

Analiza opreme za prihvat i otpremu zrakoplova temeljena na stablu odlučivanja

Drobac, Lukša

Undergraduate thesis / Završni rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:119:701642>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-06**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences - Institutional Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI**

Lukša Drobac

**ANALIZA OPREME ZA PRIHVAT I OTPREMU ZRAKOPLOVA TEMELJENA NA
STABLU ODLUČIVANJA**

ZAVRŠNI RAD

Zagreb, 2021.

Sveučilište u Zagrebu
Fakultet prometnih znanosti

ZAVRŠNI RAD

**ANALIZA OPREME ZA PRIHVAT I OTPREMU ZRAKOPLOVA
TEMELJENA NA STABLU ODLUČIVANJA
ANALYSIS OF AIRCRAFT GROUND HANDLING EQUIPMENT BASED
ON DECISION TREE METHOD**

Mentor: doc. dr. sc. Matija Bračić

Student: Lukša Drobac
JMBAG: 0135248477

Zagreb, kolovoz 2021.

Zagreb, 11. svibnja 2021.

Zavod: **Zavod za zračni promet**
Predmet: **Tehnologija prihvata i otpreme zrakoplova**

ZAVRŠNI ZADATAK br. 6366

Pristupnik: **Lukša Drobac (0135248477)**
Studij: **Promet**
Smjer: **Zračni promet**

Zadatak: **Analiza opreme za prihvata i otpremu zrakoplova temeljena na stablu odlučivanja**

Opis zadatka:

U uvodnom dijelu rada potrebno je izraditi strukturu rada te napraviti pregled dosadašnjih istraživanja u predmetnoj problematici. U narednom poglavlju potrebno je prikazati i analizirati metodu stabla odlučivanja te navesti i opisati opremu koja se koristi u procesu prihvata i otpreme zrakoplova. Primijeniti metodu stabla odlučivanja u procesu klasifikacije i analize opreme za prihvata i otpremu zrakoplova. U posljednjem dijelu završnog rada prikazati zaključna razmatranja.

Mentor:

Predsjednik povjerenstva za
završni ispit:

doc. dr. sc. Matija Bračić

Sažetak

Proces prihvata i otpreme zrakoplova se sastoji velikog broja različitih aktivnosti. Za provođenje aktivnosti je potrebna odgovarajuća oprema koja se koristi ovisno o tipu zrakoplova, zračnog prijevoznika, zračne luke i drugo. U ovom radu je opisana oprema potrebna za aktivnosti prihvata i otpreme zrakoplova. Stablo odlučivanja je grafička metoda koja se koristi za klasifikaciju znanja na temelju prethodno prikupljenih podataka, a klasifikaciju znanja provodi algoritam. Postoji više tipova algoritama koji se razlikuju se prema načinu rada. U ovom radu je metodom stabla odlučivanja analizirana oprema za prihvata i otpremu zrakoplova. Analiza je napravljena u programu C#. Metoda stabla odlučivanja se izvršava na temelju CART algoritma.

Ključne riječi: prihvata i otprema zrakoplova, oprema za prihvata i otpremu zrakoplova, metoda stablo odlučivanja

Summary

The aircraft ramp handling process consists of several activities. Activities are performed with the appropriate equipment, and this equipment depends on the type of aircraft, air carrier, airport, and others. This paper describes the equipment necessary for the aircraft ramp handling process. The decision tree is a graphic method used for knowledge classification based on previously collected data. The algorithm performs knowledge classification. There are several types of algorithms, which have different operation modes. This paper uses the decision tree method for the analysis of equipment for receiving and dispatching the aircraft. The analysis is made in the C# program. The decision tree method is based on the CART algorithm.

Key words: aircraft ramp handling process, equipment aircraft ramp handling process, decision tree method

SADRŽAJ

| | |
|---|----|
| 1. UVOD..... | 1 |
| 2. METODA STABLO ODLUČIVANJA | 2 |
| 2.1. Algoritmi stabla odlučivanja..... | 3 |
| 2.1.1. ID3 algoritam | 3 |
| 2.1.2. C 4.5 algoritam..... | 4 |
| 2.1.3. CART algoritam | 5 |
| 2.1.4. CHAID algoritam | 6 |
| 2.1.5. MARS algoritam | 6 |
| 2.2. Obrezivanje stabla odlučivanja | 7 |
| 2.3. Primjena stabla odlučivanja | 8 |
| 3. OPREMA ZA PRIHVAT I OTPREMU ZRAKOPLOVA..... | 10 |
| 3.1. Zračni most..... | 10 |
| 3.2. Putničke stepenice..... | 11 |
| 3.2.1. Vučene stepenice..... | 12 |
| 3.2.2. Samohodne stepenice | 12 |
| 3.3. Autobusi | 13 |
| 3.4. Liftmobil | 13 |
| 3.5. Podmetači | 14 |
| 3.6. Agregati | 14 |
| 3.7. Catering vozilo | 16 |
| 3.8. Kolica za prijevoz prtljage..... | 16 |
| 3.9. Mobilna pokretna traka..... | 17 |
| 3.10. ULD kolica..... | 17 |
| 3.11. Utovarivač | 18 |
| 3.12. Transporter | 18 |
| 3.13. Vučna vozila | 19 |
| 3.14. Rudo za vuču | 20 |
| 3.15. Adapter | 20 |
| 3.16. Vozilo za servis otpadnih voda | 20 |
| 3.17. Vozilo za opskrbu zrakoplova vodom | 21 |
| 3.18. Cisterna za opskrbu zrakoplova gorivom | 22 |
| 3.19. Uređaj za klimatizaciju..... | 22 |
| 3.20. Viličar..... | 23 |
| 3.21. Zračni starter | 23 |
| 3.22. Vozilo za odleđivanje i zaštitu zrakoplova od zaleđivanja | 24 |

| | |
|--|-----------|
| 4. ANALIZA OPREME ZA PRIHVAT I OTPREMU ZRAKOPLOVA METODOM STABLA ODLUČIVANJA | 25 |
| 4.1. Klase i metode stabla odlučivanja..... | 26 |
| 4.2. Rezultati primjene stabla odlučivanja u analizi opreme za prihvati i otpremu zrakoplova | 29 |
| 4.2.1. Stablo odlučivanja za samohodne stepenice | 29 |
| 4.2.2. Stablo odlučivanja za vučene stepenice..... | 30 |
| 4.2.3. Stablo odlučivanja za catering vozilo | 31 |
| 4.2.4. Stablo odlučivanja za cisternu za vodu | 33 |
| 4.2.5. Stablo odlučivanja za utovarivač..... | 34 |
| 5. ZAKLJUČAK | 37 |
| POPIS LITERATURE | 38 |
| POPIS KRATICA | 41 |
| POPIS SLIKA | 42 |
| POPIS TABLICA | 43 |

1. UVOD

Prihvat i otprema zrakoplova predstavlja niz aktivnosti kojima se završava prethodni let zrakoplova, te se zrakoplov priprema za idući let. Cilj ovog procesa je obaviti aktivnosti sigurno te u zadanom vremenu, jer kašnjenje koje se potencijalno može javiti, uzrokuje gubitke za zračnog prijevoznika i zračnu luku. Za obavljanje ovih aktivnosti je potrebna odgovarajuća oprema. Korištenje opreme je uvjetovano tipom zrakoplova, zračnim prijevoznikom, zračnom lukom i drugim čimbenicima.

Stablo odlučivanja je alat koji se izvodi kroz grafičku klasifikaciju znanja. Način rada se temelji na rekurzivnoj podjeli prethodno pripremljenih podataka. Podjelu podataka provodi algoritam. Postoji više vrsta algoritama koji se razlikuju prema načinu podjele podataka.

Cilj ovog rada je izraditi program koji izvršava analizu opreme za prihvat i otpremu zrakoplova temeljem stabla odlučivanja. Naslov ovog rada je „*Analiza opreme za prihvat i otpremu zrakoplova temeljena na stablu odlučivanja*“, te se rad sastoji od pet poglavlja:

1. Uvod
2. Metoda stabla odlučivanja
3. Oprema za prihvat i otpremu zrakoplova
4. Analiza opreme za prihvat i otpremu zrakoplova temeljena na stablu odlučivanja
5. Zaključak

U uvodnom poglavlju je definiran problem te je opisana struktura završnog rada.

