

LATVIJAS UNIVERSITĀTE
DATORIKAS FAKULTĀTE

**AVIĀCIJAS DROŠĪBAS PĀRBAUDES KONTROLPUNKTA DARBĪBAS
PLĀNOŠANAS OPTIMIZĀCIJA**

MAGISTRA DARBS

Autors: **Edgars Zēbergs**

Stud. apl. Nr. ez18029

Darba vadītājs: profesors Dr. math. Kārlis Podnieks

RĪGA 2021

ANOTĀCIJA

Darba mērķis ir piedāvāt labāku prognozēšanas metodi aviācijas drošības pārbaudes kontrolpunktam, kas ļautu veidot un plānot precīzākus kontrolpunkta pārbaudes līniju darba grafikus. Precīzāki darba grafiki ļautu samazināt rindā gaidīšanas laiku, un ļautu pasažieriem ātrāk iziet drošības pārbaudi, bet no uzņēmējdarbības puses netiktu atvērtas pārāk daudz drošības pārbaudes līnijas, tādējādi izmantojot personāla resursu jaudu pilnībā.

Tika izmantotas trīs datu kopas – izpildīto reisu datu tabula, pasažieru skenētās iekāpšanas kartes, dati par gaisa kuģu tipiem un to kapacitāti. Darbā ir aprakstīta šo datu kopu detalizēta priekšapstrāde. Plaši pielietojot dažādus vaicājumus tika iegūti jauni atribūti, bez kuriem nebūtu iespējama prognozēšanas modeļa izveide.

Pēc datu priekšapstrādes tiek izmantotas datu dimensiju redukcijas un analīzes metodes - *T-SNE* un *PCA* (galveno komponentu analīze), un iegūtie rezultāti un secinājumi tiek pārbaudīti ar datu vizuālizācijas rīkiem un aprēķiniem. Iegūtie rezultāti tiek ņemti vērā veicot tālāku modeļa izstrādi.

Tika izpētītas reisu pasažieru skaita laika rindas, veidojot modeli, tika izmēģinātas gan vienkāršas prognozēšanas metodes, kuras nebija efektīvas, gan sarežģīti laika rindu prognozēšanas algoritmi, bet par labāko tika atzīta *Holt – Winters* sezonālā izlīdzināšanas metode ar kuras palīdzību tiek prognozēts pasažieru skaits katram reisam.

Tika izveidots katra reisa pasažieru ierašanās profils drošības pārbaudes kontrolpunktā.

Ar divu komponentu palīdzību – pasažieru skaita prognozi un pasažieru ierašanās profilu izveidi tika izveidots kopējais drošības pārbaudes kontrolpunkta pasažieru skaita ierašanās grafiks, ar kura palīdzību ir iespējams plānot precīzākus drošības kontrolpunkta līniju darba grafikus salīdzinot ar pašreiz izmantotajām prognozēm, kuras drošības departamenta darbinieki veic, izmantojot lidojumu sarakstu, gaisa kuģa kapacitāti un tipu, un pieņemot, ka gaisa kuģis būs ar 80% aizpildītām sēdvietām.

Darba apjoms – 60 lpp., 4 tabulas, 39 attēli, 1 pielikums.

Atslēgvārdi: laika rindu prognozēšana, *Holt-Winters's* sezonālā izlīdzināšanas metode, aviācijas pasažieri, *Python* programmēšanas valoda

ABSTRACT

AVIATION SECURITY CHECKPOINT OPERATION PLANNING OPTIMIZATION

The objective of the thesis is to offer a better forecasting method for the aviation security checkpoint, which would allow to produce and plan more accurate work schedules of checkpoint inspection lines. More precise work schedules would reduce time spent waiting in the queues and allow passengers to pass security checks more quickly, and from the business perspective this would not open up too many security check lines, thus making full use of staff resources.

Three data sets were used: a table of completed flights data, scanned boarding passes of the passengers, data on aircraft types and their capacity. The thesis describes the detailed pre-processing of these data sets. Extensive use of various queries resulted in new attributes, without which it would not be possible to create a forecasting model.

After data pre-processing, dimensionality reduction and analysis methods are used - T-SNE and PCA (Principal component analysis), and the gathered results and conclusions are verified with data visualization tools and calculations. The obtained results are taken into account in further development of the model.

A study of time series of the number of passengers in the flights took place by applying simple modelling methods that were not efficient, as well as complex time series forecasting algorithms, however the Holt - Winters seasonal smoothing method, which predicts the number of passengers for each flight, was considered the best.

A passenger arrival profile for each flight was created at the security checkpoint.

With the use of two components - number of passengers forecast and passenger arrival profiling, a common security checkpoint schedule of passenger arrival was created, which allows to plan more accurate security checkpoint line work schedules compared to the currently used forecasts made by security department staff who determines it by using data such as flight schedule, capacity and type of aircraft, and holds the assumption that the aircraft will have 80% seating.

Thesis consists of: 60 pages, 4 tables, 39 images, 1 appendice.

Keywords: time series forecasting, Holt-Winters`s seasonal adjustment method, air passengers, Python programming language

AUTOREFERĀTS

Darba autors sniedz ieskatu aviācijas drošības pārbaudes kontrolpunkta darbībā, tā procesos. Balstoties uz literatūras avotu izpēti, tiek pētītas vienkāršas prognozēšanas metodes, gan sarežģīti laika rindu prognozēšanas algoritmi. No darbā izmantotajiem literatūras avotiem - 6 ir zinātniski raksti un 6 - grāmatas.

Autors, darba gaitā, apraksta un analizē izmantotās datu kopas, detalizēti izklāstot katru datu kopas atribūtu, pamatojot jaunizveidoto datu atribūtu nepieciešamību. Darba autors pielieto datu dimensiju redukcijas un analīzes metodes, apraksta un pamato iegūtos rezultātus. Darbā autors izmanto datu vizualizāciju, kurā veic salīdzinošu analīzi.

Python vidē ir izstrādāts laika rindu prognozēšanas, apstrādes un analīzes modelis. Autors ir aprakstījis izmantoto kodu.

Darba autors sadarbībā ar kolēģiem par darba pamatu ir izvēlējis Kopenhāgenas lidostas modeli, kurā tiek izmantotas katra reisa pasažieru skaita prognozes un pasažieru ierašanās drošības kontrolpunktā pirms reisa prognozes. Izmantojot šīs abas komponentes tiek izveidots kopējais drošības pārbaudes kontrolpunkta pasažieru skaita ierašanās grafiks.

Tika ieguldīts daudz laika datu priekšapstrādē un analīzē.

Darba teksts ir pārlasīts un atrastās ievades, komatu un stila kļūdas ir izlabotas. Autors ir pārliecinājies, ka darbā nav pareizrakstības kļūdu un darbā ir izmantota latviešu valodā oficiāli pieņemtā nozares terminoloģija. Pirms maģistra darba iesniegšanas ir apskatīts "Darba noformējuma kontrolsaraksts"

Visas idejas un formulējumi, kas aizgūti no citiem autoriem, ir atzīmēti ar attiecīgām literatūras atsaucēm.

SATURS

IEVADS	8
1. KOPENHĀGENAS LIDOSTĀ IZMANTOTAIS MODELIS	10
1.1. Realizētais lidojumu grafiks	11
1.2. Realizētais lidojumu saraksts (pēdējais periods).....	11
1.3. Pasažieru skaita prognoze	12
1.4. Pasažieru ierašanās profila prognoze	12
1.5. Resursu plānošana	14
1.6. Līniju pieprasījumu optimizācija	15
1.7. Personāla pieprasījuma optimizācija	16
2. PROGNOZĒŠANAS UZDEVUMA PAMATA DARBĪBAS	17
2.1. Problēmas definīcija	17
2.2. Informācijas apkopošana	17
2.3. Izpētes analīze.	17
2.4. Modeļu izvēle un uzstādīšana.	17
2.5. Prognozēšanas modeļa izmantošana un novērtēšana.	18
3. DATU PIRMAPSTRĀDE UN ANALĪZE	19
3.1. Datu transformācija	20
3.1.1. Dimensiju skaita samazināšana.....	20
3.1.2. Pazīmes radīšana	21
3.1.3. Mainīgo transformācija	21
3.2. Pasažieru iekāpšanas karšu skenētie dati.	22
3.3. Izlidojošo reisu saraksts	23
3.4. Gaisu kuģu tipu un to kapacitātes datu tabula.....	24
3.5. Jaunu datu tabulu un atribūtu izveide.....	24
3.5.1. Atribūta “Ierašanās” izveidošana.	25
3.5.2. Reisa pasažieru ierašanās profila izveide.....	26
3.5.3. Reisa kopējais pasažieru skaits.	27
3.5.4. Gaisa kuģa ietilpība.....	27
3.5.5. Gaisa kuģu aizpildījums (noslodze).....	27
3.5.6. Gada nedēļas numurs, gada mēneša numurs, un nedēļas dienas numurs.....	28
3.6. Dimensiju reducēšanas algoritmi	28

3.6.1. Galveno komponentu analīze (PCA)	29
3.6.2. Datu dimensijas redukcijas metode T-SNE	30
3.7. Datu vizualizācija POWER BI programmatūrā	35
3.7.1. Pasažieru skaita prognoze.	41
3.7.2. Pasažieru ierašanās profilu izveidošana.	43
4. PASAŽIERU SKAITA PROGNOZĒŠANAS METODOLOĢIJAS.	45
4.1. Dažas vienkāršākās prognozēšanas metodes.....	45
4.2. Laika rindu modeļa izveide un analīze.....	46
4.3. Holt-Winters` s sezonālā izlīdzināšanas metode	47
4.4. Nepieciešamo bibliotēku pievienošana un datu sagatavošana	48
4.5. Stacionaritāte (<i>Stationarity</i>)	50
4.6. Dickey-Fuller tests	51
4.7. Laika rindu dekompozīcija.....	51
4.8. <i>Holt-Winter`s</i> sezonālā izlīdzināšanas metodes izstrāde <i>Python</i>	52
4.9. Drošības kontrolpunkta grafika veidošana	53
5. REZULTĀTI UN SECINĀJUMI. MODEĻA UZLABOŠANA UN NĀKOTNES DARBS	
.....	56
IZMANTOTĀ LITERATŪRA UN AVOTI.....	59
PIELIKUMS.....	61

APZĪMĒJUMU SARAKSTS

AODB – lidostas operāciju datubāze.

E-vārti – vieta pirms aviācijas drošības kontrolpunkta, kur pasažierim jānoskenē sava iekāpšanas karte.

UTC – saskaņotais pasaules laiks.

GMT +2 – vietējais laiks jeb laiks, kuram pieskaitītas divas stundas no saskaņotā pasaules laika.

SQL – strukturēta vaicājumuvaloda.

Pasažieru skaita “pīķis” – vieta grafikā, kur līkne kura virzās pa horizontālo asi veido strauju lēcieni, un tad strauji samazinās.

Pasažieru skaita ierašanās profils – grafiks, kurš tiek izveidots, izmantojot vēsturisko informāciju par pasažieru skenētajām iekāpšanas kartēm drošības kontrolpunktā, mērot laiku, kur pasažieris noskenē savu iekāpšanas karti līdz plānotajai reisa izlidošanai.

Pasažiera attēls – laiks, kurš tiek patērēts viena pasažiera drošības pārbaudes veikšanai.

IEVADS

Tēmas aktualitāte. Ņemot vērā pašreizējo un ar katru gadu pieaugošo pasažieru plūsmu, un pasažieru termināļa nepietiekami lielās telpas jaunu aviācijas drošības kontrolpunktu izveidei, ir nepieciešams izstrādāt labākus pasažieru plūsmas prognozēšanas risinājumus. Tas ļautu pārveidot nākotnes lidojumu grafiku par diennakts drošības kontrolpunkta darbības grafiku, jeb to cik konkrētā laika posmā būs atvērtu drošības pārbaudes līniju, un attiecīgi cik lieli personāla resursi būs nepieciešami.

Maģistra darba mērķis ir piedāvāt labāku prognozēšanas metodi aviācijas drošības pārbaudes kontrolpunktam, kas ļautu veidot un plānot precīzākus kontrolpunkta pārbaudes līniju darba grafikus.

Lai sasniegtu izvirzīto mērķi ir nepieciešams izpildīt šādus **maģistra darba uzdevumus**:

1. Izanalizēt literatūru par datu priekšapstrādi un dažādām datu apstrādes un analīzes metodēm;
2. Veikt padziļinātu literatūras izpēti par laika rindu prognozēšanas principiem, algoritmiem un praksi;
3. Apkopot datus un izveidot datu noliktavu;
4. Pielietot dimensiju reducēšanas algoritmus un analizēt iegūtos rezultātus;
5. Veikt pasažieru skaita prognozēšanu un izveidot pasažieru ierašanās profilu katram reisam;
6. Izveidot kopējo drošības kontrolpunkta pasažieru skaita grafiku noteiktā laikā.

Ir nepieciešams izteikt prognozi pasažieru skaitam katram reisam un izveidot konkrēta reisa pasažieru ierašanās profilu aviācijas drošības kontrolpunktā, bet šo visu reisu pasažieru ierašanās profilu apvienošana izveidotu kopējo drošības kontrolpunkta pasažieru skaita grafiku noteiktā laikā. Pasažieru ierašanās profils ir grafiks, kurš raksturo pasažieru ierašanos drošības kontrolpunktā pirms lidojuma.

Pārveidojot šo grafiku tiktu izveidotas vadlīnijas drošības kontrolpunkta līniju atvēršanas un personāla resursu plānošanai.

Efektīva darbība kontrolpunktā ir atkarīga no pareizā līniju skaita un to atvēršanas pareizajā laikā, kā arī ar atbilstošu drošības pārbaucēju inventāra esamību uz vietas (rentgena iekārtas, rokas skeneri u.c.). Pēdējos gados drošības darbiniekiem ir pieejami arvien vairāk datu – iekāpšanas karšu skenētie dati, gaidīšanas laiki rindā un dati no rentgena iekārtām, un tieši šie dati spētu palīdzēt izstrādāt labākas pasažieru plūsmas prognozes, kā rezultātā spētu plānot drošības kontrolpunktu un to līniju darbību.

Risinājuma mērķis ir efektīvi plānot drošības kontrolpunktu darbību, nesamazinot cilvēkresursu efektivitāti. Rezultātā tiktu samazināts pasažieru gaidīšanas laiks uz drošības pārbaudes kontrolpunktu, tādējādi uzlabojot pakalpojumu, lai klienti vairāk laika pavadītu sterilajā sektorā, nevis drošības kontrolpunkta gaidīšanas rindā

Maģistra darba struktūra. Darbs sastāv no ievada, četrām nodaļām, rezultātiem un secinājumiem, idejām par darba uzlabošanu nākotnē, izmantotās literatūras sarakstu. Pirmajā nodaļā tiek aprakstīts Kopenhāgenas lidostā izmantotais modelis, kura idejas ir izmantotas šajā darbā. Otrajā nodaļā analizēta literatūra par prognozēšanas uzdevumu pamata darbībām. Trešajā nodaļā tiek aprakstīta datu apkopošana un aprakstītas iegūtās tabulas, literatūra par datu pirmapstrādi. Šajā nodaļā tiek aprakstīts, kā tiek pielietoti dimensiju reducēšanas algoritmi (*PCA* un *T-SNE*). Vēl šajā nodaļā tiek analizēti secinājumi no dimensiju reducēšanas algoritmu pielietošanas, tiek apstiprinātas vai noraidītas iegūtās likumsakarības un atziņas ar datu vizualizācijas programmatūras *Power BI* palīdzību. Ceturtajā nodaļā tiek aplūkotas pasažieru skaita prognozēšanas metodoloģijas, pamata darbības, laika rindu apstrāde un analīze, prognozēšanas metodes. Šajā nodaļā tiek aprakstīts izveidotais prognozēšanas modelis *Python* programmēšanas valodā. Prognozētajiem rezultātiem tiek pievienoti izstrādātie pasažieru ierašanās profili un izveidots kopējs drošības kontrolpunkta pasažieru skaita grafiks noteiktā laikā. Darba nobeigumā ir aprakstīti iegūtie rezultāti un veikti secinājumi, kā arī izteikti priekšlikumi modeļa uzlabošanai un tālākai darba tālākai attīstībai nākotnē.

1. KOPENHĀGENAS LIDOSTĀ IZMANTOTAIS MODELIS

Par darba pamatu ir ņemts organizācijas *COPENHAGEN OPTIMIZATION* prezentācija – Labāka prognoze un Labāka drošība (*Better Forecast and Better Security*). Kurā tiek piedāvāts, kā ir iespējams veikt labāku drošības kontrolpunkta pasažieru plūsmas plānošanu visā diennakts garumā un efektīvu cilvēku resursu sadali atkarībā no drošības kontrolpunkta pārbaužu līniju skaita un pasažieru plūsmas intensitātes [3].

Šis modelis izklāsta to, ka labāka pasažieru apkalpošanas pieredze rodas precīzi plānojot drošības kontrolpunkta noslogojumu un laicīgu, nepieciešamajā skaitā pārbaužu līniju (pilnībā aprīkota darba vieta ar visu nepieciešamo tehnisko nodrošinājumu (rentgena aparāts, rokas skeneri u.c.) un personāla resursiem (drošības departamenta darbiniekiem) drošai pasažieru pārbaudei pirms lidojuma) atvēršanu, lai pēc iespējas ātrāk un kvalitatīvāk tiktu apkalpoti klienti pārbaudes punktā ievērojot uzņēmējdarbības uzstādītos mērķus un kvalitātes nodrošināšanas prasības.

Prognozēšana koncentrējas uz periodu no šī brīža līdz noteiktiem mēnešiem nākotnē – jeb periodu, kurā ir izveidots zināms lidojumu grafiks. Šis mēnešu skaits var būt atkarīgs no lidostas lieluma, un aviokompāniju pieņemtajiem grafikiem, kā arī ir iespējama sadarbība ar jaunām aviokompānijām.

Prognozēšanas rezultāts ir katra atsevišķa lidojuma prognoze, ko pēc vajadzības var apkopot, lai izveidotu kopējo prezentācijas profilu drošības kontrolpunktam.

Nākotnes lidojumu saraksta piemērs redzams 1.1. att.

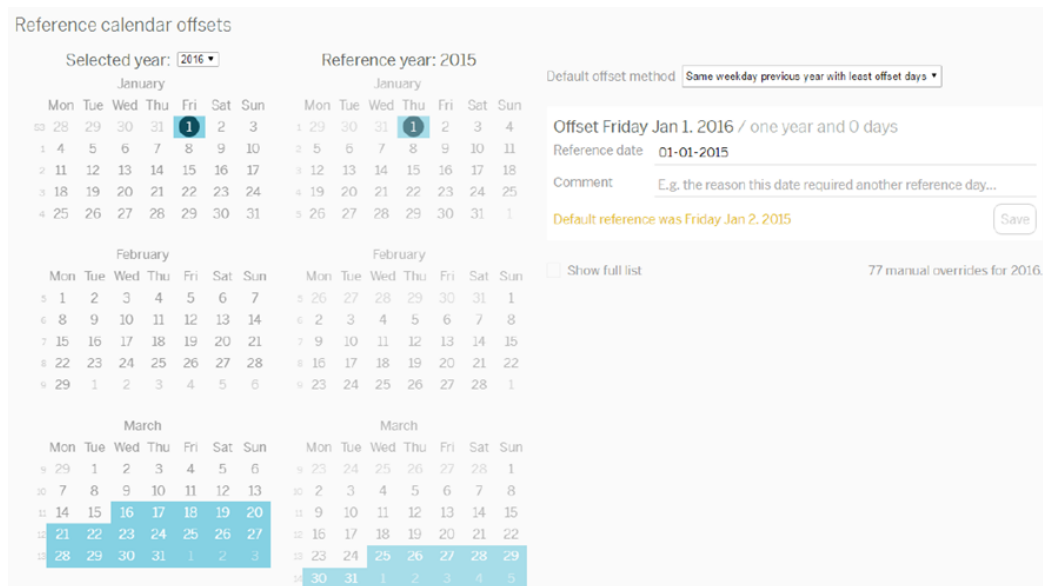
STAD	Flight day	Destination	Destination	A/C Type	Handler	Terminal	ETAD	Runway	Stand	Belt	Block hour	Registration	CodeShare all	Flight	Company
16:45	2020/01/08	ZRH*	Zurich	BCS3		B	16:52	17:02	106		16:52	YLCSB		BT643	AirBaltic
16:50	2020/01/08	MUC*	Munich	BCS3		B	16:51	17:00	219		16:51	YLCSG	IB1471 AS3263 TP6415 EV9081	BT223	AirBaltic
17:05	2020/01/08	KEF*	Reykjavik Keiflav	A320		C	17:05	17:10	302		17:00	HALYM		W62597	Wizzair
17:05	2020/01/08	VIE*	Vienna	BCS3		B	17:05	17:12	105		16:56	YLAAT	JU7862 OS7030 TP6432	BT433	AirBaltic
17:30	2020/01/08	LTN*	LondonLTN	A320		C	17:30	17:40	304		17:23	HALYI		W62503	Wizzair
18:05	2020/01/08	HEL*	Helsinki	AT72		B	18:05	18:08	103		18:04	OHATO		AY1076	Finnair
18:05	2020/01/08	HAM*	Hamburg	DH8D		C	18:05	18:06	303		17:58	YLBAA		BT253	AirBaltic
18:10	2020/01/08	BLL*	Bilund	DH8D		B	18:10	18:21	106		18:10	YLBAA		BT147	AirBaltic
18:10	2020/01/08	OSL*	Oslo/OSL	BCS3		B	18:10	18:18	104		18:08	YLCSK	SK9617	BT153	AirBaltic
18:15	2020/01/08	WAW*	Warsaw	DH8D		B	18:15	18:24	105		18:12	YLBAF	EY8112 LO5784	BT467	AirBaltic
18:15	2020/01/08	KBP*	Kyiv/Boryspil	DH8D		C	18:17	18:33	101		18:17	YLBET	PS9404	BT494	AirBaltic
18:20	2020/01/08	WAW*	Warsaw	CRJ9		B	18:20	18:23	109		18:16	ESACJ	BT5193	LO702	Polish Airlines
18:20	2020/01/08	VNO*	Vilnius	DH8D		B	18:20	18:29	110		18:18	YLBWW		BT347	AirBaltic
18:20	2020/01/08	TXL*	Berlin/TXL	BCS3		B	18:20	18:26	102		18:15	YLAQA	JU7860	BT213	AirBaltic

1.1. att. Nākotnes lidojumu saraksts

Lidojumu saraksts tiek izmantots, lai veidotu attiecīgās pasažieru plūsmas prognozes konkrētai dienai. Darba izpildē jācenšas šo nākotnes lidojumu sarakstu automātiski integrēt, lai šie dati būtu nepārtraukti, piemēram, izmantojot lidostas operāciju datubāzi (*AODB*).

