.xts)[1] +

.indexmon(dfReserve.xts)[1] / frequency(dfReserve.xts), names=colnames(dfReserve.xts))

forecast::ggseasonplot(df.ts, year.labels=TRUE, main="Months of All Years")

}

getPlotByAllYears(dataWholeAmount,dataWholeAmount$dayValue,dataWholeAmount$amountOfVisitors)

getPlotByAllYears(dataReserve,dataReserve$dayValue,dataReserve$reservSum)

#Show Scatter

getScatterplot<- function(data, dayValue, sumValue){

x<-data.frame(Date=as.Date(dayValue), Reserved=sumValue)

x<- x %>% group\_by(month=floor\_date(Date, "20 days")) %>%

summarize(Reserved=sum(Reserved))

x %>%

ggplot( aes(x=month, y=Reserved)) +

geom\_line( color="grey") +

geom\_point(shape=21, color="black", fill="#69b3a2", size=3) +

theme\_ipsum() +

ggtitle("Evolution of reserving seats")

}

getScatterplot(dataWholeAmount,dataWholeAmount$dayValue,dataWholeAmount$amountOfVisitors)

getScatterplot(dataReserve,dataReserve$dayValue,dataReserve$reservSum)

train.ts <- window(df.ts, start=2016, end=2018)

validation.ts <- window(df.ts, start=2018, end=2018 +2/frequency(df.ts))

test.ts <- window(df.ts, start=2018 +3/frequency(df.ts), end=2018 +4/frequency(df.ts))

hf <- 3 # length(validation.ts)

Subtitle <- paste0("Building by ", round((tsp(train.ts)[2] -

tsp(train.ts)[1]) \* tsp(train.ts)[3] + 1, digits = 0), " months (",

zoo::yearmon(tsp(train.ts)[1]), " - ", zoo::yearmon(tsp(train.ts)[2]), ")")

fit.stl = stl(train.ts, s.window="periodic", robust=TRUE)

autoplot(fit.stl)

Autoplot.Forecast(fcast.stl, validation.ts)

Autoplot.Forecast(fcast.arima, validation.ts)

(fcast.stl <- forecast(fit.stl, h=hf, method="naive"))

res.stl <- Proof.Residuals(fcast.stl)

df.ts = stl(train.ts, robust=TRUE)

install.packages("fpp2")

(fcast.stl <- forecast(fit.stl, h=hf, method="naive"))

(fcast.stl <- forecast(fit.stl, h=hf, method="naive"))

lam <- 0

( fit.ets <- ets(train.ts, lambda=lam, model="AAA") )

( fcast.ets <- forecast(fit.ets, h=hf, lambda=lam, level=c(80, 95)) )

qq.line <- function(data, qf, na.rm) {

# from stackoverflow.com/a/4357932/1346276/

q.sample <- quantile(data, c(0.25, 0.75), na.rm = na.rm)

q.theory <- qf(c(0.25, 0.75))

slope <- diff(q.sample) / diff(q.theory)

intercept <- q.sample[1] - slope \* q.theory[1]

list(slope = slope, intercept = intercept)

}

# Analog of car::qqPlot for ggplot2

QQ.GGplot <- function(x, distribution = "norm", ..., line.estimate = NULL,

conf = 0.95, labels = names(x)){

# from stackoverflow.com/questions/4357031/qqnorm-and-qqline-in-ggplot2/

q.function <- eval(parse(text = paste0("q", distribution)))

d.function <- eval(parse(text = paste0("d", distribution)))

x <- na.omit(x)

ord <- order(x)

n <- length(x)

P <- ppoints(length(x))

df <- data.frame(ord.x = x[ord], z = q.function(P, ...))

if(is.null(line.estimate)){

Q.x <- quantile(df$ord.x, c(0.25, 0.75))

Q.z <- q.function(c(0.25, 0.75), ...)

b <- diff(Q.x)/diff(Q.z)

coef <- c(Q.x[1] - b \* Q.z[1], b)