Drugo poglavlje se bavi problematikom stabla odlučivanja. Opisan je način rada stabla odlučivanja te svi dijelovi stabla. Navode se prednosti i nedostaci korištenja stabla odlučivanja te je napravljena podjela stabla odlučivanja. Istraženi su algoritmi na kojima se temelji stablo odlučivanja, te način rada svakog algoritma. Opisano je nekoliko primjera primjene stabla odlučivanja.

Treće poglavlje se bavi opremom potrebnom za provođenje aktivnosti prihvata i otpreme zrakoplova. Navode se ciljevi zračnih luka i prijevoznika vezani za provođenje aktivnosti prihvata i otpreme zrakoplova. Napravljena je podjela opreme prema osnovnim grupama aktivnosti koje se izvode na zrakoplovu. Za svu opremu se pojedinačno navode dijelovi, karakteristike, namjena te način rada.

U četvrtom poglavlju je opisan program koji obavlja analizu opreme za prihvat i otpremu zrakoplova temeljenu na stablu odlučivanja. Prikazani su svi dijelovi programa i njihova uloga i njegov način rada. Navedena je oprema za koju se program može primijeniti, te su prikazani rezultati analize za tu opremu.

U posljednjem poglavlju su izneseni zaključci izvedeni na temelju analiza provedenih u radu.

2. METODA STABLO ODLUČIVANJA

Stablo odlučivanja je alat koji se koristi za klasifikaciju znanja te zaključivanje na temelju prethodno klasificiranog znanja, a cilj alata je stvoriti model koji će klasificirati znanje na osnovu poznatih primjera. Ova metoda grafički prikazuje klasifikaciju znanja. Način rada metode se temelji na rekurzivnoj podjeli skupa podataka na podskupove podataka. Podjela se izvodi sve dok se ne dođe do primjera koji se ne može dalje dijeliti, te predstavlja konačnu klasifikaciju znanja.

Podjela podataka se izvodi na temelju atributa koji omogućavaju najučinkovitiju podjelu, pritom svaki primjer ima vrijednost za pojedini atribut. Taj atribut se računski dobiva na osnovu jednog od mogućih algoritama koji se koriste u stablu odlučivanja. Algoritam uključuje sve moguće atribute. Svi atributi u stablu odlučivanja su međusobno povezani, te se važnost svakog atributa može računski prikazati.

Stablo odlučivanja se sastoji od tri tipa čvora: korijenski čvor, čvor odluke i lisni čvor. Korijenski čvor uključuje cjelokupni skup primjera koji su dostupni za rješavanje problema. U ovom čvoru se cjelokupni skup podataka dijeli na više podskupova. Podjela podataka u korijenskom čvoru predstavlja najbolju podjelu u cijelom procesu, jer su uključeni svi primjeri. Grane koje proizlaze iz čvorova predstavljaju pojedine vrijednosti atributa u procesu klasifikacije znanja. Čvorovi odluke su čvorovi koji se nastavljaju na korijenski čvor. Rekurzivno dijele skupove na više podskupova, sve dok se ne budu mogli dalje dijeliti. Čvor nakon kojeg se podaci više ne mogu dijeliti se naziva lisni čvor. Mogućnosti koje proizlaze iz lisnog čvora predstavljaju konačnu klasifikaciju znanja [1].

Čvorovi se mogu podijeliti na: roditeljski čvor i dječji čvor. Roditeljski čvor je čvor iz kojeg se grana jedan ili više čvorova. Čvor ili čvorovi koji se granaju iz roditeljskog čvora su dječji čvorovi. Svaki čvor u stablu odlučivanja je roditeljski i dječji čvor, osim korijenskog i lisnog čvora. Korijenski čvor je uvijek roditeljski čvor, dok je lisni uvijek dječji čvor [2].

Stablo odlučivanja se dijeli na dva tipa: klasifikacijsko i regresijsko. Klasifikacijska stabla odlučivanja koriste nominalne vrijednosti skupa podataka, a regresijska brojeve vrijednosti. Nominalni atributi imaju konačan broj vrijednosti. Kada se koriste nominalni atributi svaki čvor može imati onoliko grana koliko ima mogućih vrijednosti. Kod brojčanih atributa, na osnovu skupa podataka, odabere se vrijednosti prema kojoj se skup podataka dijeli na više podskupova [1].

Prednosti stabla odlučivanja [3]:

- učinkovit način klasifikacije znanja,
- olakšava proces podjele podataka,
- koristi se kao pomoć u drugim alatima za klasifikaciju znanja,
- može se koristiti u primjerima s brojčanim i nominalnim vrijednostima,
- pojednostavljuje proces klasifikacije znanja.

Nedostaci stabla odlučivanja [3]:

- nije učinkovita u slučaju velike količine podataka,
- nije učinkovita u slučaju velikog broja mogućih rezultata,
- postoji mogućnost krive klasifikacije vrijednosti,
- promjene u bilo kojem dijelu stabla utječu na konačne rezultate,
- pristranost pri podjeli podataka utječe na konačne rezultate.

2.1. Algoritmi stabla odlučivanja

Postoji više tipova algoritama za izradu stabla odlučivanja. Algoritmi se razlikuju prema parametrima koji se koriste za podjelu podataka i imaju svoje prednosti i nedostatke te se na temelju istih koriste. Mogu se koristiti za izradu klasifikacijskih i/ ili regresijskih stabala odlučivanja. Neki su pogodniji za izradu jednostavnijih stabala odlučivanja, a neka za izradu složenijih stabala odlučivanja. U ovom dijelu je opisano pet algoritama: ID3, C 4.5, CART, CHAID i MARS.

2.1.1. ID3 algoritam

ID3 (engl. Iterative Dichotomiser 3) je izradio J. Ross Quinlan 1986. godine. Algoritam stvara stablo odlučivanja od vrha prema dnu, te za svaki čvor ponavlja odabir najboljeg mogućeg atributa. ID3 algoritam za izradu stabla odlučivanja može koristiti dva parametra: entropiju i informacijsku dobit [4].

Entropija mjeri količinu neizvjesnosti u skupu podataka. Vrijednost entropije će biti nula kada su podaci homogeni, a kada su podaci jednako podijeljeni tada je vrijednost entropije jedan. Prilikom izrade stabla odlučivanja koriste se modeli s manjom entropijom jer omogućavaju lakše predviđanje. Entropija se računa prema formuli [5]:

$$Entropija(S) = -\sum p_i * \log_2(p_i); i = od 1 do n$$

gdje oznake imaju sljedeće značenje:

- n – ukupan broj klasa
- p_i – vjerojatnost klase i

Informacijska dobit je vrijednost koja pokazuje za koliko se smanjila neizvjesnost u skupu podataka S nakon podjele skupa podataka S prema atributu A . Informacijska dobit je suprotna entropiji, tj. što je veća informacijska dobit, manja je entropija i obratno. Ishod koji ima veliku vjerojatnost događanja, ima veliku entropiju i malu informacijsku dobit. Ishod s malom vjerojatnosti događanja ima veliku informacijsku dobit te malu entropiju. Formula za računanje informacijske dobiti je [5]:

$$IG(S, A) = Entropija(S) - \sum \left(\frac{|S_V|}{|S|} \right) * Entropija(S_V)$$

gdje oznake imaju sljedeće značenje:

- S_v – primjeri u skupu S koje za atribut A imaju vrijednost v
- $|S_v|$ - broj primjera skupa S_v
- S – broj primjera skupa S .

Izrada stabla odlučivanja započinje računanjem entropije za cjelokupni skup podataka. Zatim se za svaki atribut računa entropija za sve tipove podataka te informacijska dobit svakog atributa. Za izradu čvora se odabire atribut s najvećom trenutnom informacijskom dobiti ili s najmanjom entropijom. Postupak se ponavlja sve dok se ne iskoriste svi atributi ili dok svi dobiveni čvorovi ne budu lisni.

Karakteristike ID3 algoritma [6]:

- ne može uvijek dati optimalno rješenje,
- nije učinkovit kada se koriste veliki skupovi podataka,
- koristi se za izradu malih stabala odlučivanja,
- nije prikladan za korištenje s brojčanim vrijednostima.