1.1. Realizētais lidojumu grafiks

Tiek izmantoti vēsturiskie dati katram lidojumam konkrētajā dienā pagājušā gadā. Ņemot vērā brīvdienu periodu maiņu, ir nepieciešama attiecīgās dienas kartēšana iepriekšējos gados. Zemāk redzamajā 1.2. att. ir parādīta atsauču dienu kartēšana.



1.2. att. Atsauču dienu kartēšana [3]

Pēc noklusējuma kartējums ir uz to pašu nedēļas dienu iepriekšējā gadā. Zilās iezīmētās dienas norāda atšķirību, kas ir pēc noklusējuma, piemēram, 1. janvāris parasti tiek kartēts ar vienu un to pašu datumu, bet brīvdienu periodi tiek kartēti pamatojoties uz attiecīgajiem brīvdienu periodiem iepriekšējā gadā.

1.2. Realizētais lidojumu saraksts (pēdējais periods)

Slodzes koeficienta palielināšana tiek veikta, pielāgojot prognozi ar neseno slodzes koeficienta attīstību – parasti pēdējo četru līdz divpadsmit nedēļu laikā, ko nosaka kalibrēšana. Šajā nolūkā tiek izmantoti pēdējās nedēļās realizētie satiksmes grafiki. Slodzes koeficienta tendences piemērs parādīts zemāk (skat. 1.3. att.).

Tendency impact 16.7%						
Based on 6 weeks.						
Date	Flights	Load factor	Reference day	Reference flights	Reference load factor	Diff load factor
2016-10-03	4	98.1%	2015-10-05	4	78.4%	25.1%
2016-10-10	4	96.5%	2015-10-12	4	76.3%	26.4%
2016-10-17	4	99.8%	2015-10-19	4	75.4%	32.3%
2016-10-24	3	88.9%	2015-10-26	3	78.1%	13.8%
2016-10-31	4	69.8%	2015-11-02	4	75.4%	-7.4%
2016-11-07	3	84.8%	2015-11-09	3	77.3%	9.7%
					Average	16.7%

1.3. att. Slodzes faktora tendenču ietekme [3]

1.3. Pasažieru skaita prognoze

Pasažieru skaita prognoze tiek veikta izmantojot atsauces dienas, kuras ir noteiktas pamatojoties uz iepriekšējā gada vēsturiskajiem datiem. Tiem tiek piemērots slodzes koeficients ar neseno pasažieru plūsmas skaita izmaiņu informāciju. Nākamais solis ir pasažieru sadalījums jeb pasažieru ierašanās profils atkarībā no paredzētā lidojuma, ņemot vērā viņu ierašanos drošības kontrolpunktā.

1.4. Pasažieru ierašanās profila prognoze

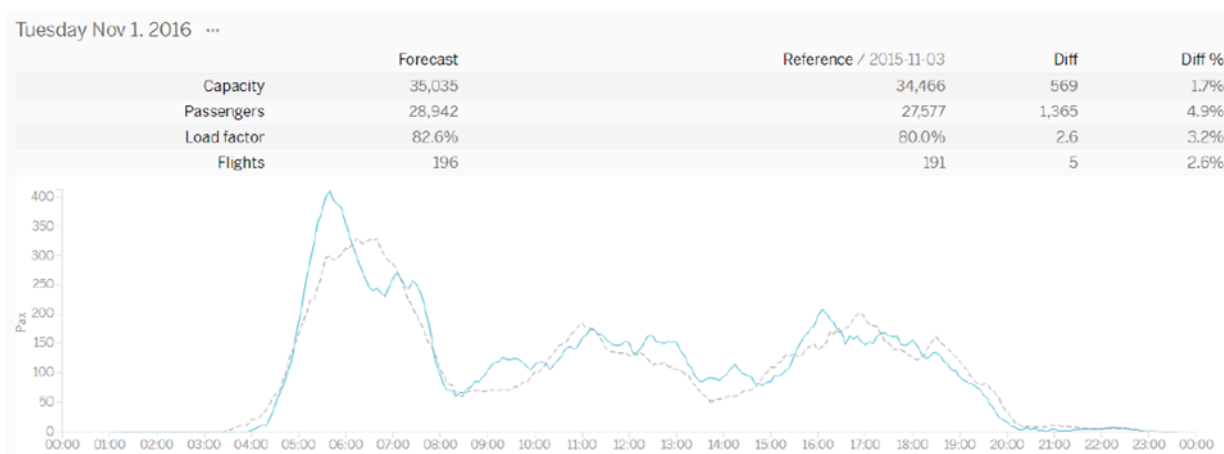
Pasažieru prezentācijas profili ir ievades dati prognozēšanai (vispārīgu profilu gadījumā, kas nav saistīts ar vienu konkrētu lidojumu), vai radīti saskaņā ar prognozi, izmantojot vēsturisko informāciju par pasažieriem (pamatojoties uz iekāpšanas karšu skenēšanas datiem). Pasažieru prezentācijas profili tiek veidoti, izmantojot iekāpšanas karšu skenēšanas datus no pēdējām sešām līdz divpadsmit nedēļām. Kalibrēšana nosaka precīzu piemērojamo nedēļu skaitu.

Gadījumos, kad gaidīšanas laiks var notikt pirms iekāpšanas kartes skenēšanas, pasažieru laika zīmogs (*timestamp*) ir jāpielāgo jebkuram gaidīšanas laikam. Turklāt, tā kā tiek izmantoti operatīvie dati, tad prognozēšana apstiprina datus, atklāj novirzes. Turklāt lietotājs var norādīt dienas, kuras iekāpšanas kartes ir jāatmet, piemēram, dienas ap Ziemassvētkiem vai Lieldienām. Zemāk redzamajā (skat. 1.4. att.) parādīts piemērs, kas balstīts uz vienu lidojumu.



1.4. att. Pasažieru ierašanās skaita profils – viens lidojums [3]

Diagramma (skat. 1.4. att.) kreisajā pusē parāda pasažieru skaita ierašanos drošības kontrolpunktā pirms lidojuma, bet labajā pusē esošā diagramma parāda drošības kontrolpunkta noslodzes grafiku konkrētā laika periodā attiecīgajam lidojumam. Konkrēta lidojuma informācija ir pieejama zem diagrammām, norādot detalizētu informāciju par paredzamo lidojumu un atsauces lidojumu. Katra lidojuma pasažieru ierašanās profili tiek uzkrāti, ģenerējot kopējo pasažieru ierašanās profilu drošības kontrolpunktam. Tas ir parādīts zemāk (skat. 1.5. att.).



1.5. att. Pasažieru skaita ierašanās profili drošības kontrolpunktam [3]

Kopējais pasažieru prezentācijas profils (skat. 1.5. att., zilā līnija) tiek salīdzināts ar pasažieru prezentācijas profilu atsauces dienā (1.5. att., pelēkā līnija).

Prognozēšanas sadaļā ir pieejami daži papildus līdzekļi:

- a) Notikumu izsekošana – lieli notikumi var ievērojami ietekmēt slodzes koeficientu. Notikumu izsekošana nav daļa no prognozēšanas, bet tā ietver funkciju, kas

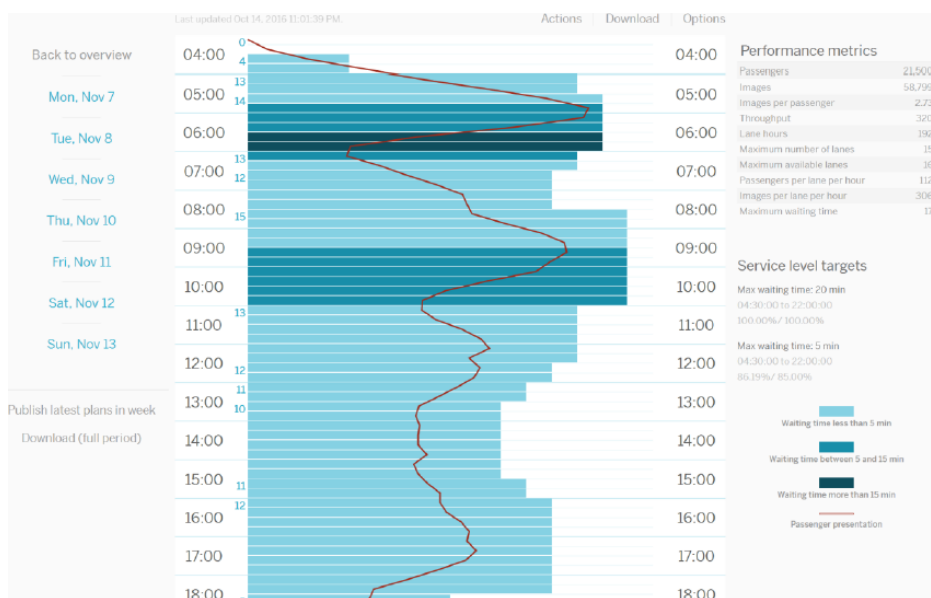
Diagrammā (uz horizontālās ass attēloti gada nedēļas numuri), attēls uz vienu pasažieri ir visaugstākais ziemas periodā un darba dienās. Tas atbilst tam, ka saskaņā ar Eiropas lidostu prognozēm, kur pasažieri ziemas periodā pārvadā vairāk apģērba. Darba dienās ir redzams, ka vairāk ir biznesa ceļotāju, kuri rada vairāk attēlu ar klēpj datoriem, planšetdatoriem un šķidrumiem, salīdzinot ar ceļotājiem, kuri dodas atpūsties.

1.6. Līniju pieprasījumu optimizācija

Lietojot matemātisku algoritmu, lai ģenerētu drošības kontrolpunkta līniju atvēršanas plānu, izmanto šādus ievades parametrus:

- pasažieru prognozes;
- pasažieru ierašanās profilus;
- caurlaidību - – maksimāli iespējamais pasažieru apkalpošanas skaits vienā drošības kontrolpunkta pārbaudes līnijā;
- gaidīšanas laika mērķus – maksimāli pieļaujamais pasažiera gaidīšanas laiks rindā uz drošības kontrolpunktu.

Balstoties uz šiem ievades parametriem, labākas plānošanas algoritms ģenerē joslu atvēršanas plānu. Algoritma mērķis ir samazināt maksimālo joslu pieprasījumu, ļaujot pīķa jeb intensīvākajos pasažieru plūsmas periodos apkalpot pasažierus, izpildot gaidīšanas laika mērķus. Līniju atvēršanas plāna piemērs ir parādīts zemāk redzamajā 1.7 att.



1.7. att. Kontrolpunkta līniju atvēršanas plāns [3]

Diagrammas joslas ilustrē atveramo līniju skaitu, kreisajā pusē tiek norādīts aktuālais skaitlis (iekrāsots zilā krāsā). Zilais numurs tiek parādīts tikai tad, ja ir notikušas izmaiņas. Jo

tumšāka ir zilā krāsa joslās, jo ilgāks ir aptuvenais gaidīšanas laiks. Sarkanā līnija attēlo pasažieru prezentāciju.

1.7. Personāla pieprasījuma optimizācija

Kad ir izveidots līniju atvēršanas plāns, izmantojot ievades datus, tiek izveidots personāla pieprasījuma plāns, kas sastāv no šādiem parametriem [3]:

- a) personāla buferi (papildus darbinieki, kas dežurē neparedzētu atvaļinājumu gadījumā, piemēram, slimības);
- b) personāla pārtraukumu shēma;
- c) statiskās amata pozīcijas (fiksētas personāla prasības, kas nemainās neatkarīgi no pasažieru skaita);
- d) personāla daudzums vienā līnijā (atkarība no līniju īpašībām).

Pārtraukuma optimizācijas algoritms ieplāno pārtraukumus tā, lai kopējais darbinieku pieprasījums tiktu samazināts [3].

2. PROGNOZĒŠANAS UZDEVUMA PAMATA DARBĪBAS

Šīs nodaļas izklāsts balstīts uz avotu [4].

Prognozēšanas uzdevums parasti ietver piecus pamata soļus – problēmas definīciju, informācijas apkopošanu, izpētes analīzi, modeļa izvēli un uzstādīšanu, prognozēšanas modeļa izmantošanu un novērtēšanu.

2.1. Problēmas definīcija.

Bieži vien šī ir visgrūtākā prognozēšanas daļa. Rūpīgi definējot problēmu, ir jāsaprot, kā tiks izmantotas prognozes, kam vajadzīgas prognozes un kā prognozēšanas funkcija iekļaujas organizācijā, kurai nepieciešamas prognozes. Prognozētājam ir jāpavada laiks, runājot ar visiem, kas būs iesaistīti datu vākšanā, datu bāžu uzturēšanā un prognožu izmantošanā turpmākajai plānošanai.

2.2. Informācijas apkopošana.

Nepieciešamā informācija vienmēr ir vismaz divu veidu:

- a) statistikas dati;
- b) cilvēku uzkrātā pieredze, kuri apkopo datus un izmanto prognozes.

Bieži varētu būt grūti iegūt pietiekami daudz vēsturisko datu, lai tie atbilstu modelim. Tādā gadījumā var izmantot vērtēšanas prognozēšanas metodes. Reizēm vecie dati būs mazāk noderīgi prognozējamo sistēmu strukturālo izmaiņu dēļ.

2.3. Izpētes analīze.

Izpēte jāskatās ar grafika izveidi. Jāizpēta vai pastāv konsekventi modeļi, vai ir kādas būtiskas tendences, vai sezonālitate ir svarīga, vai ir pierādījumi par uzņēmējdarbības cikla esamību, vai datos ir kādi izteikti rādītāji, kas ir jāizskaidro tiem, kuriem ir ekspertu zināšanas, cik stipras ir analīzei pieejamo mainīgo attiecības. Lai varētu veikt šo analīzi ir jāizmanto dažādi rīki.

2.4. Modeļu izvēle un uzstādīšana.

Vislabākais izmantojamais modelis ir atkarīgs no vēsturisko datu pieejamības, saistību stipruma starp prognozējamo mainīgo un visiem skaidrojošajiem mainīgajiem lielumiem un prognozēšanas izmantošanas veida. Parasti tiek salīdzināti divi vai trīs potenciālie modeļi. Katrs

modelis ir pats par sevi mākslīga konstrukcija, kas balstās uz pieņēmumu kopumu un parasti ietver vienu vai vairākus parametrus, kuri ir jānovērtē, izmantojot zināmos vēsturiskos datus.

2.5. Prognozēšanas modeļa izmantošana un novērtēšana.

Kad modelis ir izvēlēts un novērtēti tā parametri, modeli izmanto, lai izveidotu prognozes. Modeļa darbību var pienācīgi novērtēt tikai pēc tam, kad būs pieejami dati par prognozēto periodu. Ir izstrādātas vairākas metodes, lai palīdzētu novērtēt prognožu precizitāti, piemēram novērtētāja vidējā kvadrāta kļūda (*MSE*).

3. DATU PIRMAPSTRĀDE UN ANALĪZE

Šajā darbā ir nepieciešams prognozēt un radīt jaunus datus, tāpēc tiek izmantota datizrāce, Tiek izpētīta datu iegūšanas terminoloģija, rīki, datu pirmapstrāde.

Datizrāce (*data mining*) ir process, kura rezultātā automātiski tiek iegūta noderīga informācija no datubāzēm. Citādi var teikt, ka tā ir zināšanu iegūšana no lielām datu bāzēm, kurās ir lieli datu apjomi. Datu iegūšanas tehnoloģijas tiek pielietotas, lai varētu atrast jaunus un noderīgus paraugus [9].

Piemēram, ja datu iegūšanas procesā tiek iegūta noderīga informācija, tad šo informāciju uzskata par zināšanām, kas tiek iegūtas pētot un analizējot datus. Šādu datu iegūšanas definīciju dēvē par *zināšanu atklāšanas no datu bāzēm procesu* [9]. Šis process sastāv no šādiem soļiem [9]:

1. *Datu pirmapstrāde* – kā arī integrācija, kas viss kopā ir reālu datu bāzu pārveidošana par datu noliktavu.
2. *Datu transformācija* – pielietota, lai sagatavotu datus to analīzei ērtā formā, lai ar tiem varētu strādāt jeb tā, lai tos varētu saprast programmlīdzekļi.
3. *Datizrāce* – process, kura rezultātā tiek iegūti modeļi un paraugi, izmantojot algoritmus vai citas intelektuālas pieejas.
4. *Novērtējums un interpretācija* – procesa nobeiguma soļi, visi iegūtie rezultāti ir jānovērtē, jāidentificē un jāinterpretē. Jācenšas iegūt jaunus un noderīgus paraugus, lai tie kļūtu par zināšanām.

Mūsdienu pasaulē ar datu iegūšanas palīdzību tiek risinātas problēmas, kurās tiek izmantotas ļoti lielas datu kopas, kas bieži nāk no dažādiem avotiem un var saturēt trokšņainus, trūkstošus vai pretrunīgus datus. Šādu datu rašanās iemesli var būt dažādi. Piemēram, dati var būt nekorekti savākti vai apkopoti cilvēku vai tehnoloģisku iemeslu dēļ. Var rasties dažādas atšķirības pielietotajos nosaukumos vai kodos, vai datu formātos. Datu analizētājam ne vienmēr var būt pieejami interesējošie atribūti. Datu iegūšanā, lietojot šādus nekorektus datus var rasties kļūdaini rezultāti, tāpēc no sākuma tiek veikta datu pirmapstrāde. Kopumā pirmapstrādes process var aizņemt līdz pat 80% no visa datu iegūšanas procesa izstrādāšanas laika [9].

Darbā tiek apkopotas un izmantotas trīs datu kopas:

- a) “*Departures*” – lidojumu saraksts;
- b) “*Pax_scan*” – skenētās pasažieru iekāpšanas kartes;
- c) “*Capacity*” – Gaisa kuģu tipi un to kapacitātes tabula.

Šajā gadījumā stabilas, no iegūtajām datu kopām ir lidojumu saraksts un gaisa kuģu kapacitātes tabula, Tās ir datu kopas, kas ir iegūtas no lidostas operatīvās lidojumu datu bāzes, kur atlasīti ir tikai izpildītie reisi. No šīs datu kopas tiek ievākta informācija par gaisa kuģiem ar kuriem tiek izpildīti reisi, tādējādi arī gaisa kuģu kapacitātes tabulu var uzskatīt par precīzu datu avotu. Turpretī pasažieru iekāpšanas karšu skenētie dati ir iegūti tehnoloģiska procesa rezultātā, kur nekorektu datu iegūšanu var izraisīt gan cilvēciskais faktors – noskenēta nederīga iekāpšanas karte, vai ar citu reisa izlidošanas datumu, vai iekāpšanas karte tiek noskenēta divreiz. Kā arī varbūt ir notikusi tehniskas rakstura problēma, karte tiek noskenēta korekti, bet e-vārti netiek atvērti, kā rezultātā pasažierim jānododas uz citiem e-vārtiem u.c. gadījumi.

Pastāv dažādas pirmapstrādes tehnoloģijas. Datu attīrīšana tiek izmantota, lai attīrītu datu kopu no trokšņainiem un pretrunīgiem datiem. Datu transformācija var uzlabot dažādu datu ieguves algoritmu darbību, piemēram, izslēdzot dominēšanu. Datu integrācija ir dažādu avotu apvienošana vienā datu bāzē. Datu reducēšana var samazināt datu kopas apjomu, tādējādi uzlabojot algoritmu ātrdarbību [9].

3.1. Datu transformācija

Datu transformāciju veic, lai uzlabotu datu iegūšanas procesa precizitāti un efektivitāti. Piemēram, novēršot kādas datu daļas dominēšanu, veicot jaunu atribūtu izveidi, vai tieši otrādi – samazinot to [10].

Galvenie datu atlases un attīrīšanas procesi ir šādi [10]:

- a) datu apkopošana;
- b) paraugu ņemšana;
- c) dimensiju skaita samazināšana;
- d) pazīmju apakškopas atlase;
- e) pazīmes radīšana;
- f) diskretizācija un binarizācija;
- g) mainīgo transformācija.

3.1.1. Dimensiju skaita samazināšana

Samazinot dimensiju skaitu, var paātrināt datu iegūšanas algoritmu pielietošanu, jo pārāk liels datu kopas apjoms datu iegūšanā var aizņemt ļoti daudz laika. Dimensiju skaits tiek samazināts, lai no datu kopas iegūtu tādu apakškopu, kura pēc apjoma ir daudz mazāka, kā arī šīs kopas integritāte ir līdzīga sākotnējai datu integritātei, t.i. uz apakškopas pamata datu iegūšanas rezultātiem jābūt līdzīgiem ar sākotnējo datu kopu [9].

3.1.2. Pazīmes radīšana

No sākotnējiem atribūtiem bieži vien iespējams izveidot jaunu atribūtu kopu, kas datu kopā uztver svarīgo informāciju daudz efektīvāk. Turklāt jauno atribūtu skaits var būt mazāks nekā sākotnējo atribūtu skaits, ļaujot izmantot visas iepriekš aprakstītās dimensiju samazināšanas priekšrocības [9].