} else {

coef <- coef(line.estimate(ord.x ~ z))

}

zz <- qnorm(1 - (1 - conf)/2)

SE <- (coef[2]/d.function(df$z)) \* sqrt(P \* (1 - P)/n)

fit.value <- coef[1] + coef[2] \* df$z

df$upper <- fit.value + zz \* SE

df$lower <- fit.value - zz \* SE

if(!is.null(labels)){

df$label <- ifelse(df$ord.x > df$upper | df$ord.x < df$lower,

labels[ord], "")

}

p <- ggplot(df, aes(x=z, y=ord.x)) +

geom\_point() +

geom\_abline(intercept = coef[1], slope = coef[2], col = "red") +

geom\_ribbon(aes(ymin = lower, ymax=upper), fill="coral", alpha=0.15) +

labs(title = "Quantiles of In-Sample & Normal Distribution",

x = "Theoretical Quantiles", y = "In-Sample Quantiles")

if(!is.null(labels)) p <- p + geom\_text(aes(label = label),

size = 3, vjust = 0, nudge\_y = max(x)\*0.03)

print(p)

# coef

}

#Function to Examination Residuals of time series forecasting.

Proof.Residuals <- function(x)

{

# Load required parameters and packages

stopifnot(is.forecast(x) | is.null(x))

if (!requireNamespace("forecast", quietly = TRUE))

stop("forecast is needed for this function to work. Install it via install.packages(\"forecast\")", call. = FALSE)

if (!requireNamespace("ggplot2", quietly = TRUE))

stop("ggplot2 is needed for this function to work. Install it via install.packages(\"ggplot2\")", call. = FALSE)

if (length(x$method)>1)

Title = "Ensemble of Forecast Models"

else Title = x$method

# Plot some Charts of Residuals

ggtsdisplay(residuals(x)[-1],main=paste("Residuals from",

Title), ylab="", xlab="Months", plot.type="scatter")

z <- as.numeric(residuals(x))

names(z) <- as.yearmon(time(residuals(x)))

QQ.GGplot(z[-1])

# Test of Normality and AutoCorrelation of Residuals

TestOfResiduals <- c(NormalityOfResiduals="",

AutoCorrelationOfResiduals="")

print( shapiro.test(residuals(x)) )

cat( "Normality of Residuals is",

shapiro.test(residuals(x))$p.value > 0.05, "\n" )

TestOfResiduals$NormalityOfResiduals <-

ifelse(shapiro.test(residuals(x))$p.value > 0.05, "Normality of Residuals is TRUE", "Normality of Residuals is FALSE")

#"остатки распределены нормально","остатки не распределены нормально")

print( Acf(residuals(x), lag.max=frequency(x$residuals)\*2, plot=FALSE) )

print( Pacf(residuals(x), lag.max=frequency(x$residuals)\*2, plot=FALSE))

# Has Autocorrelation of Residuals any Lags?

print( Box.test(residuals(x), lag=24, type="Ljung") )

cat( "AutoCorrelation is", Box.test(residuals(x),

lag=frequency(x$residuals)\*2, type="Ljung")$p.value < 0.05, "\n" )

TestOfResiduals$AutoCorrelationOfResiduals <-

ifelse(Box.test(residuals(x), lag=frequency(x$residuals)\*2,

type="Ljung")$p.value < 0.05, "AutoCorrelation of Residuals is TRUE",

"AutoCorrelation of Residuals is FALSE")

return(TestOfResiduals)

}

#Function to Plotting of Forecast Models with Confidence intervals.