2.1.2. C 4.5 algoritam

C 4.5 algoritam je napravio J. Ross Quinlan 1992. godine te predstavlja napredak u odnosu na ID3 algoritam [5]. C 4.5 je bio prvi algoritam koji se često primjenjivao te postizao dobre rezultate u predviđanju konačnih ishoda. Razlika u odnosu na ID3 algoritam je upotreba faktora dobiti kao parametra umjesto informacijske dobiti. Nedostatak informacijske dobiti je taj što teži odabiru atributa s više različitih vrijednosti, dok faktor dobiti uzima u obzir broj dječjih čvorova koji nastaju iz roditeljskih čvorova. Formula za računanje faktora dobiti [7]:

$$Faktor\ dobiti = \frac{Entropija(prije) - \sum_{j=1}^K Entropija(j, poslije)}{\sum_{j=1}^K w_j \log_2 w_j}$$

$$w_j = \frac{broj\ primjera\ u\ podskupu(j, poslije)}{broj\ primjera\ u\ skupu(prije)}$$

gdje oznake imaju sljedeće značenje:

- K – broj podskupova koji su nastali podjelom

Postupak odabira atributa za podjelu podataka koristeći faktor dobiti [8]:

- određivanje informacijske dobit svih atributa,
- izračun srednje vrijednost informacijske dobiti,
- ako je informacijska dobit atributa veća od srednje vrijednost informacijske dobiti, računa se faktor dobiti za te attribute,
- odabire se atribut s najvećim faktorom dobiti za podjelu podataka.

Prednosti C 4.5 algoritma [9]:

- koristi postupak obrezivanja da bi se onemogućio prekomjeren broj podataka na stablu odlučivanja,
- može izraditi stablo odlučivanja na temelju diskretnih i neprekidnih tipova podataka,
- može se koristiti i u slučaju nekompletnih podataka.

2.1.3. CART algoritam

CART (engl. Classification and Regression Trees) je naziv za algoritam koji se koristi za izradu klasifikacijskih i regresijskih stabala odlučivanja. Taj naziv je prvi uveo Leo Breiman [10]. Uvijek obavlja podjelu podataka na dva dijela dok se temelji na podjeli podataka putem gini indeksa. Gini indeks predstavlja mjerilo različitosti podataka, tj. broj različitih tipova podataka. Gini indeks je jako sličan entropiji, ali se jednostavnije i brže računa. Smatra se najjednostavnijim i najlakšim načinom podjele podataka u stablu odlučivanja. Kada slučajnim odabirom se odabere jedan podatak iz čvora, gini predstavlja vjerojatnost da je odabrani podatak ujedno i ispravan podatak. Homogenost podataka u čvoru je veća, što je gini indeks čvora manji i obratno. Podaci u čvoru su u potpunosti homogeni, ako je gini indeks jednak 0. Gini indeks se računa pomoću formule [11]:

$$gini = 1 - \sum_{i=1}^c (p_i)^2$$

gdje oznake imaju sljedeće značenje:

- p_i – vjerojatnost da element pripada klasi C

Koraci u izradi stabla odlučivanja pomoću CART algoritma [12]:

- za svaki atribut je potrebno pronaći najbolju moguću podjelu podataka. Za svaki atribut s K različitih vrijednosti postoji $K - 1$ mogućih podjela.
- za svaki čvor se odabire najbolja moguća podjela podataka. Nakon što se odabire najbolja moguća podjela podataka za svaki atribut, potrebno je odabrati atribut s najboljom mogućom podjelom podataka.
- obavlja se podjela podataka s najboljim mogućim atributom. Ponavljaju se svi koraci dok se ne ispune kriteriji za zaustavljanje.

Najčešći kriteriji za zaustavljanje postupka rekurzivne podjele podataka je određivanje minimalnog broja primjera za svaki terminalni čvor. U slučaju da je broj primjera manji od minimalnog broja primjera, prethodni čvor postaje terminalni čvor te prestaje daljnja podjela podataka. Nakon što se izradi stablo odlučivanja, kao i kod C 4.5 algoritma može se koristiti postupak obrezivanja da bi se poboljšala učinkovitost stabla odlučivanja [10].

CART algoritam je jednostavan u usporedbi s drugim algoritmima za izradu stabla odlučivanja. Najčešće se koristi za izradu jednostavnijih stabala odlučivanja. CART algoritam predstavlja osnovu za razvoj naprednijih algoritama poput „slučajne šume“.

2.1.4. CHAID algoritam

CHAID (engl. Chi – square Automatic Interaction Detection) je najstariji algoritam za izradu stabla odlučivanja. Osmislio ga je Gordon V. Kass 1980. godine [13], a može se učinkovito upotrebljavati u slučajevima s bilo kojim tipom podataka. Može obavljati podjelu podataka na više od dva dijela. CHAID algoritam treba veliku količinu podataka da bi se primijenio, što ga čini složenijim od drugih algoritama. Ako su podaci kategorične vrijednosti za izradu se koristi hi – kvadrat test, a u slučaju da su podaci neprekidne vrijednosti koristi se f – test [14].

Hi – kvadrat test se primjenjuje u svakoj fazi CHAID algoritma. Hi – kvadrat test mjeri statističke razlike između dječjeg i roditeljskog čvora. Vrijednosti u roditeljskom čvoru se dijele na klase. Klasa s najvećim brojem podataka, predstavlja *očekivanu* klasu. *Stvarna* klasa je klasa vrijednosti koja se pojavi u dječjem čvoru. Hi – kvadrat vrijednost može biti bilo koja vrijednost u intervalu od nula do jedan. Što je vrijednost bliža nuli, to je veća razlika između klasa, a ako je vrijednost bliže jedan, to znači da razlika između klasa nije velika [15].

Hi – kvadrat vrijednost za pojedinu klasu se dobiva formulom [15]:

$$hi - kvadrat = \sqrt{\frac{(y - y')^2}{y'}}$$

gdje oznake imaju sljedeće značenje:

- y – stvarna vrijednost
- y' - očekivana vrijednost

2.1.5. MARS algoritam

MARS (engl. Multivariate Adaptive Regression Splines) je algoritam koji se koristi za izradu regresijskih stabala odlučivanja, tj. vrijednosti skupa podataka su brojčanog tipa. Ovaj algoritam se koristi za rješavanje složenih problema koji uključuju veliki broj ulaznih podataka. MARS algoritam je skup linearnih funkcija. Te funkcije nastaju prema karakteristikama dostupnih podataka te se sastoje od manjih funkcija. Izlazne vrijednosti funkcija mogu biti jednake nuli ili ulaznoj vrijednosti. Dije se na desnu i lijevu funkciju. Za svaku funkciju se uspoređuje ulazna vrijednost s unaprijed odabranom vrijednošću. U slučaju desne funkcije ako je ulazna vrijednost manja od odabrane vrijednosti, izlazna vrijednost će biti nula, a ako je veća, tada će izlazna vrijednost biti jednaka ulaznoj vrijednosti. U slučaju lijeve funkcije postupak je obrnut, ako je vrijednost manja, izlazna vrijednost će biti jednaka ulaznoj, a ako je veća biti će nula [16].

Desna funkcija: $f(x) = x$ ako je $x > 0$, inače 0

Lijeva funkcija: $f(x) = x$ ako je $x < 0$, inače 0 [16]

Ovakav tip funkcije se naziva zavisna funkcija ili „spline“ funkcija, a odabrana vrijednost s kojom se uspoređuju ostale vrijednosti se naziva čvor. Važnost svake izlazne vrijednosti je određena koeficijentom. Rješenje se dobiva zbrajanjem svih dobivenih vrijednosti [16].

MARS algoritam se sastoji od dva dijela: dio u kojem se kreće naprijed i dio u kojem se kreće prema natrag. Dio u kojem se napreduje uključuje stvaranje i dodavanje funkcija u stablo odlučivanja. Same funkcije se dodaju u parovima, kao lijeva i desna funkcija. Funkcije se dodaju ako smanjuju grešku cjelokupnog modela. U dijelu u kojem se kreće prema unatrag odabiru se funkcije koje se uklanjaju iz modela, jedna po jedna. Funkcije se uklanjaju ako nemaju utjecaj na konačno rješenje. Utjecaj na konačno rješenje se procjenjuje unakrsnom provjerom svih vrijednosti. Ukupan broj funkcija u modelu se određuje unaprijed, pa se uklanjanje funkcije zaustavlja kada se postigne taj broj [16].

Prednosti MARS algoritma [17]:

- ako su linearne veze ispravne, algoritam je jako precizan,
- vrijeme računanja je kratko,
- može se koristiti s velikim i malim brojem podataka,
- automatiziran odabir atributa,
- nelinearna veza između atributa i rješenja je lako razumljiva,
- koristi se za klasifikacijske i regresijske probleme,
- nije potrebna standardizacija atributa.