3.1.3. Mainīgo transformācija

Mainīgo transformācija attiecas uz transformāciju, kas tiek piemērota visām mainīgo vērtībām. Citiem vārdiem sakot, katram objektam transformācija tiek attiecināta uz šī objekta mainīgā vērtību. Piemēram, ja tas ir svarīgs mainīgā lielums, mainīgā lielumu var pārveidot, ņemot absolūto vērtību. Pastāv divi svarīgi transformāciju veidi: vienkāršas funkcionālas transformācijas un normalizācija [10].

Normalizācijas galvenais uzdevums ir izvairīties no situācijām, kad viens vai vairāki atribūti var dominēt, jo atribūtu vērtību intervālu diapazoni mēdz būt dažādi, kā, piemēram, atribūtiem "Vecums un ienākumi". Pastāv dažādi normalizācijas algoritmi, un zemāk tiks apskatīti daži no tiem [9].

Z - novērtējuma normalizācijā ar standartnovirzi, skat. formulu (3.1), vērtības tiek normalizētas, izmantojot atribūta vidējo aritmētisko vērtību un standartnovirzi, skat. formulu (3.2), lai aprēķinātu *a* vērtības a_i normalizēto vērtība a'_i izmanto formulu [9].

$$\delta_A = \sqrt{\frac{1}{n-1} \cdot \sum_{i=1}^n (x_i - \bar{x})^2}, \quad (3.1)$$

$$a'_i = \frac{a_i - \bar{a}}{\sigma_a}, \quad (3.2)$$

kur \bar{a} – vidējā aritmētiskā vērtība atribūtam *a*;

σ_a – atribūta *a* standartnovirze.

Šī pieeja tiek izmantota, kad faktiskās datu kopas robežas ir nezināmas, vai kad ir daudz nepiederošu datu, kas iespējams var būt dominējoši [9].

Pieprasījuma normalizācijas ar dzīves līkni aprēķina, sk. formulu (3.3), ņemot objekta (laika rindas) perioda vērtību un dalot to ar šī objekta periodu vērtību summu laika rindā [6]:

$$y_i = \frac{x_i}{\sum_{j=1}^n x_j}, \quad (3.3)$$

- kur y_i – normalizētā vērtība laika periodā i ;
 x_i – laika rindas x vērtība laika periodā i ;
 x_j – laika rindas periodu vērtību summa;
 n – periodu skaits laika rindā.

Šī pieeja ir piemērota normalizācijai laika rindām ar vēsturisku notikumu pieprasījumu (piemēram, preču pārdošanas apjoms pa mēnešiem viena vai vairāku gadu garumā). Iegūtās vērtības tiek normalizētas tikai tām vērtībām, kuras ir pozitīvas [6].

3.2. Pasažieru iekāpšanas karšu skenētie dati.

Pēc Kopenhāgenas lidostas modeļa, lai spētu noteikt pasažieru ierašanās laiku pirms reisa ir nepieciešams apkopot un analizēt pasažieru skenētos iekāpšanas karšu datus drošības kontrolpunktā. Sākotnējā iegūta tabula sastāv no pieciem atribūtiem:

- a) “*Timestamp*” – laika zīmogs skenētajai iekāpšanas kartei, jeb brīdis, kad pasažieris ir noskenējis savu iekāpšanas karti lidostas E-vārtos, un pēc tam devies uz drošības pārbaudi kontrolpunktā.
- b) “*Workstation*” – E-vārti, caur kuriem ir izgājis pasažieris.
- c) “*Result*” – Rezultāts (pareizi (*true*) vai nepareizi (*false*)), vai noskenētā iekāpšanas karte ir derīga.
- d) “*Company*” – Aviokompānijas identifikācijas numurs.
- e) “*FlightNo*” – reisa numurs.

Sākotnēji bija apkopoti dati par 9873429 skenētās pasažieru kartēm laika posmā no 2016. – 2019. gadam. Dati bija jāapstrādā jeb jāatlasa derīgie, jo daudziem ierakstiem “*Result*” kolonā bija dažādas vērtības. Vienīgā un pieņemamā vērtība ir “*OK*”, kas nozīmē to, ka noskenētā pasažiera iekāpšanas karte ir derīga un pasažieris ir izgājis cauri *E-vārtiem*. Tas nozīmē to, ka ,piemēram, ieraksti, kuru “*Result*” vērtība ir “*UnrecognizedData*”, “*WrongDepartureAirport*”, u.c. ir jāatfiltrē, tādējādi par derīgiem tika uzskatīti 95,56% no kopējiem ierakstiem, bet nederīgi bija 4,44% ierakstu.

Šajā datu kopā tika apvienotas divas kolonas - “*Company*”, piemēram ar vērtību “*BT*” un “*FlightNo*”, piemēram ar vērtību “*617*”. Tas ir nepieciešams, lai varētu šo datus sasaistīt ar “*Departures*” datu tabulu, jo šajā datu tabulā jau šie divi atribūti ir apvienoti un iegūts kopējais reisa numurs, piemēram reisa numurs “*BT617*”. Tas tika panākts ar izklājlapu programmas Excel formulas palīdzību:

CompanyFlightNo = CONCATENATE (Company, “ ”, FlightNo)

Šajā datu tabulā vēl tika reducētas nevajadzīgās dimensijas – “*Workstation*”, jo pasažieri stāv vienotā rindā uz *E-vārtiem*, tāpēc nav svarīgi, caur kuriem iziet pasažieris, “*Company*” un “*FlightNo*”, jo šie abi atribūti tika apvienoti vienā, augstāk minētajā “*CompanyFlightNo*”

Pēc datu apstrādēs tika aprakstīti datu tipi datu bāzē *Microsoft SQL Server 2019* zemāk redzamajā 3.1. tabulā.

3.1. tabula

Pasažieru iekāpšanas karšu skenēto datu atribūtu apraksts

Nosaukums	Datu tips
Timestamp	smalldatetime
CompanyFlightNo	varchar

Tiek izmantota datu bāze, jo tā ievērojami atvieglo un paātrina tālāko datu apstrādi.

3.3. Izlidojošo reisu saraksts

Izlidojošo reisu saraksts no 2016. – 2019. gadam sastāv no 142 atribūtiem un 159195 ierakstiem. Šis saraksts ir nepieciešams, lai varētu prognozēt to cik pasažieru konkrētajā dienā ieradīsies uz drošības pārbaudi. Jo pēc tam šo datu tabulu būs nepieciešams sasaistīt gan ar gaisa kuģu tipu un to kapacitātes tabulu, gan ar skenēto pasažieru iekāpšanas karšu datiem.

Šai datu tabulai tiek reducētas dimensijas, un atstātas tikai nepieciešamās:

- a) *departure_ID* – izlidojošā reisa identifikācijas numurs;
- b) *departure_flight_number* – izlidojošā reisa numurs;
- c) *schedule_departure* – reisa izlidošanas laiks;
- d) *ac_type_id* – gaisa kuģa tipa identifikators;
- e) *company_id* – aviokompānijas identifikators jeb aviokompānija;
- f) *destination* – reisa galamērķis.

Pēc datu apstrādes tabula tika importēta datu bāzē *Microsoft SQL Server 2019*, aprakstīti datu tipi zemāk redzamajā 3.2. tabulā:

3.2. tabula

Izpildīto lidojumu datu tabulas atribūtu apraksts

Nosaukums	Datu tips
departure_ID	float

departure_flight_number	nvarchar
schedule_departure	smalldatetime
ac_type_id	nvarchar
company_id	nvarchar
destination	nvarchar

Importējot tabulu datu bāzē, tā ir jāturpina apstrādāt, jo apkopotie dati ir pēc *UTC* (saskaņotais pasaules laiks) laika, bet pasažieru skenētās iekāpšanas karšu dati pēc vietējā (*GMT +2*) laika. Tāpēc visai datu kopai tika pielietots SQL vaicājums, kurš pieskaita visiem ierakstiem *schedule_departure* divas stundas.

```
UPDATE dbo.departures set schedule_departure = DATEADD(HOUR,2,
schedule_departure)
```

Bet tā kā Latvijā ir ziemas un vasaras laiks, tad katram gadam ir nepieciešams pieskaitīt vēl vienu stundu vasaras laikam. Šeit tika izveidoti četri SQL vaicājumi, jo datumi, kad notiek pāreja uz vasaras un atpakaļ uz ziemas laiku atšķiras. Piemērs SQL vaicājumam par 2016. gadu atrodas zemāk.

```
UPDATE dbo.departures set schedule_departure = DATEADD(HH, 1,
schedule_departure)
```

```
Where schedule_departure > '2016-03-27 3:00:00' and schedule_departure < '2016-
10-30 4:00:00'
```

3.4. Gaisu kuģu tipu un to kapacitātes datu tabula.

Apkopotā gaisa kuģu tipu un to kapacitātes datu tabula sastāv no diviem atribūtiem:

- a) ACtype – gaisa kuģa tips;
- b) Capacity – gaisa kuģa pasažieru ietilpība.

Kopā ir 43 gaisa kuģu tipi, bet interesanti, ka katrai aviokompānijai pie viena gaisa kuģu tipa ir dažādas ietilpības.

Šī datu tabula tiks izmantota, lai varētu noteikt to, cik lielā mērā gaisa kuģis parasti ir aizpildīts, lai prognozējot nepārkāptu maksimāli pieļaujamo aizpildījumu (noslodzi) – 100%.

3.5. Jaunu datu tabulu un atribūtu izveide.

Lai varētu aprēķināt kāds bija pasažieru skaits konkrētā reisā, un varētu aprēķināt

pasažieru ierašanās laikus drošības kontrolpunktā pirms lidojuma, ir nepieciešams izveidot

3.5.1. Atribūta “Ierašanās” izveidošana.

Šis atribūts ir nepieciešams, lai noteiktu to cik minūtes pirms reisa ierodas pasažieris drošības kontrolpunktā. To var izdarīt no konkrētā reisa izlidošanas laika atņemot attiecīgos skenēto pasažieru iekāpšanas karšu datus dotajam reisam. Tā kā šie divi atribūti atrodas dažādās datu tabulās, tāpēc tiek izveidots vaicājums, kurš sasaista izlidojošā reisa numuru un datumu ar attiecīgo skenēto pasažieru karšu datu laikiem (*Timestamp*) un reisa numuriem (*CompanyFlightNo*). Diemžēl dotajos datos nav iespējams izveidot saites, jo nav nevienas arējās atslēgas, kur varētu piesaistīt skenētās pasažiera kartes ar konkrēto reisu.

Tabulā “*departures*” tiek ņemti atribūti “*departure_flight_number*” un “*schedule_departure*” un tiek sasaistīti (*inner join*) ar “*pax_scan*” tabulas atribūtiem “*Timestamp*” un “*CompanyFlightNo*”. Tādējādi tiek sasaistīts konkrētais reiss ar tā pasažieriem. Kļūdoties šeit nav iespējams, jo katrs reiss notiek vienu reizi dienā, tādēļ tam piesaistīties var tikai tās dienas skenētās pasažieru iekāpšanas kartes, jo reisi pēc 00:00 nenotiek, vienīgais izņēmums ir reiss ar numuru “*SU2119*”, kura plānotais izlidošanas laiks visbiežāk ir plkst. 02:50, ņemot vērā to, ka sabiedriskais transports naktīs nekursē, tādēļ ir ļoti liela iespējamība, ka šī reisa pasažieri var ierasties iepriekšējā datumā, tāpēc šo reisu ir jāapskata atsevišķi. Citādi nevar sanākt situācijas, kad reisa un tā skenētās iekāpšanas kartes varētu sastāvēt no diviem datumiem, bet tomēr nevar izslēgt situāciju, kad tā var gadīties, tādēļ autora ieteikums būtu, ievācot skenēto pasažieru karšu datus, uzreiz no skenētās kartes nolasīt attiecīgo reisu un tā izlidošanas laiku, tādējādi tas ne vien atvieglotu datu apstrādi, bet arī uzlabotu datu kvalitāti. Izstrādāto vaicājumu un iegūto tabulu var aplūkot zemāk (skat. 3.1. att.).

```
Select s.CompanyFlightNo, d.schedule_departure, s.Timestamp, d.Destination,
d.company_id, d.ac_type_id from dbo.departures d
inner join dbo.pax_scan s on s.CompanyFlightNo = d.departure_flight_number and
CONVERT (date, d.schedule_departure) = CONVERT (date, s.Timestamp)
Order by schedule_departure
```

	CompanyFlightNo	schedule_departure	Timestamp	Destination	company_id	ac_type_id
1	BT418	2016-01-01 06:00:00	2016-01-01 04:43:00	Moscow/Vnukovo	Air Baltic Corporation	DH4
2	BT601	2016-01-01 07:05:00	2016-01-01 05:53:00	Brussels	Air Baltic Corporation	DH4
3	BT601	2016-01-01 07:05:00	2016-01-01 06:27:00	Brussels	Air Baltic Corporation	DH4
4	BT601	2016-01-01 07:05:00	2016-01-01 06:22:00	Brussels	Air Baltic Corporation	DH4
5	BT601	2016-01-01 07:05:00	2016-01-01 06:16:00	Brussels	Air Baltic Corporation	DH4
6	BT601	2016-01-01 07:05:00	2016-01-01 05:53:00	Brussels	Air Baltic Corporation	DH4

3.1. att. “Departures” un “pax_scan” tabulu apvienojums

Kā redzams 3.1. att., tad katram reisam, kurš ir izpildīts konkrētā dienā ir piesaistīti skenēto iekāpšanas karšu dati, no šīs tabulas var aprēķināt to cik minūtes pirms reisa drošības kontrolē ierodas pasažieris. Aprēķinam tiek izmantots vaicājums:

```
SELECT [CompanyFlightNo]
      ,[schedule_departure]
      ,[Timestamp]
      ,[Destination]
      ,[company_id]
      ,[ac_type_id]
      ,DATEDIFF(minute, Timestamp, schedule_departure) as ierasanas
FROM [master_work].[dbo].[noliktava]
order by [schedule_departure]
```

3.5.2. Reisa pasažieru ierašanās profila izveide.

Kā jau 1.4. punktā tika minēts, ir nepieciešams no apkopotajiem datiem izveidot pasažieru ierašanās profilu katram lidojumam. Proti, no tabulas ierakstiem par konkrēto reisu ir nepieciešams izveidot vienu reisa ierakstu, kur tik pievienoti atribūti “Ier10” – un tā vērtība būtu pasažieru skaits, kuri ieradušies no reisa “schedule_departure” laika līdz desmit minūtēm, brīdī, kad tiek noskenēta pasažiera iekāpšanas karte, pēc tam līdzīgi “Ier20” – no 10 līdz 20 minūtēm, “Ier30”,, “Ier170”. Šī tabula tika iegūta ar zemāk redzamā vaicājuma palīdzību.

```
SELECT [CompanyFlightNo]
      ,[schedule_departure]
      ,[Destination]
      ,[company_id]
      ,[ac_type_id]
      ,COUNT(CASE WHEN [ierasanas] > 170 and [ierasanas] <= 180 THEN 1 END) AS IER170
-----
      ,COUNT(CASE WHEN [ierasanas] > 0 and [ierasanas] <= 10 THEN 1 END) AS IER10
FROM [master_work].[dbo].[Ierasanas]
```

Group by CompanyFlightNo, schedule_departure, Destination, Company_id, ac_type_id
Order by schedule_departure

Tika izveidoti 139150 reisu pasažieru profili.

3.5.3. *Reisa kopējais pasažieru skaits.*

Lai spētu prognozēt to, cik nākotnē ieradīsies pasažieri uz konkrēto reisu, ir nepieciešams vēsturiski uzzināt to, cik pasažieru ir ieradušies uz noteiktu reisu. To var aprēķināt jau no iepriekš izveidotās tabulas par reisa pasažieru profiliem, proti, sasummēt atribūtus “Ier10”,, “Ier180”, tādā veidā iegūstot atribūtu “Total”.

3.5.4. *Gaisa kuģa ietilpība.*

Iespējama var būt situācija, kad viens un tas pats reiss tiek izpildīts ar dažādiem gaisa kuģiem, no kā izriet tas, ka šo gaisa kuģu ietilpības var būt dažādas, tādējādi var atšķirties maksimāli pieejamais biļešu skaits uz konkrētu lidojumu. Tāpēc tiks arī izpētīts tas, vai kopējais pasažieru skaits uz reisu ir atkarīgs no gaisa kuģa tipa un tā kapacitātes.

Lai pievienotu šo atribūtu datu noliktavai, ar vaicājuma palīdzību, izmantojot datu bāzes operatora *inner join*, tiek sasaistīta jau esošā datu noliktava, ar gaisa kuģu tipu un to kapacitāšu tabulu. Ir situācijas, kad vienam un tam pašam gaisa kuģa tipam ir dažādas pasažieru ietilpības dažādu aviokompāniju ietvaros, tāpēc vaicājuma daļā abas tabulas tiek sasaistītas ne tikai pēc `s.ac_type_id = d.[A/c type IATA]` (gaisa kuģa tipa), bet arī `s.[company_id] = d.Company` (aviokompānijas). Iegūtais atribūts datu noliktavā tiek apzīmēts kā “Capacity”.

3.5.5. *Gaisa kuģu aizpildījums (noslodze).*

Šis atribūts tiek izveidots, jo veicot konsultācijas ar citiem Lidostas, pašreizējā pasažieru skaita prognozēšanās tiek izmantots tieši šis atribūts – vidējais gaisa kuģa aizpildījums. Tāpēc tiks pārbaudīts vai izmantojot tieši šo atribūtu ir iespējams iegūt lielāku modeļa precizitāti. Kā arī pie iegūtās prognozes ir jāpārbauda tas vai aizpildījums nepārsniedz 100%, bet arī šādas situācijas var gadīties, jo, piemēram, bērni līdz konkrētam vecumam var neuzrādīties pasažieru sarakstā, jo viņiem lidojums var būt par bezmaksas.

Atribūtu iegūst ar formulu (3.4)

$$\text{Aizpildījums} = \frac{\text{Total}}{\text{Capacity}} * 100 \quad (3.4)$$

3.5.6. Gada nedēļas numurs, gada mēneša numurs, un nedēļas dienas numurs.

Šie atribūti ir nepieciešami, lai izpētītu to vai pasažieru skaits reisos, vai to pasažieru ierašanās profilos ir vai nav atkarīgs, piemēram, no nedēļas dienas numura, nedēļas gada numura, vai ceturkšņa numura. Šie atribūti tiek iegūti no “*schedule departure*” ar datumu daļu operatoru “*Datepart*”

3.6. Dimensiju reducēšanas algoritmi

Šim darbam par pamatu ir izmantots 1. nodaļā aprakstītais Kopenhāgenas lidostas izstrādātais modelis, bet tomēr dažādās iezīmes un pasažieru paradumi katrā valstī, iespējams pat pilsētā var atšķirties, tāpēc sākotnēji no Kopenhāgenas lidostas modeļa tiek aizgūtas divas lietas:

- a) jāprognozē cik uz katru reisu ierodas pasažieri;
- b) jāizveido reisa pasažieru ierašanās profils.

Tāpēc, izmantojot datu apstrādes algoritmus un datu vizualizācijas rīkus tiks pētītas likumsakarības un kopējas iezīmes, kas spētu definēt kopējo pasažieru ierašanās grafiku drošības kontrolpunktam, tādējādi precīzi definējot drošības kontrolpunkta līniju atvēršanas grafiku un personāla resursu skaitu.

Datu noliktavu ir jāsadala divās daļās:

- a) visi esošie atribūti izņemot “Ier10”, ..., “Ier180” – jeb tik izveidota tabula, lai spētu analizēt un prognozēt pasažieru skaitu lidojumos.
- b) pasažieru ierašanās profilu tabula - visi atribūti izņemot „[Total] „[Capacity(pax)] „[Aizpildījums].

Dimensiju redukcija ir datu pārveidošana no daudzdimensiju telpas uz telpu ar zemu dimensiju skaitu, cenšoties saglabāt dažas jēgpilnas sākotnējo datu īpašības, ideāli pietuvinot to oriģinālajiem datiem. Darbs lielaveic lineāru datu kartēšanu zemākas dimensijas telpā tādā veidā izmēra telpās var būt sarežģīts daudzu iemeslu dēļ. Dimensiju samazināšana ir izplatīta jomās, kas nodarbojas ar lielu novērojumu skaitu un/vai lielu mainīgo skaitu, piemēram, signālu apstrāde, runas atpazīšana, neiroinformātika un bioinformātika [12].

Metodes tiek iedalītas lineārajā un nelineārajā pieejā. Pieejas var iedalīt arī funkciju atlasē un iezīmju iegūšanā. Dimensiju samazināšana var tikt izmantota trokšņu samazināšanai, datu vizualizācijai, kopu analīzei vai kā starpposmu citu analīžu veikšanai [12].

Daudzdimensiju telpā daži atribūti var būt svarīgāki par citiem, citi var būt ļoti nozīmīgi, bet citiem varētu būt nekādas informatīvas nozīmes, tādēļ datu kopai, lai no tās izveidotu

precīzāku modeli, iespējams ir nepieciešama dimensiju samazināšana. Kā noteikt to kuri atribūti ir visnozīmīgākie, un kā tos var projicēt 2D telpā [12]?