Autoplot.Forecast <- function(x, y = NULL, testing=FALSE)

{

# Load required parameters and packages

stopifnot(is.forecast(x) | is.null(x) )

if (!requireNamespace("forecast", quietly = TRUE))

stop("forecast is needed for this function to work. Install it via install.packages(\"forecast\")", call. = FALSE)

if (!requireNamespace("ggplot2", quietly = TRUE))

stop("ggplot2 is needed for this function to work. Install it via install.packages(\"ggplot2\")", call. = FALSE)

if (length(x$method)>1)

Title = "Ensemble of Prediction Models"

else Title = x$method

listing <- accuracy(x, y) # In-sample & Out-sample Accuracy

rownames(listing)[2] <- ifelse(testing, "Testing Set", "Validation set")

cat("\nAccuracy of", Title, "\n")

print(listing)

p <- autoplot(x, main=Title, xlab="Year", ylab=colnames(df.xts)) +

geom\_line(data = fortify(x$fitted, melt = FALSE), aes(x = x, y = y),

na.rm = TRUE, col = "#0000AA", linetype = "dashed")

print(p)

if ( !is.null(y) ) {

# Initialise ggplot object

fcdata <- fortify(merge.xts(Forecast = as.xts(x$mean), Actual = y), melt = TRUE)

Labels <- c("Forecast", "Measured")

gptitle <- ifelse(testing, paste("Forecast of Testing Set by Final Method", Title),

paste("Forecast of Validation Set by Method", Title))

p <- autoplot(x, include = 0)

# p <- ggplot()

# # Only for Interval

# if ( !is.null(x$upper) ) {

# levels <- NROW(x$level)

# interval <- data.frame(Index = as.yearmon(rep(time(x$mean),levels)),

# lower = c(x$lower), upper = c(x$upper),

# level = rep(x$level, each = NROW(x$mean)))

# interval <- interval[order(interval$level, decreasing = TRUE),] # Must be ordered for gg z-index

#

# p <- p + geom\_ribbon(ggplot2::aes\_(x = ~Index, ymin = ~lower,

# ymax = ~upper, group = ~-level, fill = ~level), data = interval)

# # Negative group is a work around for missing z-index

# if(min(x$level) < 50){

# scalelimit <- c(1,99)

# } else {

# scalelimit <- c(50,99)

# }

#

# if(length(x$level) <= 5){

# p <- p + scale\_fill\_gradientn(breaks = x$level,

# colours = c("#596DD5", "#D5DBFF"),

# limit = scalelimit, guide = "legend")

# } else {

# p <- p + scale\_fill\_gradientn(colours =c("#596DD5", "#D5DBFF"),

# limit = scalelimit)

# }

# }

# Forecasted and Validationed points

p <- p + geom\_line(data = fcdata, aes\_(x = ~Index, y = ~Value, color = ~Series, size = ~Series))

p <- p + scale\_colour\_manual(name = "Data", labels = Labels,

values=c("Forecast" = "#0000AA", "Actual" = "magenta")) +

scale\_size\_manual(name = "Data", labels = Labels,

values = c("Forecast" = 0.5, "Actual" = 1)) +

scale\_x\_yearmon(format = "%b-%Y", n = 5) +

theme(legend.position = c(0, 1), legend.justification = c(0, 1)) +

labs(title = gptitle, subtitle = Subtitle,

x = "Months", y = colnames(df.xts)) +

theme(plot.title = element\_text(size = 12))

print(p)

}

*студента групи ІП-96 IІ курсу*

*Бурятова О.О.*

(Обсяг програми (документа), арк., Кб)

*12 арк, 339 Кб*

(Вид носія даних)

*CD-RW*

}

(Найменування програми (документа))

*Тексти програмного коду програмного забезпечення розробки «Прогнозування потоку відвідувачів у ресторані. Кількість заброньованих місць»*

Перелік посилань

1. SQL за 10 минут [Електронний ресурс] // Вильямс. – 2014. – Режим доступу до ресурсу: http://padabum.com/d.php?id=161361
2. Alexandr Rodionov. Machine learning: Forecast of Table Water Sales [Електронний ресурс] / Alexandr Rodionov – Режим доступу до ресурсу: https://rstudio-pubs-static.s3.amazonaws.com/374035\_9b3e644472834ebbacd3db26cd1e3f70.html.