Nedostatci MARS algoritma [17]:

- ako linearne veze nisu ispravne, algoritam nije precizan,
- nije precizan kao napredniji nelinearni algoritmi,
- ako se ustanovi da određene vrijednosti nedostaju, algoritam se ponavlja, jer se te vrijednosti moraju obraditi prije izvođenja algoritma.

2.2. Obrezivanje stabla odlučivanja

Kada se pri izradi stabla odlučivanja koriste skupovi podataka s velikim brojem vrijednosti, postoji mogućnost dobivanja jako složenih stabla odlučivanja s jako puno grana i čvorova. Veliki broj grana i čvorova smanjuje učinkovitost stabla odlučivanja, jer neke grane predstavljaju jako mali broj podataka. Da bi se smanjio broj grana i čvorova, te povećala učinkovitost stabla odlučivanja koristi se postupak obrezivanja. Obrezivanje u prirodi znači uklanjanje dijelova stabla, tj. grana da se stablo bolje razvija, te se ista definicija može primijeniti na stablu odlučivanja. Obrezivanjem se uklanjaju čvorovi i grane koje ne uključuju dovoljan broj podataka.

Obrezivanje može započeti iz lisnog čvora ili korijenskog čvora. Postoje dva tipa obrezivanja: predobrezivanje i postobrezivanje. Postobrezivanje se izvršava nakon što se izradi stablo odlučivanja, te se započinje iz lisnog čvora. U procesu postobrezivanja izvršava se postupak unakrsne provjere točnosti stabla odlučivanja.

Taj postupak znači da se stablo odlučivanja izrađuje na temelju većine dostupnih podataka, a s ostatkom dostupnih podataka se provjerava točnost stabla. Kada se pronade čvor s najmanjom pogreškom, uklone se svi čvorovi koji se granaju iz njega. Predobrezivanje je proces koji zaustavlja grananje stabla odlučivanja prije nego što nastanu čvorovi s nedovoljnim brojem podataka. Nakon svakog grananja u stablu odlučivanja se unakrsno provjerava točnost čvora. Ako se greška nije smanjila, prestaje se s daljnjim grananjem stabla odlučivanja. Kada se koristi postupak predobrezivanja, konačno stablo odlučivanja može biti prejednostavno, odnosno s malim brojem grana. Predobrezivanje i postobrezivanje se može koristiti zajedno, odvojeno ili ne koristiti [18].

2.3. Primjena stabla odlučivanja

Stabla odlučivanja se zbog učinkovitosti te razumijevanja problema kojeg rješava koristi u raznim industrijama i disciplinama. Metodu stabla odlučivanja koriste mnoge organizacije pri donošenju važnih strateških odluka. Može se prilagoditi istraživanju, planiranju i predviđanju svakog mogućeg ishoda procesa donošenja odluke. Grafičkim prikazom svih mogućih rješenja i posljedica se osigurava proces od nepotrebnih rizika te neželjenih rezultata te se olakšava proces donošenja odluka. Pojedincu ili grupi omogućava procjenu mogućih odluka na temelju njihovih troškova, vjerojatnosti i koristi. Kada se traži najpovoljniji ishod, trebaju se uzeti u obzir prioritete pojedinca ili grupe. Jedni traže sigurne opcije s manje rizika, dok drugi žele preuzeti rizik da bi imali veću korist.

Primjer metode stabla odlučivanja može se pronaći u jednom istraživanju iz 2007. godine na području online trgovine. Proučavala se veza između potreba kupaca i same kupnje proizvoda. Kupci su podijeljeni u dvije grupe na temelju učestalosti korištenja online trgovine. Prva grupa su oni koji se rijetko koriste online trgovinom, a druga oni koji se često koriste online trgovinom. U prvom slučaju, najvažniji atributi za kupnju proizvoda su vrijeme trajanja transakcije te vremenska potreba za proizvodom. U drugom slučaju, najvažniji atributi su cijena te broj potrebnih kontakata s zaposlenicima. Konačni zaključak stabla odlučivanja je da kupnja proizvoda najviše ovisi učestalosti korištenja online trgovine i cijeni proizvoda [19].

Stablo odlučivanja se može koristiti u raznim područjima inženjerstva. Jedna analiza iz 2007. godine je istraživala vezu između kućanstava i potrošnje električne energije u Hong Kongu za vrijeme ljeta. Istraživanje je pokazalo da je najvažniji atribut broj članova kućanstva, a zatim slijede broj klima uređaja te veličina stana. Na temelju istraživanja putem stabla odlučivanja je zaključeno da su kućanstva s četiri ili više osoba te sa stanom većim od 76 m² biti najveći potrošači električne energije, dok su kućanstva s manje od četiri osobe i bez klima uređaja najmanji potrošači [19].

Metoda se često koristi u zdravstvu. Primijenjena je u istraživanju koje je proučavalo djecu zaostalu u razvoju. Istraživala se povijest bolesti takve djece, te kakav utjecaj pojedine bolesti imaju na razvoj djece. Metoda stabla odlučivanja je pokazala da većina bolesti ima utjecaj na psihički razvoj djece, govorne i motoričke sposobnosti.

Ova studija omogućuje liječnicima da ranije interveniraju kod ovih bolesti, te omogućuje zaostaloj djeci da sustignu svoje vršnjake u razvoju [19]. Metoda stabla odlučivanja je korištena 2005. godine te je predviđala da li će pacijenti oboljeli od raka dojke preživjeti. Preciznost metode je bila 93,6%, te je jako precizno predviđala ishod bolesti. Stablo odlučivanja se pokazalo jako korisnim i preciznim alatom za primjenu u raznim područjima zdravstva [19].

3. OPREMA ZA PRIHVAT I OTPREMU ZRAKOPLOVA

Prihvat i otprema zrakoplova predstavlja niz aktivnosti koje se obavljaju na zrakoplovu koji je sletio u zračnu luku. Vrijeme prihvata i otpreme (engl. Turn Round Time -TRT) je vrijeme potrebno za obavljanje svih aktivnosti na zrakoplovu. Glavni cilj proizvođača zrakoplova, zračnih luka i zračnih prijevoznika je smanjiti potrebno vrijeme prihvata i otpreme zrakoplova. Proizvođači zrakoplova dizajniraju zrakoplove na način da se izvođenje aktivnosti prihvata i otpreme zrakoplova pojednostavi. Zračne luke i prijevoznici razvijaju brže i učinkovitije procedure obavljanja aktivnosti prihvata i otpreme, bez da ugrožavaju sigurnost zrakoplova, opreme, putnika i osoblja. Samo vrijeme prihvata i otpreme zrakoplova se može smanjiti izvođenjem više aktivnosti paralelno. Međutim, postoje aktivnosti koje se ne mogu izvoditi paralelno zbog sigurnosnih ili higijenskih razloga [20].

Oprema za prihvat i otpremu zrakoplova su specijalizirana sredstva koja se koriste za obavljanje različitih aktivnosti prihvata i otpreme zrakoplova. Navedenom opremom mora upravljati adekvatno obučeno osoblje koje slijedi propisane procedure. Prije dolaska svakog zrakoplova, sva oprema potrebna za obavljanje aktivnosti prihvata i otpreme zrakoplova se mora pregledati da bi se ustanovilo je li ispravna za upotrebu. Kada pojedina oprema obavi svoju aktivnost, mora se udaljiti od zrakoplova iz sigurnosnih razloga. Oprema za prihvat i otpremu zrakoplova se može podijeliti u četiri skupine [21]:

- za ulazak i izlazak putnika,
- za utovar ili istovar prtljage,
- za utovar ili istovar tereta,
- za servis zrakoplova.

3.1. Zračni most

Zračni most (slika 1.) je zatvorena struktura koja povezuje terminal i zrakoplov. Predstavlja najlakši i najsigurniji način za ulazak/ izlazak putnika i posade iz zrakoplova. Štiti putnike od hladnoće, kiše ili snijega. Putnici se ne kreću po platformi, što olakšava odvijanje operacija na zemlji poput punjenja goriva te ukrcaja/ iskrcaja tereta. Jedan kraj zračnog mosta je fiksno povezan s terminalom, dok se drugi proteže do vrata zrakoplova. Zračni most se može biti:

- pomični,
- nepomični,
- djelomično pomični,
- fiksni.

Visina na kojoj se nalaze vrata zrakoplova ovisi o tipu zrakoplova, tako da operater mora prilagođavati visinu mosta zrakoplovima.