3.6.1. Galveno komponentu analīze (PCA)

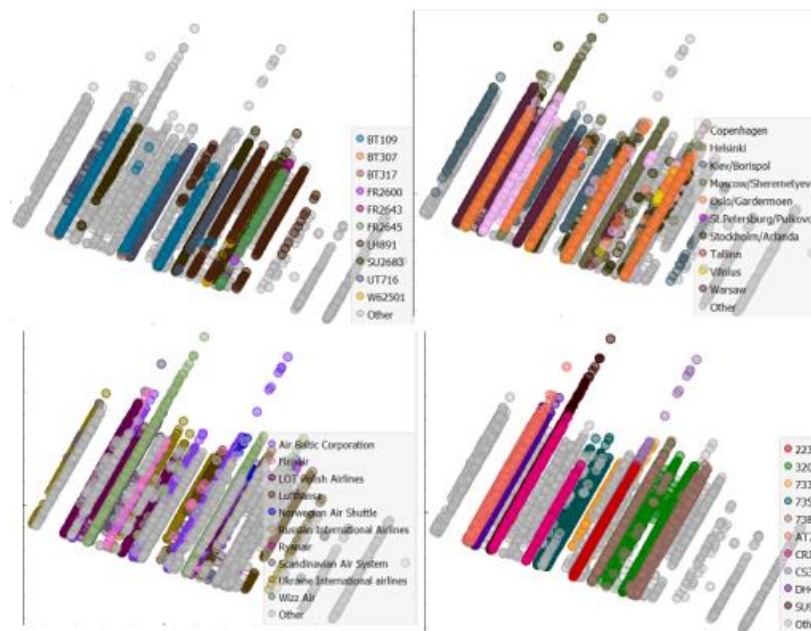
Viena no galvenajām lineārajām dimensiju samazināšanas tehnikām ir galveno komponentu analīze, kas veic lineāru datu kartēšanu telpā ar zemāku dimensiju skaitu tādā veidā, ka maksimāli palielinās datu dispersija zemu dimensiju attēlojumā. Praksē tiek konstruētas datu kovariācijas matrica un tiek aprēķināti šīs matricas īpašvektori (vektors, kuru darbinot ar noteiktu operatoru, iegūst pats sevis skalāru reizinātāju). Īpašvektorus, kas atbilst lielākajām īpašvērtībām (galvenajiem komponentiem), var izmantot, lai rekonstruētu lielu daļu sākotnējo datu dispersijas. Turklāt dažus pirmos īpašvektorus bieži var interpretēt kā plaša mēroga sistēmas fizisko uzvedību, jo tie bieži dod lielāko daļu sistēmas enerģijas, īpaši zemu dimensiju sistēmās. Tomēr tas ir jāpierāda katrā gadījumā atsevišķi, jo ne visās sistēmās ir šāda rīcība. Sākotnējā telpa (ar punkta skaita dimensiju) ir samazināta (ar datu zudumu, bet, cerams, saglabājot vissvarīgāko dispersiju), līdz telpai, kuru aptver daži īpašvektori [9].

Vispirms tiek analizēta pirmā datu (par pasažieru skaitu) kopa, kurā nepieciešams noteikt likumsakarības, kas spētu dot kādu informāciju par to vai pasažieru skaits reisos ir atkarīgs no tabulas atribūtiem. Vispirms jānoskaidro vai tie ir informatīvi. Eksperimenti ar datiem tiek lietoti izmantojot programmatūru *Orange*.

PCA algoritmam tiek padoti 12 atribūti. Kā mērķa (*target*) atribūts tiek norādīts "Total" jeb kopējais pasažieru skaits. Dispersija 99% tiek sasniegta, izmantojot 3 galvenos komponentus. Dažkārt labāki rezultāti tiek iegūti, ja dati tiek normalizēti, bet tomēr datu normalizācija šajā gadījumā labākus rezultātus nedeva, jo, izmantojot visus komponentus, netika iegūta pietiekami liela dispersija.

Apkopojot rezultātus grafikā, piemēram, PC1 → PC2, tad tika iegūts izteikts grupējums, kuru var redzēt 3.2. att.

Projicējot uz iegūtā grupējuma dažādus atribūtus tika iegūti vēl citi grupējumi, piemēram, gaisa kuģa kapacitāte, aizpildījums, gaisa kuģa tips, galamērķis, reisa numurs, kā arī kopējais pasažieru skaits. Ļoti izklaidēti atradās, lielumi – gada ceturksnis, gads, gada mēnesis, nedēļas dienas numurs un gada nedēļas numurs.



3.2. att. PCA algoritma izveidotais grupējums

Lai novērtētu, kuri atribūti ir visinformatīvākie, tika izmantot logrīks “Rank”, bet kā vērtēšanas metode – loģistiskā regresija (*logistic regression*). Tika noteikts, ka visinformatīvākie atribūti ir: “Destination”, “CompanyFlightNo”, “CompanyID”, “ac_type_id”.

Loģistiskā regresija ir piemērota regresijas analīze, kad atkarīgais mainīgais ir binārs. Tāpat kā visas regresijas analīzes, arī loģistiskā regresija ir prognozējoša analīze. Loģistisko regresiju izmanto, lai aprakstītu datus un izskaidrotu saikni starp vienu atkarīgo bināro mainīgo un vienu vai vairākiem nominālajiem, kārtas, intervāla vai koeficienta neatkarīgajiem mainīgajiem [7].

Pēc tam ar PCA algoritmu tiek pētīta otra datu kopa, kurā ir katra reisa pasažieru ierašanās profils. Arī šeit mērķis ir noteikt, no kādiem atribūtiem tiktu iegūts visprecīzākais pasažieru ierašanās profils katram reisam. Diemžēl ar PCA metodi, šeit netika iegūts kaut kāds konkrēts grupējums, kurš varētu dot kaut kādu informāciju par datu kopu.

3.6.2. Datu dimensijas redukcijas metode T-SNE

T-SNE ir nelineāra metode dimensiju samazināšanai, kas īpaši labi ir piemērota lielu dimensijas (daudzdimensiju) datu kopu vizualizēšanai.

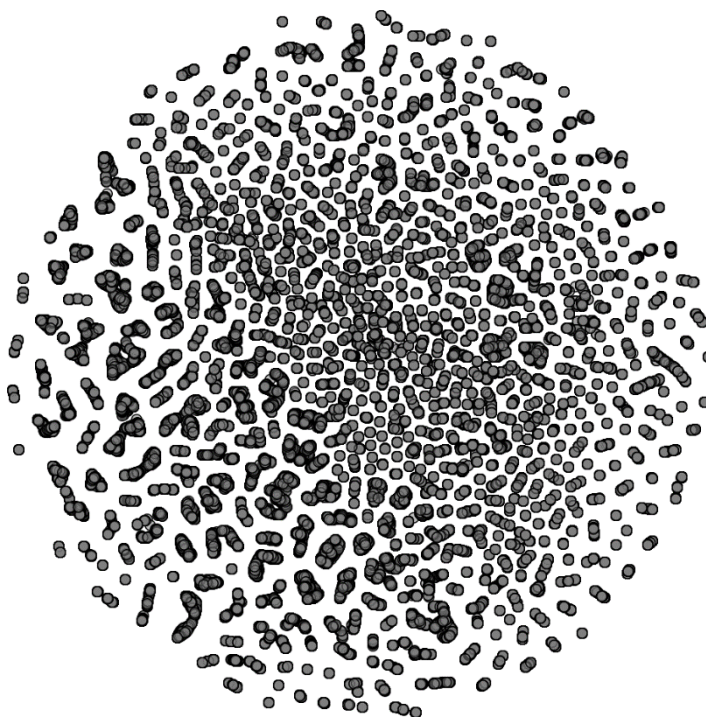
Algoritms sākas aprēķinot punktu līdzības varbūtību lielu dimensiju telpā un aprēķinot punktu līdzības varbūtību attiecīgajā mazdimensionālajā telpā. Punktu līdzību aprēķina kā nosacītu varbūtību, ka punkts “A” izvēlētos punktu “B” kā savu kaimiņu, ja kaimiņi tiktu

izvēlēti proporcionāli to varbūtības blīvumam zem Gausa (normālā) sadalījuma, kura centrā ir "A". Pēc tam tas mēģina samazināt atšķirību starp šīm nosacītajām varbūtībām vai līdzībām augstāka un zemāka izmēra telpā. Lai izmērītu nosacītās varbūtības starpības summas minimizāciju, *T-SNE* samazina kopējo datu punktu *Kullback-Leibler* (viens varbūtības sadalījums atšķiras no otrā paredzamā varbūtības sadalījuma) atšķirību summu, izmantojot gradienta nolaišanās metodi [11].

Vienkāršāk izskaidrojot *T-SNE* samazina atšķirību starp diviem sadalījumiem. Sadalījumu, kas mēra pāru līdzības ar atbilstošiem zemo izmēru punktiem iegultnī (*embedding*). Tādā veidā *T-SNE* kartē daudzdimensionālos datus mēģina atrast modeļus, identificējot novērotas kopas, pamatojoties uz datu punktu līdzību ar vairākām pazīmēm [11].

Tomēr tikai no šīs metodes nevar izdarīt secinājums, pārsvarā tas tiek izmantots kā datu vizualizācijas paņēmieni [11].

Šajā metodē līdzīgi kā ar *PCA* tiek padotas abas datu kopas. Vispirms tiek izpētīta datu kopa par pasažieru skaitu. Ar visiem komponentiem, datus normalizējot, tiek iegūts zemāk redzamais 3.3. att.



3.3. att. T-SNE algoritma pielietošana datu kopai ar pasažieru skaitu

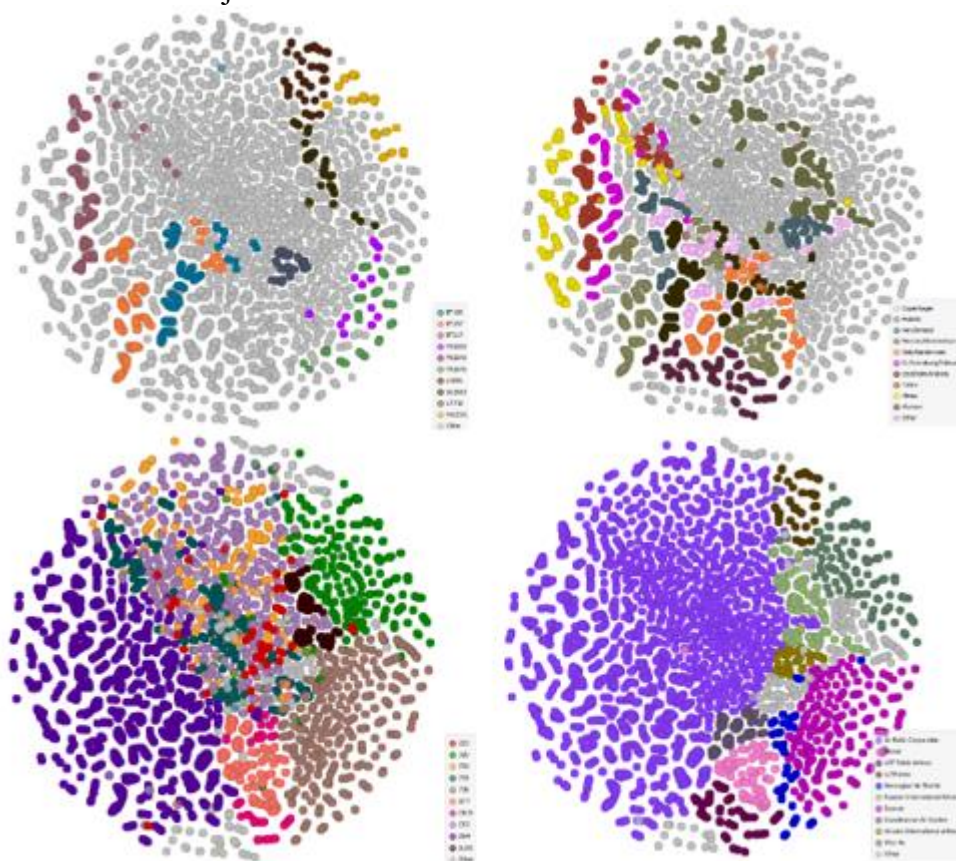
Iegūtais 3.3. att. ir liels mākonis, kurš sastāv no vairākiem maziem mākoņiem, skatoties no kā sastāv katrs mazais mākonis, tad gandrīz katram kopējās iezīmes ir:

- a) galamērķis;
- b) aviokompānija;

- c) gada mēneša numurs;
- d) gada ceturkšņa numurs.

Tas nozīmē to, ka mazais mākonis veidojas no augstāk minētajiem atribūtiem vairākiem reisiem, kuri izlido uz vienu un to pašu galamērķi un ar vienu aviokompāniju, un veido savstarpējās attiecības. Kā arī tas liek secināt, ka reisa numuram nav tik lielas nozīmes kā pašam galamērķim, no kā izriet, ka mazāk svarīgi ir tas, vai reiss notiek no rīta vai vakarā, bet toties iezīmējas, ka nozīme ir sezonalitātei.

Iekrāsojot šo mākonī attiecīgajās atribūtu krāsās iezīmējas arī citi, lielāki mākoņi, kas redzami zemāk redzamajā 3.4. att.

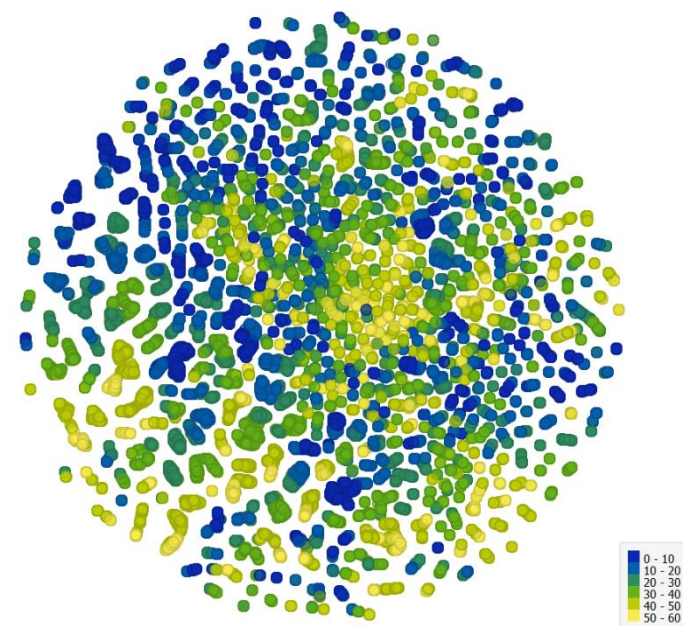


3.4. att. T-SNE algoritma vizualizācija nr.1

Reisu mākoņi pārtop par galamērķa mākoņiem, tad šie mazie mākoņi apvienojas un veido gaisa kuģa tipa mākonī, un kā virsotne izveidojas aviokompānija.

Vēlreiz izmantojot “Rank” ar vērtēšanas metodi – loģistiskā regresija (*logistic regression*), tiek izvadīts atribūtu informatīvums šādā secībā: galamērķis, reisa numurs, aviokompānija, un kā izrādās, tad pārējie atribūti pēc šīs metodes tiek uzskatīti par maznozīmīgiem.

Iekrāsojot šo mākonī nedēļas dienas krāsas, tas ir ļoti izsvaidīts, bet ejot tālāk un izmantojot gada nedēļas numurus, mākoņi nokrāsojas zemāk redzamajā 3.5. att. Samazinot komponentu skaitu attēls maina formu, bet kopumā informāciju, kuru tas sniedz, nemainās.

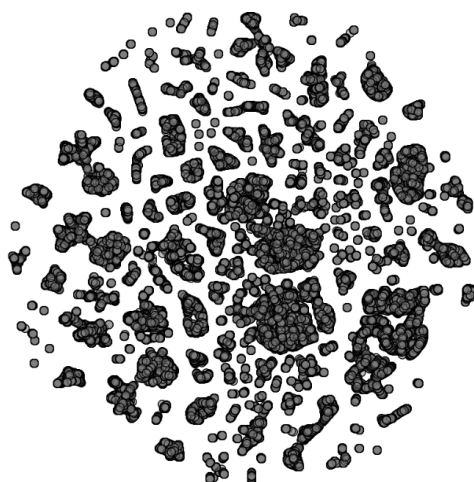


3.5. att. *T-SNE algoritma vizualizācija nr.2*

Katrs mazais mākonis ir nokrāsots vienā krāsā, kas apzīmē grupējumu ar vienādiem gada nedēļas numuriem.

Eksperiments tika veikts, arī neizmantojot datu normalizāciju, bet nekādi rezultāti iegūti netika.

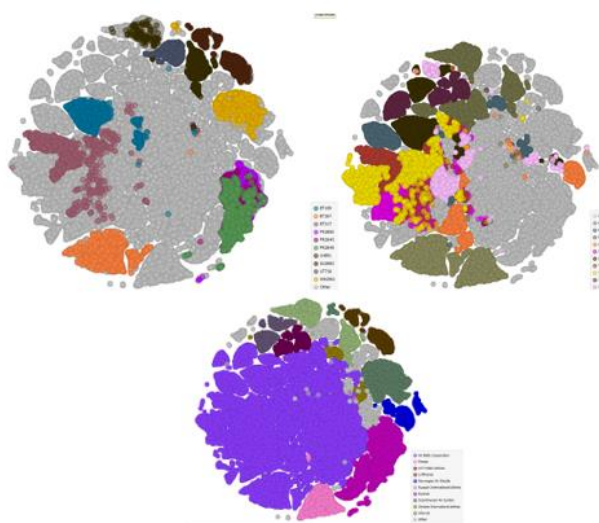
T-SNE algoritms tika pielietots arī pasažieru skaita ierašanās profilu datu kopai. Zemāk redzamajā 3.6. att. var redzēt, ka ir maza veida grupējumi, kur galvenās kopējās pazīmes ir



3.6. att. *T-SNE algoritma pielietošana datu kopai ar pasažieru ierašanās profiliem*

galamērķis, aviokompānija, gaisa kuģa tips. Reisa numuri no vienas aviokompānijas apvienojas galamērķī.

Arī logrīks “Rank” uzrāda, ka tikai augstāk minētie (galamērķis, reisa numurs, aviokompānija) atribūti ir informatīvi. Datu kopa tika testēta izmantojot arī datus, kuriem netika veikta datu normalizācija, bet līdzīgi kā iepriekš ar šāds datu apstrādes veids neuzrādīja nekādas likumsakarības vai pavedienus. Gan galamērķis, gan aviokompānijas bija “izkaisītas” pa visu attēlu. Ar samazinātu komponentu skaitu tika iegūti ļoti interesanti rezultāti, kur redzami zemāk (skat. 3.7. att.).



3.7. att. T-SNE algoritma vizualizācija nr.3

Izveidojās izteikti mākoņi, kur attēla (skat. 3.7. att.) augšā kreisajā pusē redzams, kā šie mākoņi ir iezīmēti kā reisa numurs, un ir tādi reisi, kuri cits citam ir uzvirzījušies virsū, bet attēlā pa labi, var redzēt kā abi šie reisi ir viens un tas pats galamērķis. 3.7. att. augšā pa labi var redzēt, ka viena galamērķa mākoņi ir vairāki, kas nozīmē to, ka ir kaut kāda īpašība, kas šos abus mākoņus atšķir, un tā ir aviokompānija, jo kā redzams trešajā 3.7. att. izveidojas vairāki lieli mākoņi, kuri katrs atsevišķu aviokompāniju.

Reisa izlidošanas laikam nav nozīmes, kā arī nav īsti sakarības starp dažādiem laika periodiem jeb neveidojas sezonālitate. Kā arī atribūts “gads” nav izcēlies, kas nozīmē, ka pasažieru paradumi ierodoties uz reisu ik gadu, nedēļu, vai nedēļas dienu nemainās.

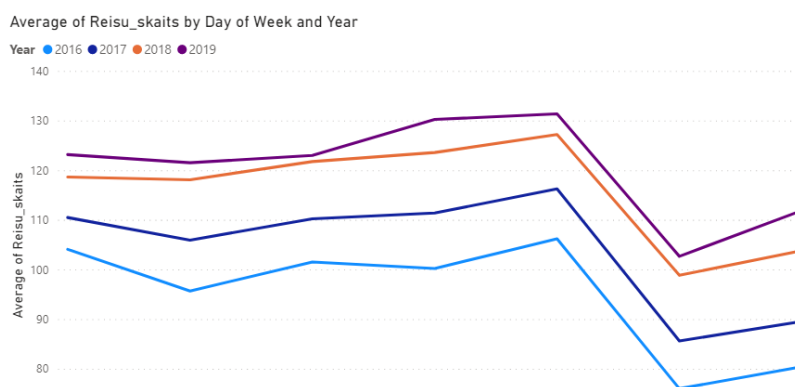
Secinājums ir tāds, kā pasažieri neatkarīgi no laika vai sezonas uz reisu ierodas noteiktā profilā atkarībā no aviokompānijas un galamērķa, taču atšķirība ir tāda, ka katrai aviokompānijai šis pasažieru ierašanās profils uz vienu un to pašu galamērķi atšķiras. Reisa izlidošanas laikam nav nozīmes, kā arī nav īsti sakarības starp dažādiem laika periodiem jeb

neveidojas sezonālitate. Kā arī atribūts gads nav izcēlies, kas nozīmē, ka pasažieru paradumi ierodoties uz reisu ik gadu, nedēļu, vai nedēļas dienu nemainās.

3.7. Datu vizualizācija POWER BI programmatūrā

Lai varētu nostiprināt iegūtās zināšanas un pārlicināties par iegūtajām likumsakarībām un secinājumiem ir nepieciešams veikt matemātiskus aprēķinus, ko pēc autora domām, visefektīvāk var izdarīt vizualizējot datu kopas grafiski. Tādēļ dati tika ielādēti *Power BI* programmatūrā ar kuras palīdzību tiks pārbaudītas iegūtās likumsakarības. Vispirms ir nepieciešams saprast, vai ir kādas tendences un sakarības starp reisu skaitu, un pasažieru skaitu drošības kontrolpunktā. Sākotnēji tiek izpētītas vai ir vispārējas korelācijas, un kā tās ietekmē viena otru.