Zračni most se može kretati u vertikalnom smjeru (dizanje ili spuštanje), horizontalnom smjeru (pomicati lijevo ili desno) te produžiti ili skratiti. Nakon odlaska zrakoplova zračni most se pomiče da bi se omogućio parkiranje sljedećeg zrakoplova [22].

Prednosti korištenja zračnog mosta [22]:

- veća sigurnost,
- jednostavniji pristup zrakoplovu,
- potreban je manji broj ljudi,
- veća udobnost za putnike,
- manja potreba za korištenjem zemaljskih usluga.

Nedostaci korištenja zračnog mosta [22]:

- mogućnost oštećenja motora, trupa ili vrata zrakoplova,
- zračni most se može koristiti jedino za zrakoplove u blizini terminala,
- nije učinkovito ako zrakoplov ima ugrađene stepenice.



Slika 1. Zračni most [23]

3.2. Putničke stepenice

Putničke stepenice pripadaju opremi za ukrcaj/iskrcaj putnika te se koriste za siguran ulazak/izlazak putnika u/iz zrakoplova. Postavljanje putničkih stepenica na zrakoplov započinje kada se ugase *anticollision* svjetla te postave podmetači pod kotače zrakoplova. Kada se postavljaju putničke stepenice, iz sigurnosnih razloga je potrebno koristiti osobu za navođenje. Ta osoba se nalazi ispred stepenica i/ili vučnog vozila te upozorava na udaljenost do zrakoplova. Putničke stepenice se ne koriste ako zrakoplov ima ugrađene stepenice. Zračni prijevoznik određuje visinu na koju će se postaviti putničke stepenice, a stepenice se postavljaju tako da dotiču trup zrakoplova i moraju biti osigurane od nekontroliranog kretanja.

Postoje tri vrste stepenica [21]:

- Tipične, prevoze se pomoću vučnog vozila,
- Stepenice montirane na vozilo,
- Samohodne stepenice.

3.2.1. Vučene stepenice

Vučene stepenice (slika 2.) se moraju prevoziti do zrakoplova pomoću vučnog vozila jer nemaju vlastiti pogon. Vučno vozilo se zaustavlja na udaljenosti dva metra od zrakoplova, da bi se spriječila moguća oštećenja zrakoplova. Nakon što se vozilo zaustavi, putničke stepenice se ručno postavljaju na zrakoplov. Da bi se stepenice osigurale od nekontroliranog kretanja koriste se fiksatori.



Slika 2. Vučene stepenice [24]

3.2.2. Samohodne stepenice

Samohodne stepenice (slika 3.) posjeduju vlastiti motor te se samostalno kreću prema ili od zrakoplova. Provjera kočnica se obavlja na šest metara od zrakoplova te ponovno na dva metra od zrakoplova. Mogu se upotrijebiti jedino uz suglasnost zračnog prijevoznika. Samohodnim stepenicama upravlja vozač u kabini pomoću hidrosustava. Samohodne stepenice imaju sljedeće mogućnosti [21]:

- dizanje i spuštanje fiksatora stepenica,
- dizanje i spuštanje stepeništa,
- postavljanje točne visine stepeništa,
- uvlačenje i izvlačenje pokrova.



Slika 3. Samohodne stepenice [25]

3.3. Autobusi

Autobusi (slika 4.) služe za prijevoz putnika i njihove ručne prtljage od terminala do zrakoplova. Ovi autobusi su širi i duži od normalnih te nemaju sjedeća mjesta da bi mogli primiti što veći broj putnika. Autobusi se koriste [26]:

- kada ne postoji mogućnost korištenja zračnog mosta,
- kada je udaljenost između zrakoplova i terminala velika,
- radi sigurnosti putnika.

Autobusi za prijevoz putnika su se razvijali kroz povijest te su prošli ove faze [21]:

- prijevoz klasičnim autobusima za ulazak u zrakoplov sa strane autobusa,
- prijevoz autobusima za ulazak u zrakoplov s prednje strane autobusa,
- prijevoz autobusima na dva kata, koji imaju hidraulički podešavajući tunel, za izravan ulazak u zrakoplov,
- prijevoz autobusima koji imaju hidraulički sustav za podešavanje kabine za izravan ulazak u autobus s terminala i izravan ulazak u zrakoplov iz autobusa.



Slika 4. Autobus [27]

3.4. Liftmobil

Liftmobil (slika 5.) se koristi za ukrcaj/ iskrcaj putnika sa smanjenom pokretljivošću (eng. Passengers with reduced mobility - PRM). Zračne luke koje koriste zračni most, ne trebaju koristiti liftmobil za ukrcaj i iskrcaj PRM putnika. Osnovni dijelovi liftmobila su [21]:

- vozilo
- zadnja kabina
- prednja platforma
- hidraulički sustav
- dizalo
- sustav za stabiliziranje vozila.



Slika 5. Liftmobil [28]

3.5. Podmetači

Zrakoplov se nakon slijetanja parkira na određenu poziciju. Svi zrakoplovi posjeduju sustave kočenja koji ih održavaju na jednom mjestu. Usprkos tome, postoji rizik da se zrakoplov pod utjecajem vanjske sile (npr. vjetar) pomakne. To predstavlja veliki problem jer se zrakoplov radi veličine jako teško može zaustaviti. Da bi se to spriječilo pod kotače zrakoplova se postavljaju podmetači (slika 6.) da bi se ostvarila dodatna sigurnost. Postavljaju se nakon gašenja motora zrakoplova i *anticollision* svjetala, osim u slučajevima kada je zrakoplov potrebno spojiti na zemaljski izvor napajanja električnom energijom. Jedan podmetač se postavlja iza kotača, a drugi ispred. Zračni prijevoznik odlučuje pod koje kotače se postavljaju podmetači. Podmetači mogu biti različitih veličina te je za svaki zrakoplov potrebno odabrati odgovarajući podmetač, te ga postaviti na propisan način. Izrađuju se od različitih materijala, a najčešći materijali su guma, plastika ili aluminij [29].



Slika 6. Podmetači [30]

3.6. Agregati

Tijekom leta, zrakoplovu svu potrebnu električnu energiju omogućavaju motori zrakoplova. Kada se zrakoplov parkira na poziciju, prije nego što se motori isključe, u određenim slučajevima potrebno je zrakoplov priključiti na drugi izvor napajanja električnom energijom. Agregat za napajanje električnom energijom (engl. Ground power unit - GPU) (slika 7.) opskrbljuje zrakoplov električnom energijom dok se nalaze na zemlji. Električna energija koja se koristi u zrakoplovu je napona 115 V i frekvencije 400 Hz. Frekvencija od 400 Hz omogućava korištenje električnih uređaja manje veličine i težine.

Agregat se spaja na zrakoplov prije nego što se ugase motori, da zrakoplov u svakom trenutku bude priključen na izvor električne energije. Nakon što se postave podmetači pod kotače zrakoplova, operater može pristupiti zrakoplovu. Prije svakog spajanja na zrakoplov agregat se provjerava dok se priključak na svim zrakoplovima nalazi u blizini nosa zrakoplova. Kod uskotrupnih zrakoplova priključak se nalazi na otprilike 2 m visine, te mu operater može pristupiti izravno bez dodatnih tehničkih sredstva. Širokotrupni zrakoplovi imaju dva priključka jer zahtijevaju veću količinu električne energije. Priključci se kod širokotrupnih zrakoplova nalaze na visini većoj od 2,5 m, te su operatoru potrebna dodatna tehnička sredstva prilikom priključivanja. Kada operater ispravno uključi kabel u priključak, prikaže se znak da je zrakoplov spreman za napajanje. Prije odlaska zrakoplova, kabel se isključi i ukloni da se zrakoplov može izgurati s pozicije [31].

Postoje tri vrste agregata s obzirom na mjesto postavljanja i mogućnost kretanja [31]:

- fiksni agregat
- agregat postavljen na zračni most
- mobilni agregat.

Fiksni agregat je postavljen na stajanku te može napajati samo zrakoplove koji su parkirani na toj stajanki. Na stajanki može biti postavljen jedan ili dva fiksna agregata. Broj ovisi o veličini zrakoplova za koji je stajanka namijenjena. Fiksni agregati se koriste na zračnim lukama s velikim prometom, a sama upotreba fiksnih agregata pojednostavljuje odvijanje aktivnosti na zračnoj luci te smanjuje vjerojatnost nesreće.