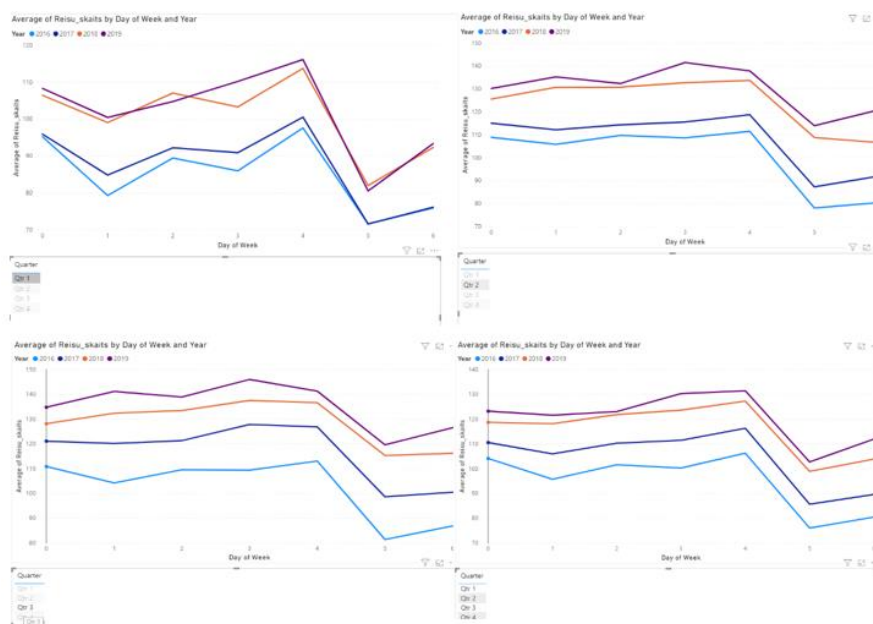
Sākotnēji tika izpētīts vidējais reisu skaits pa dienām (skat. 3.8. att.).



3.8. att. Vidēji izpildīto reisu skaits nedēļas dienā pa gadiem

Kā redzams 3.8. att., tad gadu no gada reisu skaits pa dienām ir proporcionāli ļoti līdzīgs, bet ar katru gadu tas pieaug. Attēlā (skat. 3.8. att.) redzams, ka mazākais reisu skaits ir piektdienās un sestdienās, bet lielākais ceturtdienās. No 3.8. att. var secināt, ka arī pasažieru skaitam vajadzētu būt līdzīgām tendencēm.

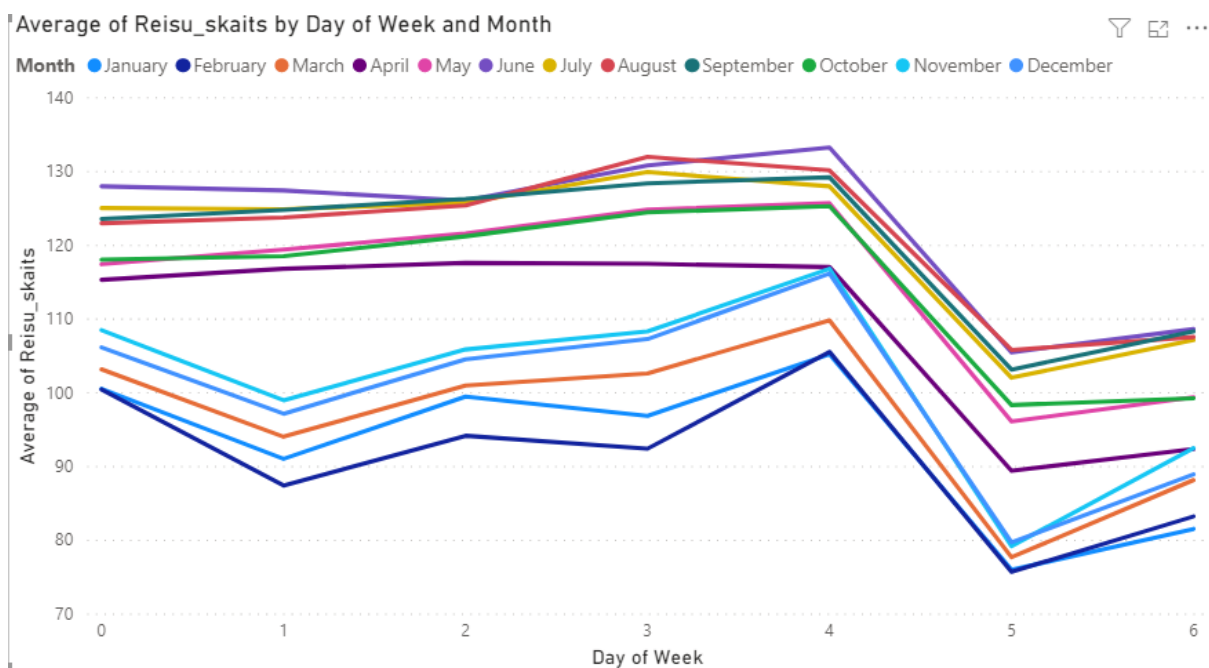
Sadalot gadu sīkāk, jeb dalot to ceturkšņos, tad zemāk redzamajā 3.9. att. izlidojošo reisu skaits dienā būtiski atšķiras tikai pirmajā ceturksnī, bet pārējos trijos tas ir ļoti līdzīgs, kā arī jau iepriekšējā attēlā bija redzams, tad arī dalot pa ceturkšņiem, gadu no gada reisu skaits, dalot pa dienām ir ļoti līdzīgs.



3.9. att. Vidēji izpildīto reisu skaits nedēļas dienā pa gadiem sadalot to ceturkšņos

Turpinot, par to kādas izpildīto reisu skaita tendences katru mēnesi atsevišķi, tad lielākās svārstības tika atklātas skatoties mēnešus, kuri atrodas pirmajā ceturksnī, tas ir – janvāris, februāris, marts, bet pārējo mēnešu atšķirības ir ļoti nelielas. Toties saglabājas tendence, ka gadu no gada šie grafiku līknes savu formu nemaina, bet toties šo reisu skaits, katru gadu, gan mēnesī, gan ceturksnī pieaug pakāpeniski.

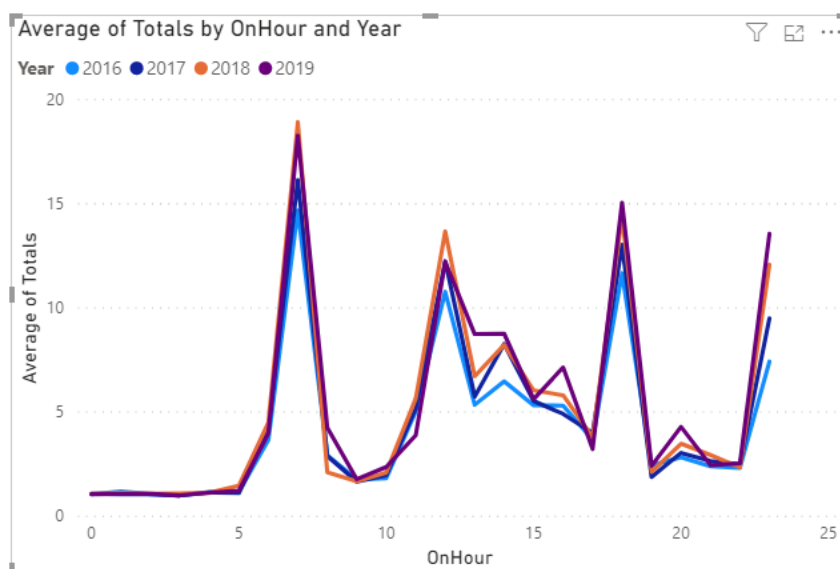
Zemāk redzamajā 3.10. att. var redzēt vidējo reisu skaitu pat dienām katrā mēnesī. Arī šeit var novērot, ka katru dienu, mēnesi no mēneša reisu skaits pa dienām ir ļoti proporcionāls. Kā redzams zemāk (skat. 3.10. att.), tad tikai aprīlis ļoti atšķiras no pārējiem mēnešiem.



3.10. att. Vidēji izpildītais reisu skaits pa mēnešiem un dienām.

Tālāk tiek aplūkots un salīdzināts, ka izskatās reisu skaits vidēji katru gadu, mēnesi, un dienu atsevišķi, un attiecīgi kopējais pasažieru ierašanās profils drošības kontrolpunktā. Tā kā augstāk redzamajos attēlos (skat. 3.8, 3.9, 3.10. att.) gadu no gada, un kā arī pa mēnešiem ņemot šīs līknes ir ļoti līdzīgas jeb paralēlās viena otrai, tāpēc arī turpmākajos pētījumos tiek rēķināti vidējie lielumi un tie tiek salīdzināti.

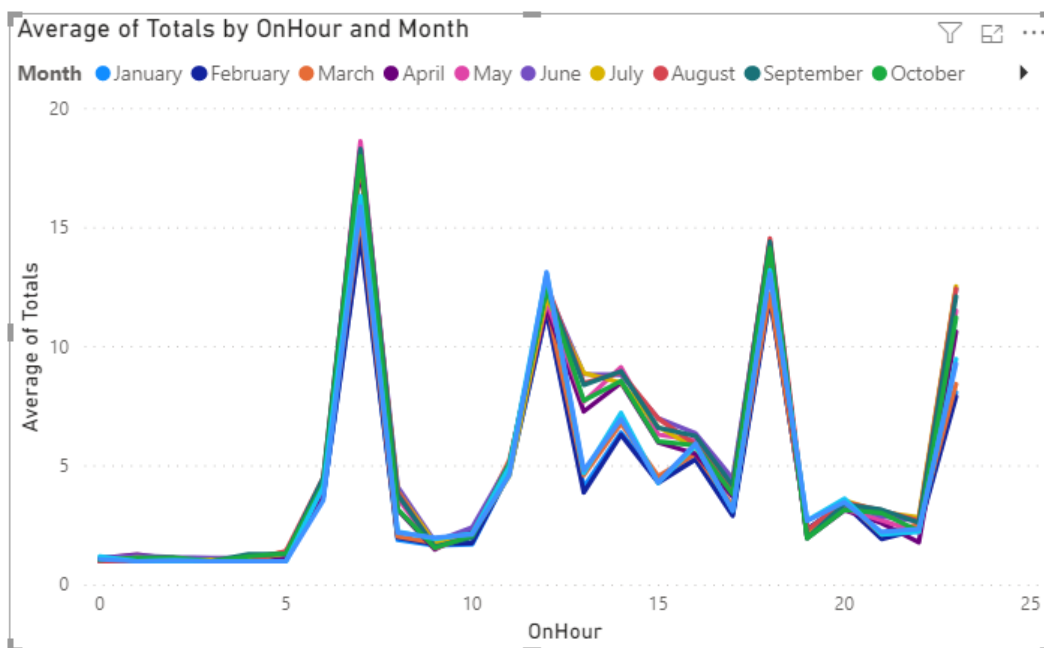
Zemāk redzamajā 3.11. att. var redzēt vidējo izpildīto reisu skaitu dienā noteiktā diennakts laikā katru gadu.



3.11. att. Vidējais izpildītais reisu skaits diennakts stundās pa gadiem

No šī attēla (skat. 3.11. att.) var secināt, ka reisu skaits vidēji dienas stundā pieaug, ko varēja jau izsecināt pirms tam. Šajā 3.11. att. ļoti interesanti ir tas, ka veidojas “pīķi”, un šie pīķi gadu no gada ir vienās un tajās pašās diennakts stundās, un skatoties uz šo attēlu, jau varētu provizoriski modelēt līkni, kura varētu attēlot kopējo diennakts pasažieru ierašanās profilu drošības kontrolpunktam.

Zemāk redzamajā attēlā redzams reisu skaits pa mēnešiem, diennakts stundā.

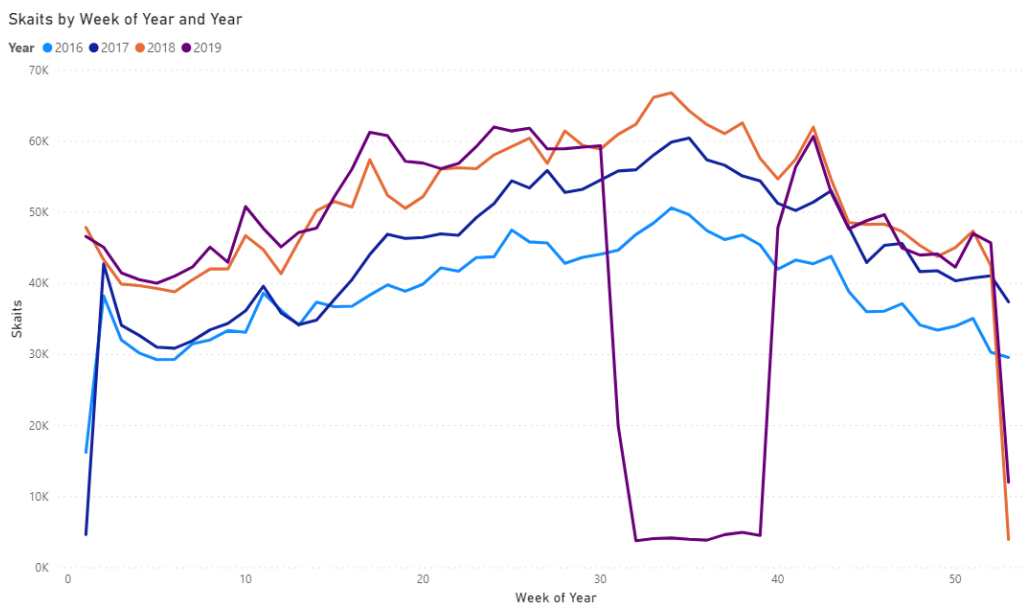


3.12. att. Vidējais izpildīto reisu daudzums diennakts laikā pa mēnešiem

Arī šajā 3.12. att. vislielākie pīķi ir redzami katrā mēnesī plkst. 7:00, 12:00, 18:00.

Pētot katru gadu atsevišķi, kā paredzams, pīķi nemainās un ir nelielas atšķirības tikai to starpposmos.

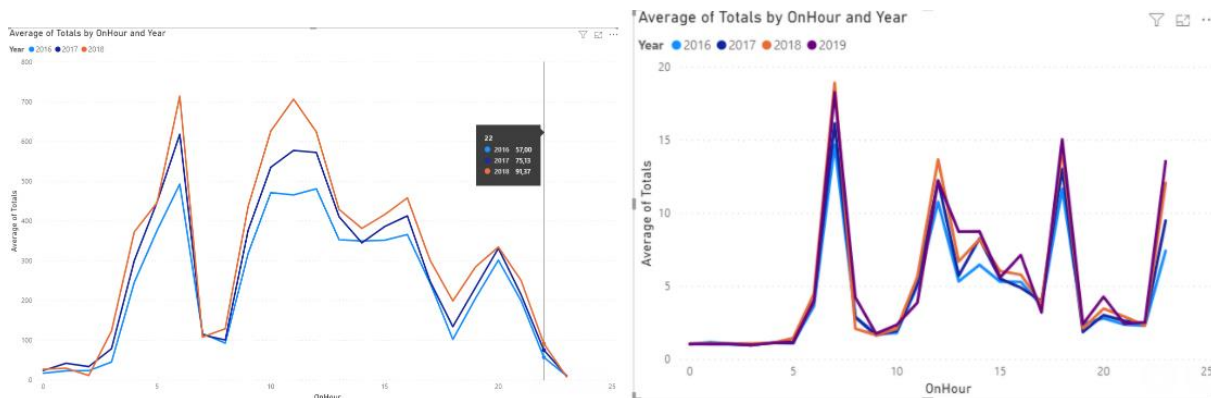
Tālāk tiek pētīts kopējais drošības kontrolpunkta pasažieru skaita profils, un tas tiek salīdzināts ar reisu skaitu. Zemāk redzamajā 3.13. att. tiek salīdzināts kopējais pasažieru skaits katru gadu katrā gada nedēļā (uz horizontālās ass attēloti gada nedēļas numuri), un tika iegūts šāds attēls.



3.13. att. Apkalpoto pasažieru skaits drošības kontrolpunktā pa gadiem

Kā redzams 3.13. att., tad tika atklāts, ka pasažieru iekāpšanas karšu skenētajos datos ir nepilnības par 2019. gadu, proti, no 30 – 41 nedēļai iztrūkst liels daudzums datu. Konsultējoties ar kolēģiem, tika noskaidrots, ka šajā laika posmā bija tehniskas problēmas ar E-vārtu darbību. Tādēļ tālākajos pētījumos šis laika posms netiek ņemts vērā, lai nebojātu kopēju datu kvalitāti. Interesantākais, ko var novērot šajos datos ir tas, ka kopējais pasažieru skaits katru gadu konkrētajā gada nedēļā arī pieaug ļoti pakāpeniski, pat varētu teikt lineāri, jo šīs līknes ir diezgan līdzīgas un arī paralēlas viena otrai.

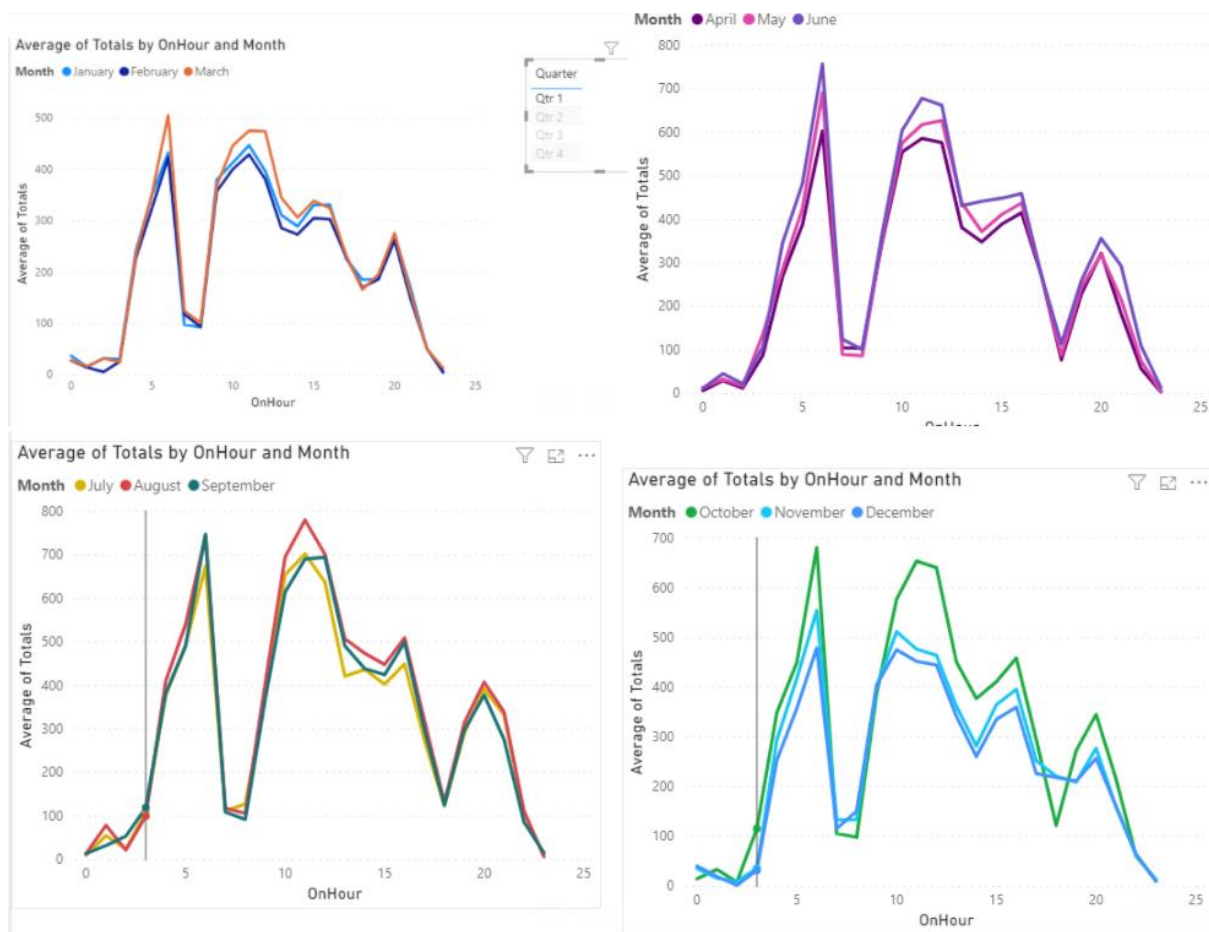
3.15. att. redzama trīs gadu (2016, 2017, 2018) vidējais pasažieru ierašanās profils drošības kontrolpunktā (kreisajā 3.15. att.) un izpildīto reisu skaits (labajā 3.15. att.) diennakts laikā. Redzamas izteiktas korelācijas starp šie abiem attēliem. Tiem ir četri izteikti pīķi, un vidēji rēķinot (ik pa stundai) pasažieri ierodas vidēji vienu stundu pirms reisa izlidošanas, proti,



3.14. att. Vidējais izpildīto reisu saraksta salīdzinājums ar vidējo pasažieru skaitu diennaktī

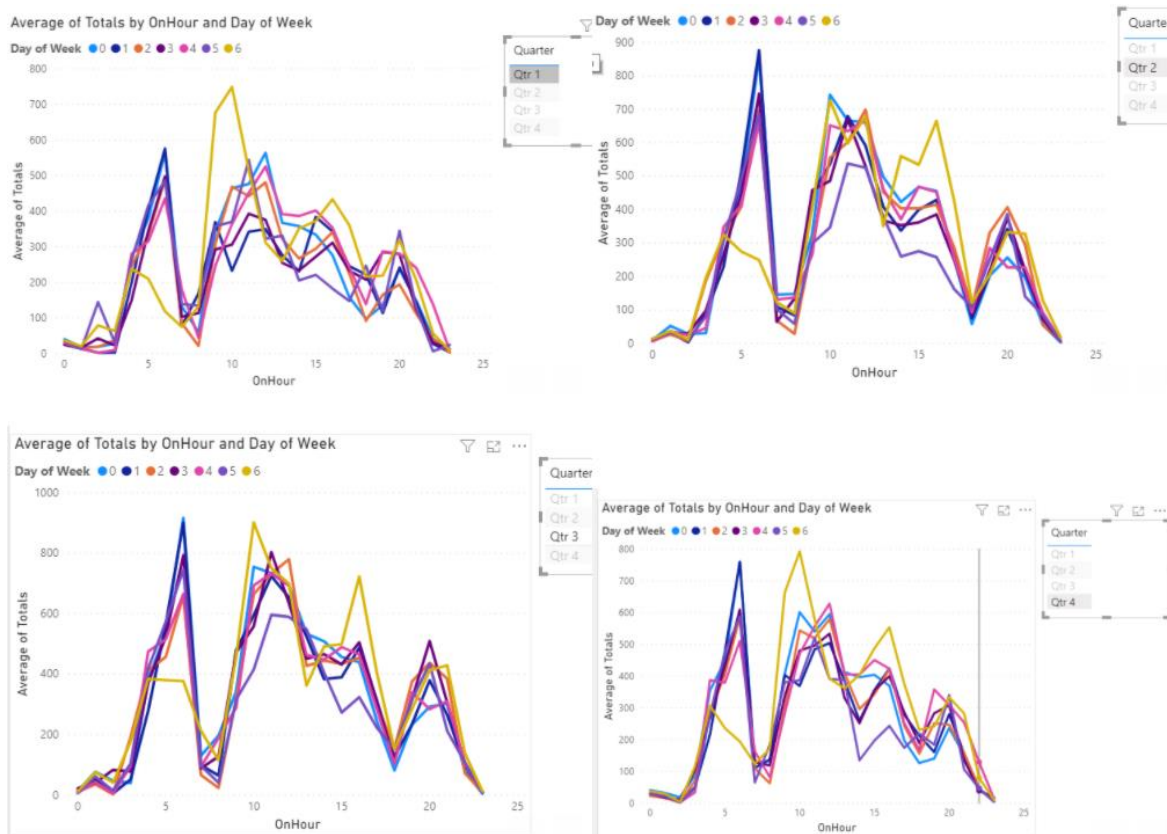
pasažieru pīķis ir vienu stundu pirms reisu izlidošanas pīķa, kas nosaka to, ka vispārēji pasažieris ierodas drošības kontrolpunktā vienu stundu pirms reisa.