Druga vrsta agregata je postavljena s donje strane zračnog mosta. Ovaj agregat se kreće zajedno sa zračnim mostom te se može koristiti samo na jednoj poziciji. Prednost ovog agregata je što ne zauzima prostor na stajanki. Pozicija agregata značajno otežava održavanje i popravak agregata.

Mobilni agregati se nalaze na kolicima koje vuče traktor mogu napajati različite zrakoplove koji se nalaze na različitim stajankama. Kretanje mobilnih agregata povećava promet na stajanki i vjerojatnost nesreće. Zračne luke koriste mobilne agregate za: napajanje zrakoplova parkiranih na udaljenim pozicijama, zamjenu za neispravne fiksne agregate i za stajanke gdje je razina prometa niska [31].



Slika 7. GPU [32]

3.7. Catering vozilo

Catering vozilo (slika 8.) se koristi za dostavu hrane i pića na zrakoplov. Hrana se pohranjuje u kolica za hranu koja se nalaze na stražnjoj platformi. Stražnja platforma se podiže na razinu vrata zrakoplova, a prednja platforma omogućava osoblju sigurno preuzimanje i dostavljanje kolica s hranom. Kada se hrana pohrani u zrakoplov, prazna kolica za hranu se vraćaju u vozilo. Sustav stabilizatora onemogućava pomicanje vozila zbog utjecaja vjetra ili kretanja osoblja. Vozila posjeduju rashladne sustave koji održavaju hranu na odgovarajućoj temperaturi. To je velika prednost za velike zračne luke, gdje catering vozilo prelazi veliki put do zrakoplova [33].

Catering vozila se mogu razlikovati prema mogućoj visini, nosivosti i veličini kontejnera. Ove karakteristike vozila ovise o veličini najvećeg zrakoplova kojeg mogu opsluživati.



Slika 8. Catering vozila [34]

3.8. Kolica za prijevoz prtljage

Kolica za prijevoz prtljage (slika 9.) su dizajnirana za prijevoz prtljage od zgrade zračne luke do zrakoplova. Mogu biti nosivosti od 1000 do 5000 kg, a zaštitna ograda štiti prtljagu od mogućih oštećenja. Traktor koji služi za vuču kolica je opremljen branikom da bi izbjegao oštećenja u dodiru s drugim površinama. Rudo za vuču posjeduje klip s oprugom koji osigurava kolica kada se kreću po oštećenim površinama [35].



Slika 9. Kolica za prijevoz prtljage[36]

3.9. Mobilna pokretna traka

Mobilna pokretna traka (slika 10.) predstavlja kamion koji na sebi ima instaliranu pokretnu traku kojom se provodi proces ukrcanja/iskrcanja prtljage u/iz zrakoplova. Traka se postavlja pod određenim nagibom koji ovisi o visini prtljažnog prostora zrakoplova. Pokretnu traku pokreće motor koji se može rotirati te tako omogućava ukrcavanje i iskrcavanje prtljage. Na vrhu opreme se nalazi branik koji sprečava oštećenja na trupu zrakoplova. Jedna ili dvije osobe ukrcavaju prtljagu na traku koja prevozi do vrha odnosno ulaza u prtljažni prostor zrakoplova [37]. Osnovni dijelovi mobilne pokretne trake [21]:

- vozilo
- pogonski motor
- hidraulički sustav
- most
- pokretna traka.



Slika 10. Mobilna pokretna traka [38]

3.10. ULD kolica

U zračnom prometu ULD (eng. Unit Load Device) predstavlja jedinično sredstvo utovara. ULD kolica (slika 11.) su posebno dizajnirana za prijevoz ULD – a do zrakoplova. Za vuču ULD kolica se koristi traktor dok su kolica opremljena sustavom valjaka koji štite ULD od oštećenja prilikom ukrcanja ili iskrcanja. Ugrađene kočnice na kolicima onemogućavaju padanje ULD – a s kolica, a platforma ULD kolica ima mogućnost rotiranja te se orijentacija ULD – a može prilagoditi prije ukrcanja na zrakoplov. Kada se kolica parkiraju, uključuje se automatska kočnica koja onemogućava pomicanje kotača. Osigurači za blokiranje kontejnera onemogućavaju pad ULD – a s kolica [37].



Slika 11. ULD kolica [39]

3.11. Utovarivač

Utovarivač (slika 12.) je uređaj dizajniran za ukrcaj ULD – a u teretni prostor zrakoplova. Sustav za ukrcaj ULD–a je postavljen na šasiju kamiona, a utovarivač posjeduje dvije platforme: prednju i stražnju. Prednja platforma se naziva most jer povezuje teretni prostor zrakoplova i stražnju platformu. Hidraulički škarasti sustav omogućuje prilagođavanje visine mosta prema visini teretnog prostora zrakoplova. ULD se iz ULD kolica ukrcava na stražnju platformu. Kao i prednja, stražnja platforma je postavljena na hidraulički škarasti sustav koji omogućuje podizanje i spuštanje platforme. Kada se teret ukrcava na stražnju platformu, prednja i stražnja platforma se podižu na visinu teretnog prostora zrakoplova. Stražnja platforma je opremljena sustavom valjaka koji automatski prenosi ULD od stražnje platforme preko mosta do teretnog prostora zrakoplova. Sustav valjaka omogućava pomicanje ULD – a naprijed, natrag te rotiranje bez ljudskog utjecaja. Utovarivači se proizvode s različitim kapacitetima ovisno o zrakoplovima kojima su namijenjeni [37].



Slika 12. Utovarivač [40]

3.12. Transporter

Transporter je sredstvo koje se koristi za prijevoz ULD – a ili tereta na paletama od terminala do zrakoplova i obratno te ima mogućnost ukrcaja i iskrcaja ULD – ova. Transporter pokreće vlastiti motor koji omogućava aktivnosti ukrcaja i iskrcaja ULD – ova. Hidraulički škarasti sustav, kao i kod utovarivača, prilagođava visinu platforme visini teretnog prostora zrakoplova. Osnovni dijelovi transportera [21]:

- hidraulički sustav
- pogonski motor
- platforma

- hidraulički škarasti sustav za podizanje platforme
- prostor za vozača.

3.13. Vučna vozila

Na zračnoj luci se najčešće koriste dva tipa vučnih vozila: traktori i vozila za izguravanje. Traktori se koriste za vuču opreme koja nema mogućnost samostalnog kretanja tj. nemaju vlastiti motor [21].

Vozila za izguravanje (slika 13.) se koriste za izguravanje zrakoplova s parkirne pozicije gdje ne mogu/ smiju izlaziti snagom vlastitih motora te za vuču zrakoplova do stajanke za održavanje, uzletno – sletne staze, stajanke za odleđivanje... Vozila su velike snage i težine, niskog dizajna te posjeduju velike kotače. Dizajn smanjuje mogućnost sudara sa zrakoplovom te omogućava veću stabilnost. Veliki kotači povećavaju vučnu snagu koja je neophodna za guranje zrakoplova. Proizvođači izrađuju različite tipove vozila za izguravanje. Mogu se razlikovati po kapacitetu ili karakteristikama, te njihova upotreba ovisi o tipu zrakoplova i uvjetima kojima su namijenjene [41]. Postoje dvije vrste vozila za izguravanje zrakoplova [21]:

- koja koriste rudo za vuču
- koja ne koriste rudo za vuču.

Kod vozila koja koriste rudo za vuču, rudo se jednim krajem spoji za vozilo, a drugim za zrakoplov. Vozilo gura zrakoplov natrag te ga usmjerava na voznu stazu. Kada se zrakoplov postavi u pravu poziciju, rudo se od spaja te se vozilo odmiče od zrakoplova.

Vozila koja ne koriste rudo za vuču, posjeduju hidraulički sustav koji podiže prednji kotač zrakoplova te ga podupire vlastitom šasijom. Na taj način gura zrakoplov do željene pozicije, gdje ga spušta te se zrakoplov nastavlja samostalno kretati. Vozila za izguravanje koja ne koriste rudo su lakša od onih koji koriste jer težina zrakoplova omogućava potrebnu vučnu snagu. Prednosti vozila koja ne koriste rudo [41]:

- manja potrošnja goriva,
- manji troškovi održavanja motora, šasije i mjenjača,
- ne postoje troškovi vezani za rude.