Izskatot grafikus pa ceturkšņiem (skat. 3.17. att.) un konkrētāk katru gadu pa mēnešiem, un dienām, tad šie četri galvenie pīķi ir nemainīgi. Protams, pasažieru ierašanās grafiks mainās ik mēnesi, jo katru mēnesi mainās arī izlidojošo reisu skaits, bet jo lielāks laika periodos tiek ņemts, tā grafiki ir līdzīgāki viens otram.



3.17. att. Vidējais apkalpotais pasažieru skaits drošības kontrolpunktā pa ceturkšņiem diennakts laikā

Lielākās atšķirības ir pētīt, katru dienu atsevišķi, tas varētu būt saistīts ar to, ka katrā dienā izlidojošo reisu skaits ir atšķirīgs, tādējādi tas ietekmē arī pasažieru skaitu, bet gadu no gada lielas atšķirības nav novērojamas.



3.18.att. Vidējais pasažieru skaita salīdzinājums drošības kontrolpunktā pa nedēļas dienas numuriem diennakts laikā

Augstāk redzamajā 3.18. att. var redzēt, otrais un trešais ceturksnis (3.18. att. augšējais labais un apakšējais kreisais attēls) ierašanās laikos ņemot pa konkrētu dienu ir ļoti līdzīgi, kā arī pirmais (3.18. att. augšējais kreisais attēls) un ceturtais (3.18. att. apakšējais labais attēls). Summējot šos abus grafikus, piemēram, pirmo un ceturto, tad iegūst vasaras un ziemas laiku sezonālītāti drošības kontrolpunktā, kas norāda uz to, ka šajās sezonās pasažieru ierašanās paradumi ir nedaudz atšķirīgi.

3.7.1. Pasažieru skaita prognoze.

Iepriekš analizējot izveidotos grafikus tika secināts, ka reisu kopējās pazīmes starp kurām būtu ievērojamas sakarības ir:

- a) galamērķis;
- b) aviokompānija;
- c) gada mēneša numurs;
- d) gada ceturkšņa numurs.

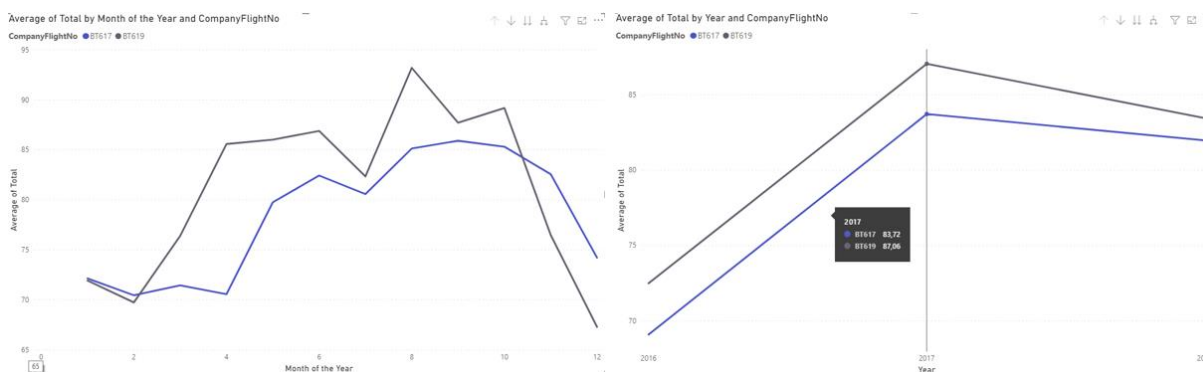
Izmantojot šos atribūtus, pasažieru skaitam vajadzētu būt konstanti lielam, piemēram, lidojot uz kādu noteiktu galamērķi, ar vienu aviokompāniju, noteiktā mēnesī no kā seko arī konkrētais ceturksnis. Tika izpētīti vairāki galamērķi un to reisi, lai varētu apstiprināt iepriekš minētos secinājumus, bet iegūtie grafiki, kā, piemēram, uz galamērķi “Amsterdama”, ka ņemot kā piemēru otrā mēneša jeb februāra reisu trešdienās, ir redzams (skat. 3.19. att.), ka pasažieru skaits ir ļoti svārstīgs un nav stacionārs, un tas nekādi nekorelē ar izmantotajiem atribūtiem. Apvienojot to ar pārējām nedēļas dienām iegūtie rezultāti ir tādi paši – ļoti svārstīgi.



3.19. att. Reisu uz galamērķi “Amsterdama” pasažieru skaita tendences

Šīs līknes sāk stabilizēties tikai ņemot lielākus periodus, piemēram, veselus mēnešus, ceturksņus vai gadus. Kā redzams 3.20. att., šī paša galamērķa pasažieru skaita līknes pa mēnešiem un gadiem.

Tika izpētīti vēl citi galamērķi, un tajos bija šādas pašas īpašības un tendences. Tāpēc, dienu no dienas ir ļoti grūti noteikt kādas prognozes, tuvākajiem lidojumiem. Tāpēc, tas vedina



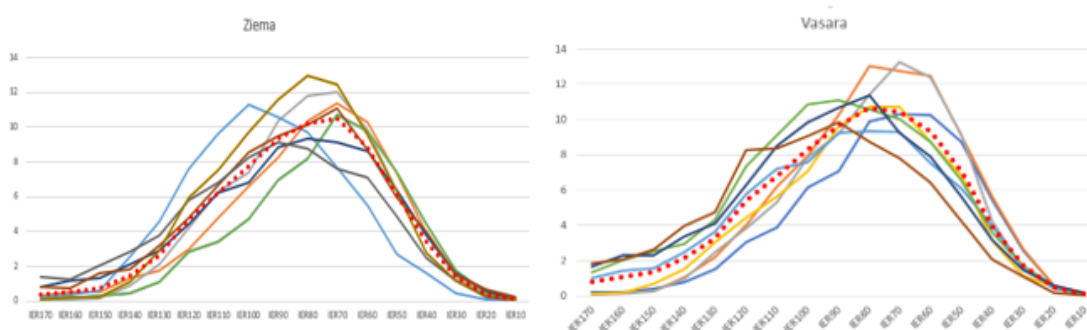
3.20. att. Reisu uz galamērķi “Amsterdama” pasažieru skaita tendences pa mēnešiem un gadiem

uz domu, ka veidojot prognozes ir jāizmanto 4.1 punktā aprakstītā naivā sezonālā vidējā metode, kur pasažieru skaita prognoze būtu iepriekšējā gada katra mēneša vidējā vērtība, bet lai tomēr iespējami palielinātu precizitāti, pasažieru skaits tiek prognozēts katram reisam atsevišķi, nevis galamērķim kopumā, jo ir pietiekami daudz gadījumu, kad sakarība ar reisiem uz vienu un to pašu galamērķi neizpildās.

3.7.2. Pasažieru ierašanās profilu izveidošana.

Veicot pasažieru ierašanas profilu datu analīzi (3.6.2. punktā) tika iegūts, ka ierašanās profiliem vajadzētu būt līdzīgiem neatkarīgi no reisa, bet gan no galamērķa un aviokompānijas ar kādu tiek izpildīts šis reiss. Apskatot dažādus pasažieru ierašanās profilus *Power BI* programmatūrā vienam galamērķim ar vairāk nekā vienu reisa skaitu, un no vienas aviokompānijas.

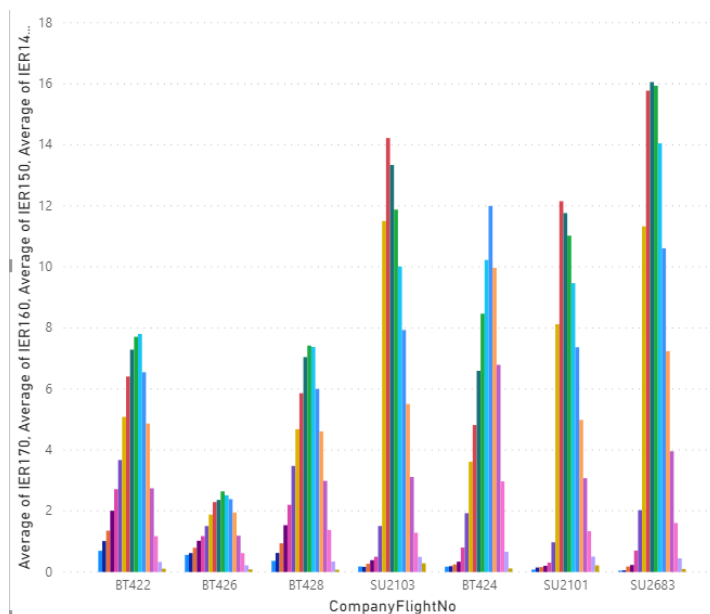
Iepriekš tika izpētīti dažādu reisu pasažieru ierašanās profili, kuri veic lidojumus uz vienu un to pašu galamērķi un aviokompānijas. Attēlā (skat. 3.21. att.) parādīts jau iepriekš aplūkotā galamērķa Amsterdamā pasažieru ierašanās grafiki reisiem *BT617* un *BT619* ziemas un vasaras sezonām. 3.21. att. redzamā sarkanā raustītā līnija ir abu divu sezonu kopējais vidējais pasažieru



3.21. att. Pasažieru ierašanās profilu salīdzinājums vasaras un ziemas periodiem

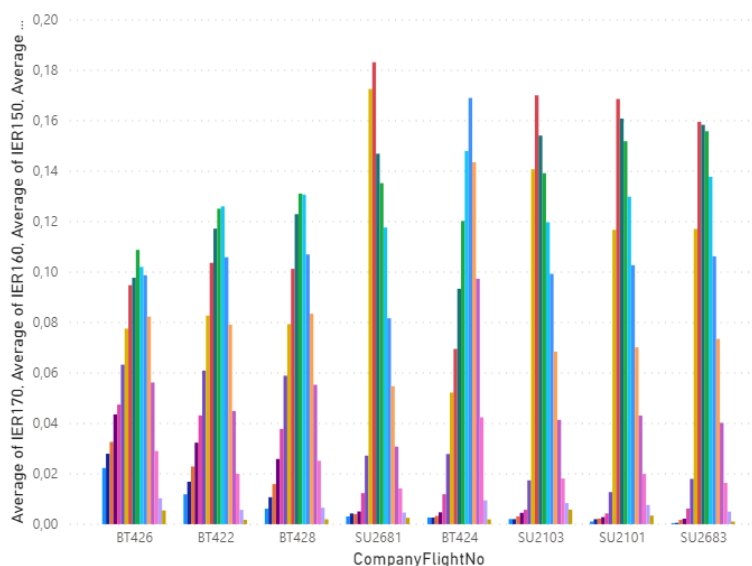
ierašanās grafiks galamērķim Amsterdamā, bet pārējās līnijas ir šo sezonu mēneši abiem šiem reisiem *BT617* un *BT619*. Abu divu sezonu vidējie ierašanās grafiki ir ļoti līdzīgi viens otram, kā arī atsevišķu mēnešu līknes no vidējām iet paralēli viena otrai, neskatoties uz to, ka reiss *BT617* izlido no rīta, bet *BT619* vakarā.

3.23. att. redzamais galamērķis ir Moscow/Sheremetyevo, kur ir parādīti reisi no divām aviokompānijām (SU-Russian International Airlines, BT - Air Baltic) BT reisiem šis grafiks aug pakāpeniski, bet SU ļoti strauji.



3.22. att. Pasažieru ierašanās profili galamērķim Moscow/Sheremetyevo

Stabiņu augstumi ir dažādi, un tas norāda uz to, ka pasažieru skaits ir dažāds, tāpēc pasažieru ierašanās profili tiek pārveidoti procentuālā formā (piemērām 25% pasažieru ierodas 90 minūtes pirms reisa), lai noteiktu kopējās pasažieru ierašanās drošības kontrolpunktā tendences neatkarīgi no pasažieru skaita. Kā redzams 3.22. att. šie grafiki uzreiz izlīdzinās un kļūst ļoti līdzīgi. Tāpēc šī būtu universāla pieeja, kā izveidot pasažieru ierašanās profilus modelējot tos procentuālā formā, tādējādi tos būtu ļoti ērti attiecināt uz konkrētā galamērķa reisu, neatkarīgi no tā cik pasažieru varētu ierasties uz šo reisu.



3.23. att. Pasažieru ierašanās profili galamērķim Moscow/Sheremetyevo procentuālā formā

4. PASAŽIERU SKAITA PROGNOZĒŠANAS METODOLOĢIJAS.

Būtu daudz vienkāršāk, ja aviokompānijas dalītos datos ar lidostām par jau rezervētajām iekāpšanas kartēm, bet diemžēl tā nenotiek un lidostai pašai ir jāprognozē, cik daudz pasažieru tiem būs jāapkalpo.

Pasažieru skaita datu kopai tiek pielietota laika rindu analīze, kura aplūko vērtību kolekciju, kas novērota secīgi laika gaitā [13].

Regresijas analīze var palīdzēt saprast attiecības un palīdzēt prognozēt nepārtrauktus mainīgos, pamatojoties uz citiem datu kopas mainīgajiem. Šis paņēmieni ir paredzēts, lai identificētu nozīmīgas sakarības starp datu mainīgajiem, īpaši aplūkojot saistības starp atkarīgo mainīgo un citiem neatkarīgiem faktoriem, kas to var vai nevar ietekmēt [13].

Atbilstošās prognozēšanas metodes lielā mērā ir atkarīgas no pieejamiem datiem [4].

Ja nav pieejami dati, vai ja pieejamie dati nav saistīti ar prognozēm, jāizmanto kvalitatīvas prognozēšanas metodes. Šīs metodes nav tikai minējumi, bet gan labi izstrādātas un strukturētas pieejas, lai iegūtu labas prognozes, neizmantojot vēsturiskos datus [4].

Kvantitatīvo prognozēšanu var piemērot, ja ir izpildīti divi nosacījumi [4]:

- a) ir pieejama skaitliska informācija par pagātni;
- b) ir pamatoti pieņemt, ka daži pagātnes modeļu aspekti turpināsies arī nākotnē;

Ir plašs kvantitatīvās prognozēšanas metožu klāsts, ko bieži izstrādā konkrētās disciplīnās īpašiem mērķiem. Katrai metodei ir savas īpašības un precizitāte, kas jāņem vērā, izvēloties konkrētu metodi [4].

Lielākajā daļā kvantitatīvo prognožu problēmu tiek izmantoti vai nu laika rindu dati (savākti ar regulāriem laika intervāliem), vai šķērsriezuma dati (apkopoti vienā laika posmā) [4].

4.1. Dažas vienkāršākās prognozēšanas metodes

Dažas prognozēšanas metodes ir vienkāršas un pārsteidzoši efektīvas. Zemāk tiek parādītas četras vienkāršas prognozēšanas metodes [4].

Vidējā (average) metode. Šeit visu nākotnes vērtību prognozes ir vienādas ar vēsturisko datu vidējo vērtību. Ja vēsturiskos datus apzīmē ar $y_1 + \dots + y_T$, tad prognozes var rakstīt kā (skat. formulu $\hat{Y}_{T+h|T} = \bar{y} = (y_1 + \dots + y_T)/T$) (4.1) [4]:

$$\hat{Y}_{T+h|T} = \bar{y} = (y_1 + \dots + y_T)/T \quad (4.1)$$

Apzīmējums $\hat{Y}_{T+h|T}$ ir īss rādītājs Y_{T+h} pamatojoties uz datiem $y_1 + \dots + y_T$ [4].

Naivā metode. Naivām prognozēm tiek iestatīts, ka visas prognozes ir pēdējo novērojumu vērtības. Tas ir (skat. formulu $\hat{Y}_{T+h|T} = Y_T$ (4.2) [4],

$$\hat{Y}_{T+h|T} = Y_T \quad (4.2)$$

Šī metode darbojas izcili labi daudzās ekonomikas un finanšu laika rindās [4].

Sezonālā naivā metode. Līdzīga metode ir noderīga ļoti sezonāliem datiem. Šajā gadījumā katru prognozi iestata kā vienādu ar pēdējo novēroto vērtību tajā pašā gada sezonā (piemēram, iepriekšējā gada tajā pašā mēnesī) (skat. formulu $\hat{Y}_{T+h|T} = Y_{T+h-m(k+1)}$ (4.3)). Formāli laika prognoze T+h ir [4]:

$$\hat{Y}_{T+h|T} = Y_{T+h-m(k+1)} \quad (4.3)$$

Kur m ir sezonas periods un k ir veselā skaitļa daļa no (h-1)/m (t.i. pabeigto gadu skaits prognozes periodā pirms laika T+h. Piemēram, izmantojot mēneša datus, visu turpmāko februāra vērtību prognoze ir vienāda ar pēdējo novēroto februāra vērtību. Izmantojot ceturkšņa datus, visu turpmāko Q₂ vērtību prognoze ir vienāda ar pēdējo novēroto Q₂ vērtību (kur Q₂ nozīmē otro ceturksni). Līdzīgi noteikumi attiecas arī uz citiem mēnešiem un ceturkšņiem, kā arī uz citiem sezonas periodiem [4].

4.2. Laika rindu modeļa izveide un analīze

Prognozējot gaisa satiksmes aktivitāti, vēsturisko datu (laika rindu) izpēte un spēja noteikt satiksmes attīstības tendenci. Īstermiņa, vidēja termiņa vai ilgtermiņa prognozēšanas kontekstā satiksmes tendence atspoguļo satiksmes attīstību daudzu gadu garumā, izolējot no īstermiņa satiksmes līmeņa svārstībām, iegūstot vidēja termiņa vai ilgtermiņa prognozi. Ekstrapolējot no satiksmes tendences, prognozētājs pieņem, ka faktori, kas noteica satiksmes vēsturisko attīstību, turpinās darboties tāpat kā agrāk, izņemot gadījumus, kad to ietekme var mainīties pakāpeniski. Tendences analīzes izmantošanā prognozēšanā lielā mērā ir atkarīga no iepriekšējo notikumu stabilitātes un prognozētāja pārliecības, ka pastāvīgu tendenču pieņēmums ir piemērots konkrētajai darbības videi [5].

Tā kā šajā darbā ir nepieciešams plānot gan pasažieru plūsmu katru dienu, tad ir jāpielieto īstermiņa prognozes.

Vienkāršākajā veidā tendenču projekcijas analīze nav nekas sarežģītāks par satiksmes (*traffic*) datu sērijas uzzīmēšanu grafikā. Prognozējamais satiksmes mainīgais (*the dependent variable*) tiek attēlots uz vertikālās ass, bet laiks (paskaidrojošais vai neatkarīgais mainīgais) –

uz horizontālās ass. Kad katrs laika rindas punkts ir iezīmēts, tad ar vienmērīgu līkni, kas šķiet, tuvojas visiem punktiem var uzzīmēt ar brīvu roku, vai arī var zīmēt līniju caur datiem [5].

Viss, kas laika gaitā tiek novērots secīgi ir laika rinda. Prognozējot laika rindu datus, mērķis ir novērtēt, kā novērojumu secība turpināsies arī nākotnē.

Šajā darbā ir nepieciešams izstrādāt modeli, kā prognozēt iknedēļas gaisa pasažieru satiksmi lidostas piedāvātajos maršrutos, ņemot vērā to, ka uz vienu un to pašu galamērķi ir iespējams aizlidot ar dažādām aviokompānijām. Gaisa pasažieru skaitu ietekmē skolēnu brīvdienas, galvenie sporta notikumi, reklāmas kampaņas u.c.

4.3. Holt-Winters` s sezonālā izlīdzināšanas metode

Charles Holt (1957) un *Peter Winters* (1960) izstrādāja paplašinātu *Holt* metode ar sezonālātes uztveršanu. *Holt-Winters` s* sezonālā metode ietver prognozes vienādojumu un trīs izlīdzināšanas vienādojumus [7]:

- a) Līmenim l_t ;
- b) Tendencei b_t ;
- c) Sezonālātes komponente s_t ;

Ar atbilstošiem izlīdzināšanas parametriem α , β^* un γ . Tiek izmantots m , lai apzīmētu sezonālātes biežumu, tas ir, sezonālātes biežumu gadā. Piemēram, ceturkšņa datiem $m=4$, un mēneša datiem $m=12$ [7].