Slika 13. Vozilo za izguravanje [42]

3.14. Rudo za vuču

Rudo za vuču je sredstvo koje se koristi za povezivanje zrakoplova i vozila za izguravanje. To je čelična cilindrična šipka koja na oba kraja ima kuku. Jedna strana se priključuje na nos zrakoplova, a druga na vozilu za izguravanje. Osnovni dijelovi ruda [21]:

- rude
- kotači
- kuka za spajanje
- sustav za podizanje i spuštanje zrakoplova
- sjedište za adapter.

Vučna snaga vozila za izguravanje pokreće zrakoplov pomoću rude. Postoje različiti tipovi ruda koji se koriste za različite tipove zrakoplova. Zrakoplovi se razlikuju prema dizajnu nosa, te se pojedina ruda može koristiti samo za „obitelji“ zrakoplova koje imaju jednak dizajn nosa. Iz tog razloga zračne luke moraju imati veliki broj različitih tipova ruda za vuču zrakoplova [41].

3.15. Adapter

Adapter se koristi za spajanje rude za vuču na nos zrakoplova. Adapteri se razlikuju s obzirom na koji tip zrakoplova su namijenjeni. Ako se na pojedinoj rudi promjeni adapter ta ruda će se moći koristiti za spajanje na tip zrakoplov za koji je novi adapter namijenjen. Osnovni dijelovi adaptera [43]:

- adapter
- priključak za zrakoplov
- priključak za rude
- osigurač položaja.

3.16. Vozilo za servis otpadnih voda

Aktivnost servisa otpadnih voda zrakoplova je jako važna zbog broja putnika u zrakoplovu te trajanja samog leta. Ova aktivnost se obavlja kod tradicionalnih zračnih prijevoznika dok se kod niskotarifnih obavlja po potrebi. Servis otpadnih voda uključuje pražnjenje otpadnih voda zrakoplova te njihovu dezinfekciju. Vozilo za servis otpadnih voda sa integriranim pogonom (slika 14.) se najčešće koristi za ovu aktivnost jer se može koristiti na više zrakoplova te mu nije potrebno vučno vozilo. Vozilo predstavlja kamion na kojem se nalazi sva potrebna oprema za provođenje ove aktivnosti.

Vozilo na sebi ima dva spremnika. U jedan se isprazne otpadne vode zrakoplova, a drugom se nalazi sredstvo za dezinfekciju koje se koristi za ispiranje otpadne vode. Prvi spremnik ima sustav vakuuma koji ubrzava proces usisavanja otpadnih voda. Spremnik za ispiranje ima pumpu koja pumpa sredstvo za dezinfekciju u spremnik s otpadnom vodom.

Određena vozila u oba spremnika imaju postavljene grijače da se voda ne bi zaledila pri niskim temperaturama. Vozilo posjeduje platformu koja se može podići da bi se izvršio servis na većim zrakoplovima [44].



Slika 14. Vozilo za servis otpadnih voda [45]

3.17. Vozilo za opskrbu zrakoplova vodom

Aktivnost opskrbe zrakoplova pitkom vodom obavlja cisterna za opskrbu zrakoplova pitkom vodom (slika 15.). Zrakoplovi tradicionalnih zračnih prijevoznika se uvijek opskrbljuju pitkom vodom, dok se kod niskotarifnih ova aktivnost obavlja po potrebi. Osnovni dijelovi vozila za opskrbu zrakoplova pitkom vodom [21]:

- pogonski motor
- spremnik od nehrđajućeg čelika (1500 – 4000 l)
- crpka za vodu
- mjerač protoka vode – ručna crpka
- gumeno crijevo sa standardnim priključkom za zrakoplov
- stepenice promjenjive veličine koje se koriste za veće tipove zrakoplova.

Vozilo za opskrbu zrakoplova pitkom vodom mora biti parkirano na određenu poziciju u blizini zrakoplova prije nego li je zrakoplov sletio, te mu može prići tek kada se ugase *anticollision* svjetla. Za opskrbu zrakoplova se koristi crijevo, da bi se to crijevo priključilo mora se otvoriti odgovarajući panel na zrakoplovu. Da bi se započelo sa punjenjem mora se otvoriti odgovarajući ventil. Kada se zrakoplov napuni vodom potrebno je zatvoriti ventil, maknuti cijevi te udaljiti vozilo od zrakoplova.



Slika 15. Vozilo za opskrbu zrakoplova vodom [46]

3.18. Cisterna za opskrbu zrakoplova gorivom

Cisterna za opskrbu zrakoplova gorivom (slika 16.) je sredstvo koje zračne luke koriste za opskrbu zrakoplova gorivom. Dijelovi cisterne [21]:

- pogonski motor
- spremnik
- crpka za gorivo
- mjerač protoka goriva
- gumeno crijevo sa standardnim priključkom
- stepenice promjenjive veličine koje se koriste za veće tipove zrakoplova.

Gorivo se nalazi u spremniku koji može imati kapacitet od nekoliko tisuća litara. Crijevo i crpka se koriste za pumpanje goriva u zrakoplov. Mjerač protoka goriva mjeri količinu goriva sa kojom je zrakoplov opskrbljen, da bi se zračnom prijevozniku mogao izdati račun. Svaka cisterna posjeduje sigurnosne uređaje za slučaj opasnosti. Broj cisterna za opskrbu zrakoplova na pojedinoj zračnoj luci ovisi o njezinom prometu [47].



Slika 16. Cisterna za opskrbu zrakoplova gorivom [48]

3.19. Uređaj za klimatizaciju

Uređaj za klimatizaciju (slika 17.) se koristi za održavanje temperature u zrakoplovu, ako klima uređaj u zrakoplovu nije uključen. Ovi uređaji su jako slični klima uređajima koji se koriste u kućanstvima, samo imaju puno veći kapacitet. Opskrbljuju zrakoplov isključivo sa svježim zrakom, te taj zrak nije potrebno dodatno obraditi klima uređajem u zrakoplovu. U ljetnim mjesecima zrak se hladi, dok se u zimskim mjesecima grije prije opskrbe zrakoplova. Osnovni dijelovi su [21]:

- podvozje
- rudo za vuču
- pogonski motor
- generator
- ventilator
- izmjenjivač topline – rashladni sustav

- crijevo sa standardnim priključkom za zrakoplov.



Slika 17. Uređaj za klimatizaciju [49]

3.20. Viličar

Kada se pri utovaru ili istovaru tereta iz zrakoplova pojavi teret kojim nije moguće manipulirati uz pomoć transportera ili utovarivača, upotrebljava se viličar. Viličari se upotrebljavaju u skladištima tereta na zračnim lukama koje imaju veliki obujam teretnog prometa. Osnovni dijelovi [21]:

- podvozje
- pogonski motor
- kabina vozača
- most za podizanje vilica
- vilice.

3.21. Zračni starter

Zračni starter (slika 18.) je oprema koja se koristi za pokretanje motora kada se zrakoplov sprema za odlazak. Upotreba ovog uređaja je moguća neposredno prije aktivnosti izguravanja zrakoplova s pozicije. Zračni starter se prevozi pomoću vučnog vozila, a pilot mora dozvoliti priključivanje i korištenje ovog uređaja. Kada se pokrene zračni starter ventilator, kompresor i turbina se počinju okretati uz pomoć energije dobivene iz zračnog startera. Kada pilot uključi sustav ubrizgavanja goriva, zračni starter se isključi. Zračni starter se obično koristi za pokretanje samo jednog motora [50]. Dijelovi zračnog startera: [21]

- vozilo
- mlazni motor
- upravljačka ploča
- crijevo za protok zraka
- priključna glava.



Slika 18. Zračni starter [51]

3.22. Vozilo za odleđivanje i zaštitu zrakoplova od zaleđivanja

Na pojedinim zračnim lukama postoji mogućnost pojave jako niskih temperatura, koje uzrokuju stvaranje leda, mraza ili snijega na zrakoplovima. „De – icing“ je postupak uklanjanja leda, mraza ili snijega sa zrakoplova prije polijetanja. Ovaj postupak može obaviti zemaljsko osoblje ili zračni prijevoznik koristeći specijaliziranu opremu. Led, snijeg ili mraz se uklanja pomoću tekućine koja se nanosi po trupu zrakoplova. „Anti – icing“ je postupak kojim se sprečava nastanak leda, snijega ili mraza na trupu zrakoplova. Ovaj postupak osigurava da se nakon otapanja leda, snijega ili mraza, isti neće ponovno formirati određeno vrijeme. Isto kao i „de – icing“, ovaj postupak se obavlja pomoću tekućine koji se nanosi na trup zrakoplova. Vozilo za odleđivanje i zaštitu od zaleđivanja zrakoplova (slika 19.) je kamion koji na sebi svu potrebnu opremu te ima mogućnost vožnje po stajanci. Vozilo je opremljeno s hidrauličkom platformom koja omogućava osoblju da dosegne sve dijelove zrakoplova. Platforma može biti otvorena ili zatvorena. Zatvorena štiti osoblje od loših vremenskih uvjeta i plinova koje ispuštaju tekućine za „de – icing“ i „anti – icing“. Tekućina se nanosi na zrakoplov pomoću sustava za pumpanje koji instalirani na vozilu. Sustavi za pumpanje omogućavaju prskanje tekućine proizvoljnim pritiskom i protokom. Vozilo je opremljeno sigurnosnim sustavima da bi se spriječile moguće nesreće [52].