Šai metodei ir divas variācijas, kas atšķiras pēc sezonas komponenta rakstura. “*Additive*” metode ir vēlama, ka sezonālās variācijas ir aptuveni nemainīgas visā sērijā, savukārt “*multiplicative*” metode ir ieteicama, ja sezonālās variācijas mainās proporcionāli sērijas līmenim. Izmantojot “*additive*” metodi, sezonas komponents tiek izteikts absolūtos skaitļos novēroto sēriju skalā, un līmeņa vienādojumā sērija tiek sezonāli koriģēta, atņemot sezonas komponentu. Katra gada laikā sezonas sastāvdaļa būs aptuveni nulle. Izmantojot “*multiplicative*” metodi sezonas komponents tiek izteikts relatīvā izteiksmē (procentos), un sērijas tiek sezonāli koriģētas, dalot tās ar sezonas sastāvdaļu, katra gada laikā sezonas sastāvdaļa tiks summēta aptuveni $m=12$ [7].

Ļoti liels ieguvums šai metodei ir tas, ka izmantojot *Holt-Winters* konfigurāciju, tā var izstrādāt arī divkāršās un vienreizējās eksponenciālās izlīdzināšanas modeļus.

“*Additive*” metodes komponenta forma ir [4]:

$$\hat{Y}_{t+h|t} = l_t + hb_t + s_{t+h-m(k+1)} \quad (4.4)$$

$$l_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(l_{t-1} + b_{t-1}) \quad (4.5)$$

$$b_t = \beta^*(l_t - l_{t-1}) + (1 - \beta^*)b_{t-1} \quad (4.6)$$

$$s_t = \gamma(y_t - l_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} \quad (4.7)$$

Kur k ir veselā skaitļa daļa $(h-1)/m$, kas nodrošina, ka prognozēšanai izmantoto sezonālo indeksu aplēses nāk no izlases pēdējā gada. Līmeņa vienādojums pierāda vidējo svērto vērtību starp sezonāli izlīdzināto novērojumu $(y_t - s_{t-m})$ un nesezonas prognoze $(l_t - l_{t-1})$ laikam t . Tendences vienādojums ir identisks *Holta* lineārajai metodei. Sezonas vienādojums parāda vidējo svērto vērtību starp pašreizējo sezonas indeksu, $(y_t - l_{t-1} - b_{t-1})$ un sezonas indekss no tās pašas sezonas pēdējā gadā (t.i. “ m ” pirms laika periodiem) [4].

Sezonas komponenta vienādojumu bieži izsaka kā [4]:

$$s_t = \gamma^*(y_t - l_t) + (1 - \gamma^*)s_{t-m} \quad (4.8)$$

Ja tiek aizstāts l_t no iepriekš minētā komponenta formas līmeņa izlīdzināšanas vienādojuma tiek iegūts [4]:

$$s_t = \gamma^*(1 - \alpha)(y_t - l_{t-1} - b_{t-1}) + [1 - \gamma^*(1 - \alpha)]s_{t-m} \quad (4.9)$$

Kas ir identisks sezonas komponenta izlīdzināšanas vienādojumam, kuru tiek norādīts $\gamma^*(1 - \alpha)$. Parastais parametru ierobežojums ir $0 \leq \gamma^* \leq 1$, kas nozīmē $0 \leq \gamma \leq 1 - \alpha$ [4].

Šī eksponenciālā izlīdzināšana ir laika rindu prognozēšanas metode vienveidīgiem datiem, ko var paplašināt, lai atbalstītu datus ar sistemātisku tendenci vai sezonālātes komponenti. Iepriekšējie novērojumi tiek vērtēti ar ģeometriski samazinošu attiecību. Prognozes, kas iegūtas, izmantojot eksponenciālas izlīdzināšanas metodes, ir iepriekšējo novērojumu svērtie vidējie rādītāji. Svāri eksponenciāli samazinās novērojumiem novecojot - jo jaunāks novērojums, jo lielāks attiecīgais svārs [4].

4.4. Nepieciešamo bibliotēku pievienošana un datu sagatavošana

Izveidotais prognozēšanas modelis un salīdzinošā analīze ir realizēta *Python 3.9.1* programmatūras vidē.

Tiek sagatavoti laika rindu dati par katru no reisiem “.csv” formātā, izveidotais modelis tiek pielietots katram reisam atsevišķi. Zemāk aprakstītais modelis ir viena reisa apstrādes piemērs.

Sākotnēji tiek importētas nepieciešamās bibliotēkas, kuras ir nepieciešamas datu apstrādei, to analīzei un prognozēšanas modeļa izveidei. Ar komandu `“import pandas as pd”` palīdzību tiek izsaukts pandas modulis. Šis modulis ļauj izmantot divdimensionālu datu struktūru, kā arī ļauj šajā struktūrā ievietot datus. Vēl ir nepieciešamas tādas bibliotēkas kā `“datetime”` – nodrošina klases manipulēšanai ar datumiem un laikiem, `“numpy”` – galvenā bibliotēka zinātniskai skaitļošanai, tā nodrošina augstas veiktspējas daudzdimensionālu masīvu objektus un rīkus darbam ar šiem masīviem, `“sklearn.metrics”` - īsteno funkcijas, kas noteiktiem mērķiem novērtē prognozēšanas kļūdas, `“mean_squared_error”` – novērtētāja vidējā kvadrāta kļūda (*MSE*) mēra kļūdu kvadrātu vidējo līmeni, tas ir, vidējo kvadrātu starpību starp novērtētajām vērtībām un patieso vērtību, `“math”` – matemātiskas funkcijas, `“sqrt”` – iebūvēta funkcija *Python* programmēšanas valodā, kura atgriež jebkura skaitļa kvadrātsakni, `“matplotlib.pyplot”` – funkciju kopums, kas liek *matplotlib* darboties kā *MATLAB* (augstas veiktspējas valoda tehniskai skaitļošanai, tā integrē skaitļošanu, vizualizāciju un programmēšanu viegli lietojamā vidē), `“sys”` – nodrošina piekļuvi mainīgajiem, `“warnings”` – brīdinājuma ziņojumi, `“itertools”` – šis modulis īsteno vairākus iteratora veidošanas blokus, kuri būvēti no *APL*, *Haskell* un *SML* konstrukcijas. Katrs no tiem ir pārveidots *Python* piemērotā formā, `“statsmodels.tsa”` – satur modeļu klases un funkcijas, kas noder laika rindu analīzei. Pamata modeļi ietver vienfaktora (*univariate*) autoregresīvos modeļus (*AR*), vektoru autoregresīvos modeļus (*ARMA*). Nelineāri modeļi ietver Markova komutācijas dinamisko regresiju un autoregresiju. Tas ietver arī aprakstošu statistiku par laika rindām, piemēram, autokorelāciju, daļējas autokorelācijas funkciju periodogramu, kā arī atbilstošās *ARMA* vai saistīto procesu teorētiskās īpašības. Tas ietver arī metodes darbam ar autoregresīviem un slīdošiem vidējiem lag-polinomiem (*lag-polynomials*). Turklāt ir pieejami saistīti statistikas testi un dažas noderīgas palīgfunkcijas. `“statsmodel.tsa.arima_model”` – laikarindas regresija ar *ARIMA* troksni, iztrūkstošām vērtībām un izlecošām vērtībām (*Autoregressive Integrated Moving Average ARIMA(p,d,q) Model*).

Dati tiek ievietoti datu struktūrā ar komandas `“air_passanger_data = pd.read_csv(csv_faila_nosaukums)”` un izvadīti ekrānā `“air_passanger_data.head()”`. Kad dati ir ievietoti, tos ir nepieciešams apstrādāt, atribūtu `“Day”` ir nepieciešams pārveidot no datu tipa *String* uz *DateTime* un piešķirt indeksu pēc dienas. Pārveidošanu var redzēt zemāk (skat. 4.1. att.).

```
#Pārveido Day no String uz DateTime formātu
air_passanger_data['Day'] = pd.to_datetime(air_passanger_data['Day'],infer_datetime_format=True)
air_passanger_data = air_passanger_data.set_index(['Day'])
```

4.1. att. Atribūta “Day” pārvēršana DateTime formātā

4.5. Stacionaritāte (*Stationarity*)

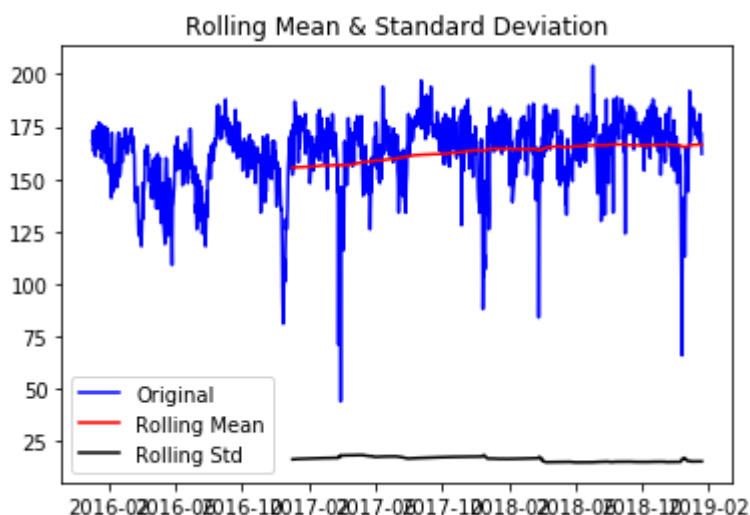
Stacionaritāte ir nemainīgo statistisko īpašību (vidējo, dispersijas, autokorelācijas u.c.) īpašība. Ja laika rindas vidējais rādītājs laika gaitā palielinās, tas nav stacionārs [2].

Pārveidojumi, ko izmanto datu stacionēšanai [2]:

- a) Datu indeksēšana;
- b) Neindeksēti dati;
- c) Atšķiršana – sezonālos vai cikliskos modeļus var novērst atņemot periodiskās vērtības. Ja dati ir 12 mēnešu sezonāli, atņemot sēriju ar 12 kavējumu starpības sēriju, tiks iegūta “*plakanākā*” sērija;
- d) Reģistrēšana – var palīdzēt linearizēt sēriju ar eksponenciālu tendenci.

Stacionaritātes pārbaudi var veikt vizuāli, aprēķinot slīdošās vidējās vērtības (*rolling mean*) un slīdošo vidējo standartnovirzi (*rolling std*) [2].

Ar nākošo koda fragmentu tiek pārbaudīts vai laika rindu dati ir stacionāri (*stationary*) vai nē. Tiek novērtēta pasažieru skaita slīdošā vidēja vērtība (*rolling mean*) ar komandas palīdzību “*rolmean = timeseries.rolling(window=12).mean()*” un slīdošā standartnovirze (*rolling std*) “*rolstd = timeseries.rolling(window=12).std()*”. Iegūtie rezultāti redzami zemāk iegūtajā grafikā (skat. 4.2. att.). Skaitlis 12 nozīmē to aprēķini veikti ik mēnesi.



4.2. att. Slīdošās vidējās vērtība un standartnovirzes rezultāti

Slīdošās vērtības, gan slīdošā standartnovirze šajā gadījumā ir mērenas un konstantas, kas nozīmē to, ka kopējais pasažieru skaits šim reisam laika gaitā pa mēnešiem ir nemainīgs. Šī analīze ir veikta ar sezonalitāti – 12 mēnešiem. Veicot eksperimentus ar ikdienišķiem datiem slīdošās vērtības vairs nav konstantas. Tas nozīmē to, ka konkrēta noturības šim reisam veidojas ilgākā laika periodā, bet ikdienišķi pasažieru skaits ir svārstīgs.

4.6. Dickey-Fuller tests

Šo testu izmanto, lai novērtētu, vai laika rinda ir stacionāra. Šī pārbaude dod rezultātu, kuru sauc par “testa statistiku”, uz ko pamatojoties, var teikt vai laika rindas ir stacionāras vai nē [2].

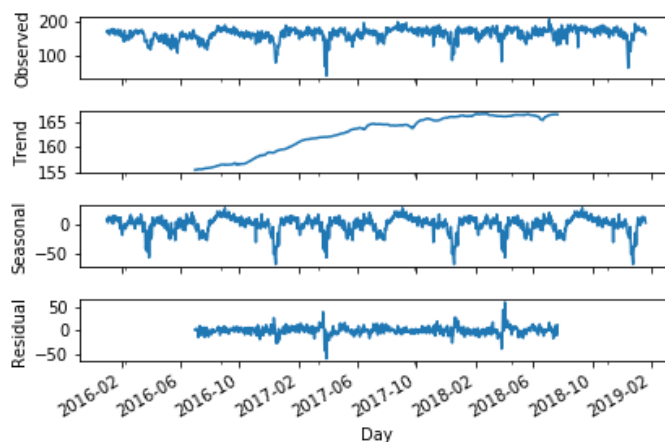
Ar šādu koda fragmentu (skat. 4.3. att.) tiek izpildīts *Dicker-Fuller* tests:

```
#Izpilda Dickey-Fuller testu:
print ('Results of Dickey-Fuller Test:')
dfctest = adfuller(timeseries['#Passengers'], autolag='AIC')
dfoutput = pd.Series(dfctest[0:4], index=['Test Statistic', 'p-value', '#Lags Used', 'Number of Observations Used'])
for key,value in dfctest[4].items():
    dfoutput['Critical Value (%s)'%key] = value
print (dfoutput)
```

4.3. att. Dicker-Fuller testa realizācija

4.7. Laika rindu dekompozīcija

Tālāk šajā modelī notiek laika rindu dalīšana (*decomposition*), lai varētu noteikt šīs datu kopas sēriju kā līmeņu kombināciju, tendenču, sezonālītes noteikšanai. Šī dalīšana sniedz noderīgu abstraktu modeli par laika rindām kopumā, un labāk izprotot problēmas laika rindu analizē un prognozēšanas laikā [2]. *Python* valodā tas tiek panākts ar komandas “*airpax_decompose = sm.tsa.seasonal_decompose(air_passanger_data['#Passengers'], model="additive", freq=365)*”. Šī komandrinda sadala datu kopu sezonāli pārvietojot vidējās vērtības, kā modelis tiek izmantots “*additive*”, jo to var uzskatīt kā rezultātu, kurā pievieno skaitļus. No iepriekš secinātā šī tipa datiem ir lineāra tendence, ir iespējams izmantot vēl “*multiplicative*” modeli, bet izmantojot šo, tam ir tendence uzradīt eksponenciālu tendenci. Ir nepieciešams veikt eksperimentus, lai noteiktu, kura metode ir labāka. Kā indeksu biežums (*freq*) tiek norādīts 365, jo tas atbilst dienu daudzumam gadā. Iegūtais 4.4. att. redzams zemāk:



4.4. att. Laika rindu dekompozīcija

Šī laika rinda parāda skaidru augšupejošu tendenci, bet neveido sezonālītāti.

4.8. Holt-Winter`s sezonālā izlīdzināšanas metodes izstrāde Python

Tiek pielietota *Holt-Winter`s* sezonālās izlīdzināšanas metode (*Holt-Winter`s Seasonal Smoothing method*), kura ir līdzīga vienkāršai eksponenciālai izlīdzināšanai. Tā aprēķina līmeņa komponentu, lai mērītu līmeni prognozē. Tas pievieno tendences un sezonālītātes komponentu.

Datu kopa tiek sadalīta treniņu un testa apmācību datu kopās – attiecībā 70% treniņu datu un 30% testa datu. Tas tiek izdarīt ar zemāk redzamā (skat. 4.5. att.) koda fragmenta palīdzību.

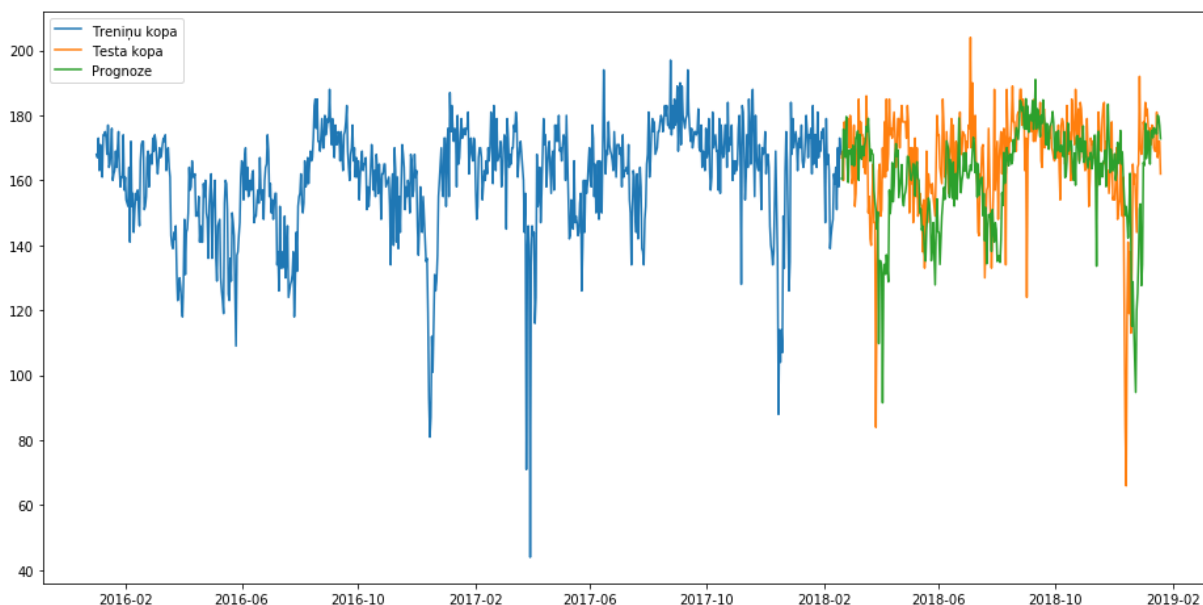
```
train=air_passanger_data[0:int(len(air_passanger_data)*0.70)]
test=air_passanger_data[int(len(air_passanger_data)*0.70):]
#Sadālīt treniņu un testa kopās (70%)

train_log = np.log10(train['#Passengers'])
```

4.5. att. Datu kopas sadalīšana testa un treniņu datu kopās.

Ar komandas “ExponentialSmoothing(np.asarray(train[‘#Passengers’]), seasonal_periods=365, trend=‘add’, seasonal=‘mul’).fit(optimizerd=True)” palīdzību, tiek pielietota trīskāršās eksponenciālās izlīdzināšanas metode, kurai kā parametri tiek padoti “seasonal_periods = 365” , jo dati ir ikdienišķi (daily), tiek pielietota iepriekš aprakstītā “additive” metode tendences tipam , sezonālītātes tips “multiplicative”. Ar komandas “airpax_Holt[‘Holt’]=model_airpax.forecast(len(test[‘#Passengers’]))” tiek veikta prognozēšana un ar “mean_absolute_percentage_error” aprēķināta vidējā absolūtā procentuālā kļūda, kas šajā gadījumā ir 8.623.

Grafiskos modeļa rezultātus var aplūkot zemāk redzamajā 4.6. att.



4.6. att. Modeļa grafiskie rezultāti

Tā kā var izmantot divu veidu tendenču metodes un divu veidu sezonālās metodes, tad lai noteiktu, kura darbojas labāk, ir nepieciešams katram reisam veikt eksperimentus, un tad tos salīdzināt. Par salīdzinošo mēru tiek pieņemta vidējā absolūtā procentuāla kļūda un šī eksperimenta rezultāti ir redzami 4.1. tabulā.

4.1. tabula

Reisa “W62501” modeļa salīdzinošie rezultāti

Reiss	Tendence	Sezonalitāte	Kļūda
W62501	additive	additive	8.91
	additive	multiplicative	8.62
	multiplicative	additive	9.35
	multiplicative	multiplicative	9.05

4.9. Drošības kontrolpunkta grafika veidošana

Kad katram reisam ir atrasta vispiemērotākā modeļa konfigurācija un veikta prognoze, tad ir nepieciešams tiem piemērot pasažieru skaita iekāpšanas grafikus. Par testa datumu tiek nejauši izvēlēts 2019. gada 10. jūlijs (tā vērtība datubāzē – ‘2019-07-10’), No ‘departures’ tabulas tiek eksportēts izlidojošo reisu saraksts šajā dienā. Katram reisam tiek veikta pasažieru skaita prognoze. Tiek izveidota tabula no trim kolonām. Ar reisa izlidošanas laiku, reisa numuru un prognozēto pasažieru skaitu. Šīm tabulas rindām ir jāpievieno pasažieru ierašanās laiki pirms reisa izlidošanas. Par piemērotāko tika atzīta metode, kur tiek vēsturiski ņemti vērā pēdējā

mēneša dati no viena un tā paša galamērķa un aviokompānijas - kopējais vidējais pasažieru profils galamērķim. Pasažieru skaita ierašanās profili tiek piemēroti procentuālā formā, jo tika atklātas tendences, ka pasažieru skaits dienu no dienas ir svārstīgs, bet kopējās ierašanās tendences ir nemainīgas. Prognozētās datu tabulas fragments ir redzams 4.2. tabulā.

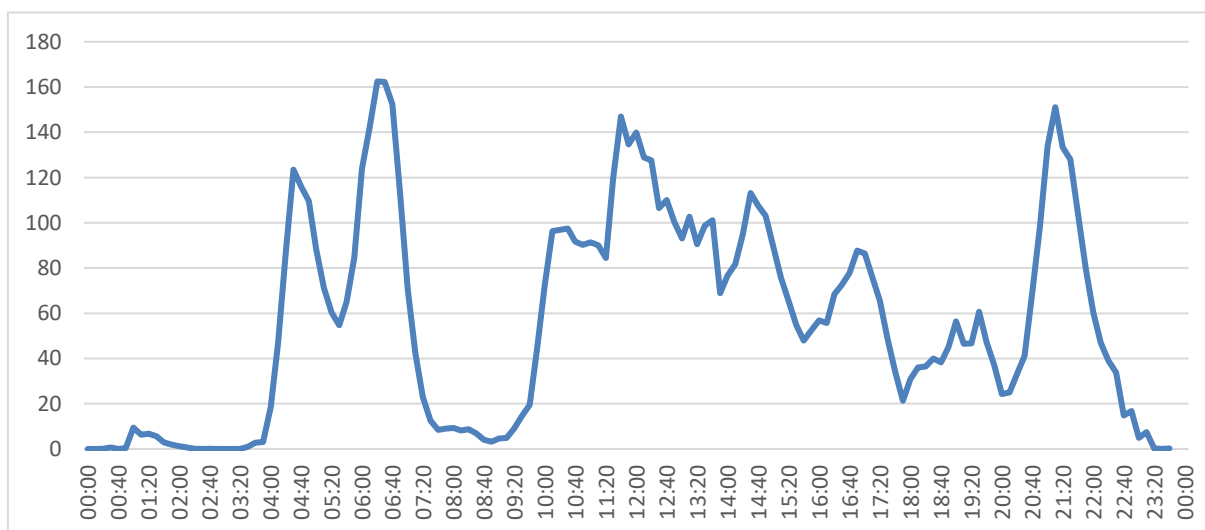
4.2. tabula

Pasažieru skaita prognozes un pasažieru ierašanās profilu prognožu apvienojums

Datums	Reiss	Prognoze	IER170	...	IER100	...	IER10
10.07.2019 05:50	BT7747	124	0%	...	21%	...	0%
10.07.2019 06:00	PS188	27	0%	...	14%	...	0%
10.07.2019 06:05	LO794	65	0%	...	10%	...	0%

Tālāk šī kolonas IER170 ... IER10 tiek pārveidotas skaitliskā formā. Tādā veidā skaidri tiek definēts, ka reisam *LO794* noapaļojot 100 – 110 minūtes pirms reisa drošības kontrolpunktā ieradīsies septiņi pasažieri.

Veicot attiecīgos aprēķinus un pārveidojums, ar intervālu 10 minūtes tiek izveidots darbā galvenais izvirzītais mērķis – kopējais aviācijas drošības punkta pasažieru ierašanās grafiks, kurš redzams zemāk (skat. 4.7. att.).



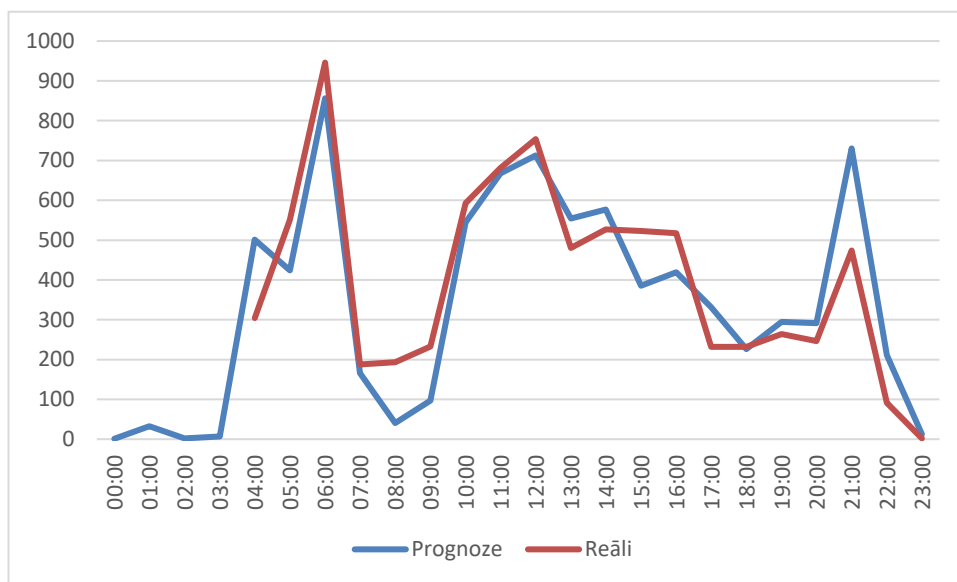
4.7. att Pasažieru skaita ierašanās grafiks drošības kontrolpunktā

Ar šī grafika palīdzību drošības departamenta darbinieki var ļoti viegli paredzēt rindas garumus, atkarībā no atvērto līniju skaita. Atkarībā no drošības departamenta un izvirzītajām pasažieru apkalpošanas kvalitātes prasībām, ir ļoti ērti saplānot, kad būs nepieciešams atvērt papildus drošības pārbaudes līnijas (noteikti tas būtu jādara pirms grafikā redzamo pīķu izveidošanās).

Šo grafiku visvieglāk ir interpretēt šādi – ja grafiks ir taisna līnija paralēla horizontālai asij, tad atvērto līniju skaitu var saglabāt (ja apmierina pasažiera gaidīšanas laiks rindā). Ja redzams, ka sagaidāms pasažieru skaitā pīķis, tad vienu vai divus intervālus pirms (10 vai 20 minūtes pirms sagaidāmā pīķa) ir nepieciešams palielināt līniju skaitu, un attiecīgi samazināt, kad tas ir nepieciešams.

5. REZULTĀTI UN SECINĀJUMI. MODEĻA UZLABOŠANA UN NĀKOTNES DARBS

Darbā izveidotais modelis izveido ticamu prognožu modeli aviācijas drošības kontrolpunktā, kas ļauj precīzāk plānot līniju atvēršanas grafiku noteiktā laikā. Zemāk redzamajā 5.1. att. parādīts prognozētā drošības kontrolpunkta noslodzes grafiks pa diennakts stundām – prognozētais un faktiskais.



5.1. att. Aviācijas drošības kontrolpunkta noslodzes grafiks

Grafikā ir visnotaļ precīzi prognozēta drošības kontrolpunkta noslodze pa diennakts stundām. Laiki, kad pasažieru skaits būs vislielākais ir noteikti pareizi, piemēram tas ir no plkst. 5:00 – 7:00. Neskatoties uz to, ka dažiem reisiem prognozes tika izteiktas samērā neprecīzas, tomēr reisiem, kuriem tas tika prognozēts par daudz, notika izlīdzināšanās ar reisiem, kuriem pasažieru skaits tika prognozēts par maz, nemainīja kopējo apkalpoto pasažieru skaitu kontrolpunktā. Par to liecina kopējais prognozētais pasažieru skaits diennaktī, kurš bija 8087, bet reāli 8031. Atšķirība ir 56 pasažieri, kas nekādu ietekmi uz pasažieru apkalpošanas kvalitāti neatstāj.

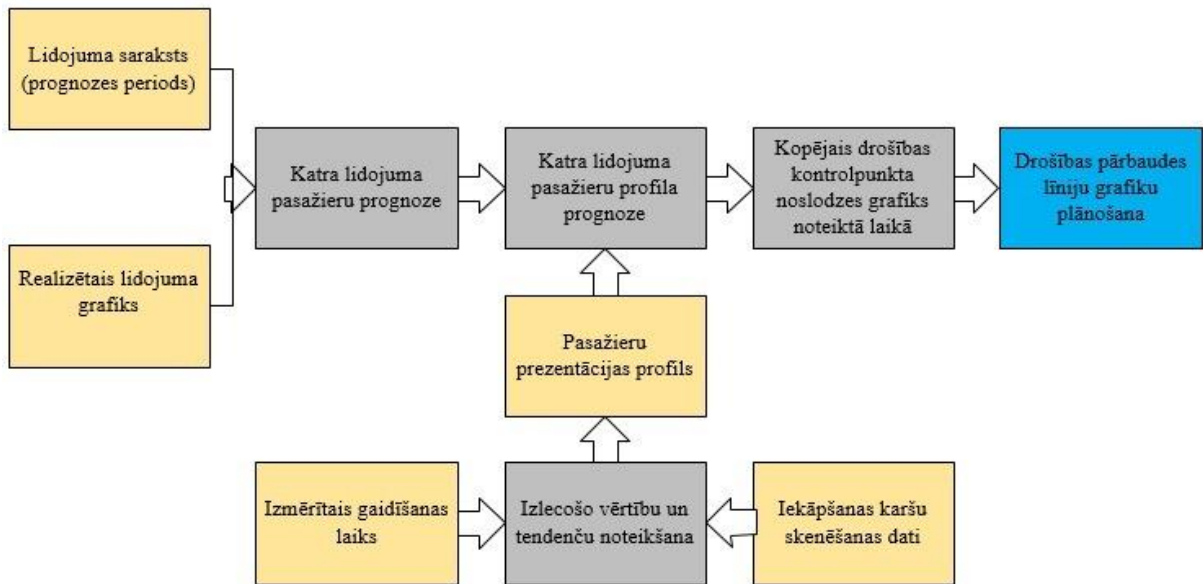
Pašreiz modelī ir jāiegulda ļoti liels manuālais darbs, lai varētu pēc iespējas precīzāk noteikt pasažieru skaitu, kas ieradīsies uz reisu. Ir jāveic testēšana gan sezonālajiem periodiem, gan izvēlei starp tendences un sezonālātes modeļiem, jo reisi cits no cita ir ļoti atšķirīgi, ko, pierāda arī veiktie eksperimenti, kas ļoti apgrūtina piemērotāko parametru izvēli. Pasažieru skaita prognožu modeli vajadzētu uzlabot tā, lai šie parametri tiktu noteikti automātiski. Piemēram, ir jāizveido tāds algoritms, kurš spēj atrast labākos parametrus un pēc tam tos pielietot modelī.

Diemžēl, ir tādi reisi, kuros pasažieru skaitu dienišķi prognozēt ar laika rindu prognozēšanas algoritmiem nav iespējams, jo šajos reisos pasažieru skaits ir pārāk svārstīgs, tiem nav nekādas tendences vai sezonalitātes, vai arī ir notikuši tikai daži reisi. Tāpēc prognozēt šo reisu pasažieru skaitu ir iespējams tikai ar 4.1 punktā norādītājām metodēm, piemēram, vidējo metodi ņemot pēdējos desmit izpildītos reisus.

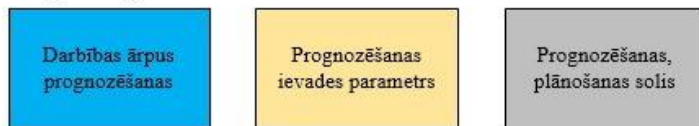
Modelī ir arī jāparedz situācijas, kad izlidojošo reisu sarakstā parādās, kāds jauns reiss. Uz jau vēsturiski notikušu galamērķi, kā arī jauns reiss uz pilnīgi jaunu galamērķi. Ja jau notikušam reisam jau ar zināmu galamērķi un zināmu aviokompāniju ir izveidots šis jaunākais reiss, tad situācija būtu vienkārša un varētu atsaukties uz jau zināmiem vēsturiskiem datiem. Sarežģīti būtu, ja parādās reiss, ar jaunu galamērķi, nezināmu aviokompāniju. Kā rīkoties šādās situācijās, būtu nepieciešams noskaidrot ar aviācijas nozares ekspertu viedokli.

Izveidoto modeli varētu uzlabot izmantojot reālā laika datus, un attiecīgi salīdzināt šos datus ar prognozēm. Tas ir, prognozes tiktu atjaunotas ik pēc noteikta laika perioda, piemēram, ik pēc desmit minūtēm. Noteikti būs situācijas, kad prognozētais pasažieru skaits būs lielāks vai mazāks par prognozēto, vai arī pasažieru ierašanās profils būs atšķirīgs no prognozētā, tādēļ jau pasažieru ierašanās laikā to varētu mēģināt laicīgi noteikt un kalibrēt, un attiecīgi iekļaut pārējās dienas prognozēs. Piemēram, situācijā, kad 140 minūtes pirms reisa ierodas mazāk pasažieru nekā prognozēts, tad ar kādu varbūtību šie pasažieri varētu ierasties 120 vai 100 minūtes pirms reisa. Attiecīgi pretēja darbība būtu, ja kādā laika posmā ierastos vairāk pasažieru nekā paredzēts, tad būtu jāuzdod jautājums, kuros laika periodos šo pasažieru būs mazāk, vai, ja ir situācija, kad kopējais prognozēto pasažieru skaits varētu būt lielāks nekā prognozētais. Tādējādi kopējais drošības kontrolpunkta noslodzes grafiks būtu dinamisks, kas spētu drošības departamenta darbiniekus laikus brīdināt par izmaiņām (iespējams nepieciešams atvērt papildus drošības pārbaudes līnijas, vai arī pretēji).

Procesa plūsma prognozēšanai ir izklāstīta dotajā procesa kartē (skat. 5.2. att.).



Apzīmējumi



5.2. att. Procesu plūsma

IZMANTOTĀ LITERATŪRA UN AVOTI

- [1] Barnett T.P., Preisendorfer R., *Origins and Levels of Monthly and Seasonal Forecast Skill for United States Surface Air Temperatures Determined by Canonical Correlation Analysis*. 1987, [tiešsaiste] – [atsauce 03.01.2021.]. Pieejams: https://journals.ametsoc.org/view/journals/mwre/115/9/1520-0493_1987_115_1825_oaloma_2_0_co_2.xml
- [2] Etienne B., *Time Series in Python – Exponential Smoothing and ARIMA processes*, 2019, [tiešsaiste] - [atsauce 14.09.2020.]. Pieejams: <https://towardsdatascience.com/time-series-in-python-exponential-smoothing-and-arima-processes-2c67f2a52788>
- [3] Hernandez B., Andrea V., *Airport Design and Situational Passenger Flows and Practices: Exploring Design as a Method in Copenhagen Airport*, 2020, [tiešsaiste] – [atsauce 15.12.2020.]. Pieejams: <https://sfx.aub.aau.dk/sfxaub?sid=pureportal&doi=10.1080/23800127.2020.1847397>
- [4] Hyndman J.R., Athanasopoulos G., *Forecasting: Principles and Practice*. 2013, [tiešsaiste] – [atsauce 03.01.2021.]. Pieejams: <https://otexts.com/fpp2/data-methods.html>
- [5] International Civil Aviation Organization, *Manual on Air Traffic Forecasting*, 2006, [tiešsaiste]. – [atsauce 15.12.2020.]. Pieejams: https://www.icao.int/MID/Documents/2014/Aviation%20Data%20Analyses%20Seminar/8991_Forecasting_en.pdf
- [6] Kiršners Arnis. *Īsu laika rindu un to raksturojošo parametru apstrādes sistēma prognozēšanas uzdevumiem*. Promocijas darbs. Rīga: RTU, 2015.
- [7] Menard S., *Applied logistic regression analysis, Second edition*, London, 2002., [tiešsaiste] – [atsauce 03.01.2021.]. Pieejams: https://books.google.lv/books?hl=lv&lr=&id=EAI1QmUUsbUC&oi=fnd&pg=PR5&dq=logistic+regression+analysis&ots=4VCKIZoXKP&sig=XHkFCXparx0bRddy_FuEw3nhNsA&redir_esc=y#v=onepage&q=logistic%20regression%20analysis&f=false
- [8] Paršutins S. Lekciju materiāli priekšmetā “Datu ieguve”. RTU, 2017.
- [9] Sukovs A., Aleksejeva L., Mekejeva, K., Borisovs A., *Datu ieguves Pamati* – Rīga: RTU, DITF, ITI, 2007.
- [10] Tan P. N., Steinbach M., Kumar V., *Introduction to Data Mining*. – Boston: Pearson Addison - Wesley, 2006.
- [11] Van der Maaten L., Hinton G., *Visualizing Data using t – SNE*. 2008, [tiešsaiste] – [atsauce 15.12.2020.]. Pieejams:

<https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf>

- [12] Van der Maaten L., Postma E., Van den Heric J., *Dimensionality Reduction: A Comparative Review*. Tilburg University, 2009, [tiešsaiste] – [atsauce 03.01.2021.].

Pieejams:

https://members.loria.fr/moberger/Enseignement/AVR/Exposes/TR_Dimensiereductie.pdf

- [13] Wilson E., *The 6 models used in forecasting algorithms*, Apr. 2019. [tiešsaiste]. – [atsauce 12.11.2020.]. Pieejams: <https://demand-planning.com/2019/04/22/forecasting-algorithms/>

PIELIKUMS

Laika rindu prognozēšanas modeļa izveide un analīze *Python* programmēšanas valodā

```
import pandas as pd
import datetime as dt
import numpy as np
from sklearn.metrics import mean_squared_error
from math import sqrt
import matplotlib.pyplot as plt
%matplotlib inline
from matplotlib.pylab import rcParams
rcParams['figure.figsize'] = 15, 6
from patsy import dmatrices
from pandas import Series
from matplotlib import pyplot
from datetime import datetime
from matplotlib.pyplot import figure

import statsmodels.api as sm

figure(num=None, figsize=(16, 12), dpi=80, facecolor='w', edgecolor='k')

from statsmodels.tsa.holtwinters import ExponentialSmoothing, SimpleExpSmoothing,
Holt

import sys
import warnings
import itertools
warnings.filterwarnings("ignore")

import statsmodels.tsa.api as smt
```

```

import statsmodels.formula.api as smf

from statsmodels.tsa.stattools import adfuller
from statsmodels.tsa.arima_model import ARIMA

air_passanger_data = pd.read_csv("../input/bt661/BT661.csv")
air_passanger_data.head()
#Pārveido Day no String uz DateTime formātu
air_passanger_data['Day'] =
pd.to_datetime(air_passanger_data['Day'],infer_datetime_format=True)
air_passanger_data = air_passanger_data.set_index(['Day'])
plt.xlabel('Diena')
plt.ylabel('Pasažieru skaits')
plt.plot(air_passanger_data)

#Pārbauda laika rindas stacionaritāti
def test_stationarity(timeseries):

    #Determining rolling statistics
    rolmean = timeseries.rolling(window=365).mean()
    rolstd = timeseries.rolling(window=365).std()

    #Plot rolling statistics:
    orig = plt.plot(timeseries, color='blue',label='Original')
    mean = plt.plot(rolmean, color='red', label='Rolling Mean')
    std = plt.plot(rolstd, color='black', label = 'Rolling Std')
    plt.legend(loc='best')
    plt.title('Rolling Mean & Standard Deviation')
    plt.show(block=False)

    #Perform Dickey-Fuller test:
    print ('Results of Dickey-Fuller Test:')
    dfstest = adfuller(timeseries['#Passengers'], autolag='AIC')

```

```

dfoutput = pd.Series(dfctest[0:4], index=['Test Statistic','p-value','#Lags
Used','Number of Observations Used'])
for key,value in dfctest[4].items():
    dfoutput['Critical Value (%s)%key] = value
print (dfoutput)

test_stationarity(air_passanger_data)

airpax_df1 = air_passanger_data.diff(periods=1)
airpax_df1.dropna(inplace=True)
test_stationarity(airpax_df1)

airpax_log = np.log10(air_passanger_data)
airpax_log.dropna(inplace=True)
test_stationarity(airpax_log)

airpax_log_df = airpax_log.diff(periods=1)
airpax_log_df.dropna(inplace=True)
test_stationarity(airpax_log_df)

airpax_decompose = sm.tsa.seasonal_decompose(air_passanger_data['#Passengers'],
model="additive", freq=12)
airpax_decompose.plot()
plt.show()

trend = airpax_decompose.trend
seasonal = airpax_decompose.seasonal
residual = airpax_decompose.resid

print("Trend \n",trend.head(24))
print("Seasonal \n",seasonal.head(24))
print("Residual \n",residual.head(24))

train=air_passanger_data[0:int(len(air_passanger_data)*0.70)]

```

```

test=air_passanger_data[int(len(air_passanger_data)*0.70):]
train_log = np.log10(train['#Passengers'])

def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

model_airpax =
ExponentialSmoothing(np.asarray(train['#Passengers']),seasonal_periods=20,
trend='mul',seasonal='add').fit(optimized=True)

airpax_Holt = test.copy()
airpax_Holt['Holt']=model_airpax.forecast(len(test['#Passengers']))
mean_absolute_percentage_error(test['#Passengers'],airpax_Holt['Holt'])

model_airpax.params

airpax_Holt['Pax'] = model_airpax.forecast(len(test['#Passengers']))
plt.figure(figsize=(16,8))
plt.plot(train['#Passengers'], label='Treniņu kopa')
plt.plot(test['#Passengers'], label='Testa kopa')
plt.plot(airpax_Holt['Holt'], label='Prognoze')
plt.legend(loc=0)

print (airpax_Holt['Pax'])

```

Maģistra darbs: **AVIĀCIJAS DROŠĪBAS PĀRBAUDES KONTROLPUNKTA
DARBĪBAS PLĀNOŠANAS OPTIMIZĀCIJA**

Darba teksta galīgā versija izgatavota 2021. gada 20. janvārī.

Ar savu parakstu apliecinu, ka pētījums veikts patstāvīgi, izmantoti tikai tajā norādītie informācijas avoti un iesniegtā darba elektroniskā kopija atbilst izdrukai.

Autors: Edgars Zēbergs

(Autora paraksts)

Ar savu parakstu apliecinu, ka esmu lasījis augstāk minēto maģistra darbu un atzīstu to par **p i e m ē r o t u / n e p i e m ē r o t u** (nevajadzīgo svītrot) aizstāvēšanai Latvijas Universitātes datorzinātņu maģistrantūrā.

Darba vadītājs: profesors Dr. math. Kārlis Podnieks

(Vadītāja paraksts)

Darbs iesniegts maģistrantūras sekretariātā

(Iesniegšanas datums)

Ar šo es apliecinu, ka darba elektroniskā versija ir augšupielādēta LU informatīvajā sistēmā.

Studiju metodiķe: _____.

(Metodiķes paraksts)

Recenzents: doc., Dr. dat. Agris Šostaks

(Akad. amats, zin. grāds, vārds, uzvārds)

Darbs aizstāvēts maģistra gala pārbaudījuma komisijas sēdē

_____ prot. Nr. _____

(Darba aizstāvēšanas datums)

Komisijas sekretārs: _____

(Sekretāra paraksts)