Slika 19. Vozilo za odleđivanje i zaštitu zrakoplova od zaleđivanja [53]

4. ANALIZA OPREME ZA PRIHVAT I OTPREMU ZRAKOPLOVA METODOM STABLA ODLUČIVANJA

Cilj ovog završnog rada je izraditi program koji će na temelju metode stabla odlučivanja analizirati opremu za prihvat i otpremu zrakoplova. Za izradu programa je korišten programski alat C#. Stablo odlučivanja je izrađeno na temelju CART algoritma koji izvodi podjelu podataka na temelju *gini* vrijednosti. Program je izrađen na temelju tablice koja se sastoji od 768 redova i 9 stupaca. Svaki red u tablici predstavlja jedan primjer, a stupci u tablici predstavljaju uvjete koji imaju više mogućih vrijednosti. Jedan primjer (tablica 1.) predstavlja skup vrijednosti svih stupaca. Svaki primjer se razlikuje od prethodnog za vrijednost jednog stupca. Ukupan broj primjera iznosi 768. Vrijednosti se mijenjaju sve dok u tablicu nisu uključene sve moguće kombinacije vrijednosti. Za svaki primjer se na temelju vrijednosti pojedinih uvjeta donosi odluka da li će se oprema koristiti ili ne. Uvjeti na temelju kojih se donosi odluka ovise o tipu opreme, tj. uvjeti na temelju kojih se donosi odluka nisu jednaki za utovarivač i catering vozilo. Uvjeti odnosno stupci te njihove moguće vrijednosti su sljedeće:

- tip zračnog prijevoznika (niskotarifni, tradicionalni i teretni)
- tip zrakoplova (uskotrupni i širokotrupni),
- duljina leta (kratkolinijski i dugolinijski),
- tip leta (tuzemni i inozemni),
- ugovorna obveza (ima ugovornu obvezu i nema ugovornu obvezu),
- razina kvalitete usluge zračne luke (viša razina kvalitete usluge i niža razina kvalitete usluge),
- da li zrakoplov ima ugrađene stepenice (ima ugrađene stepenice i nema ugrađene stepenice),
- da li zračna luka ima dopuštenje prijevoznika za korištenje samohodnih stepenica (ima dopuštenje i nema dopuštenje),
- model zrakoplova (A318, A319ceo, A319neo, A320ceo, A320neo, A321ceo, A321neo, A220 – 100, A220 – 300, B737 – 700, B737 – 800, B737 – 900, B737 MAX7, B737 MAX8, B737 MAX9, B737 MAX10, E170, E175, E190, E195, E175 – E2, E190 – E2, E195 – E2, Tu – 204, Tu – 214)

Tablica 1. Prikaz jednog primjera skupa vrijednosti svih uvjeta

| prijevoznik | tip zrakoplova | duljina leta | tip leta | ugovorna obveza | kvaliteta usluge | ugrađene stepenice | dopuštenje prijevoznika | model |
|--------------|----------------|----------------|----------|-----------------|------------------|--------------------|-------------------------|---------|
| niskotarifni | uskotrupni | kratkolinijski | tuzemni | ima | viša | ima | ima | A321neo |

Izvor: Autor

4.1. Klase i metode stabla odlučivanja

Program se sastoji od tri klase: *stabloOdlučivanja*, *podjelaPodataka* i *čvor*. Svaka klasa se sastoji od nekoliko atributa. Atributi klase *podjelaPodataka* su: stupac na temelju kojeg se obavlja podjela podataka, *gini* vrijednost stupca te liste različitih vrijednosti u stupcu. Da bi se dobile vrijednosti ovih atributa koriste se tri metode. Metoda *giniVrijednost* računa gini vrijednost za svaku različitu vrijednost u stupcu.

```
    }
    #references
    static double giniVrijednost(int[] polje, List<int> lista, int brojRazlicitihVrijednosti)
    {
        if (lista.Count == 0) return 0.0;
        int[] brojac = new int[brojRazlicitihVrijednosti];
        double[] broj = new double[brojRazlicitihVrijednosti];
        for (int i = 0; i < lista.Count; i++)
        {
            int idx = lista[i];
            int c = polje[idx];
            brojac[c]++;
        }
        for (int i = 0; i < brojRazlicitihVrijednosti; i++)
        {
            if (brojac[i] == 0) broj[i] = 0.0;
            else broj[i] = (brojac[i] * 1.0) / lista.Count;
        }
        double rezultat = 0.0;
        for (int i = 0; i < brojRazlicitihVrijednosti; i++)
        {
            rezultat += broj[i] * broj[i];
        }
        return 1.0 - rezultat;
    }
    #references
```

Slika 20. Metoda *giniVrijednost*

Izvor: Autor

Metoda *srednjaGiniVrijednost* je izrađena pomoću metode *giniVrijednost*. Ova metoda računa gini vrijednost stupca, na temelju prethodno izračunatih gini vrijednosti.

```
    #references
    static double srednjaGiniVrijednost(int[] polje, List<int> lista0, List<int> li:
    {
        if (lista0.Count == 0 && lista1.Count == 0 && lista2.Count == 0 && lista3.Count
        double gini0 = giniVrijednost(polje, lista0, brojRazlicitihVrijednosti);
        double gini1 = giniVrijednost(polje, lista1, brojRazlicitihVrijednosti);
        double gini2 = giniVrijednost(polje, lista2, brojRazlicitihVrijednosti);
        double gini3 = giniVrijednost(polje, lista3, brojRazlicitihVrijednosti);
        double gini4 = giniVrijednost(polje, lista4, brojRazlicitihVrijednosti);
        double gini5 = giniVrijednost(polje, lista5, brojRazlicitihVrijednosti);
        double gini6 = giniVrijednost(polje, lista6, brojRazlicitihVrijednosti);
        double gini7 = giniVrijednost(polje, lista7, brojRazlicitihVrijednosti);
        double gini8 = giniVrijednost(polje, lista8, brojRazlicitihVrijednosti);
        double gini9 = giniVrijednost(polje, lista9, brojRazlicitihVrijednosti);
        double gini10 = giniVrijednost(polje, lista10, brojRazlicitihVrijednosti);
        double gini11 = giniVrijednost(polje, lista11, brojRazlicitihVrijednosti);
        double gini12 = giniVrijednost(polje, lista12, brojRazlicitihVrijednosti);
        double gini13 = giniVrijednost(polje, lista13, brojRazlicitihVrijednosti);
        double gini14 = giniVrijednost(polje, lista14, brojRazlicitihVrijednosti);
        double gini15 = giniVrijednost(polje, lista15, brojRazlicitihVrijednosti);
        double gini16 = giniVrijednost(polje, lista16, brojRazlicitihVrijednosti);
        double gini17 = giniVrijednost(polje, lista17, brojRazlicitihVrijednosti);
        double gini18 = giniVrijednost(polje, lista18, brojRazlicitihVrijednosti);
        double gini19 = giniVrijednost(polje, lista19, brojRazlicitihVrijednosti);
        double gini20 = giniVrijednost(polje, lista20, brojRazlicitihVrijednosti);
        double gini21 = giniVrijednost(polje, lista21, brojRazlicitihVrijednosti);
        double gini22 = giniVrijednost(polje, lista22, brojRazlicitihVrijednosti);
        double gini23 = giniVrijednost(polje, lista23, brojRazlicitihVrijednosti);
        double gini24 = giniVrijednost(polje, lista24, brojRazlicitihVrijednosti);

        int brojac0 = lista0.Count;
        int brojac1 = lista1.Count;
        int brojac2 = lista2.Count;
        int brojac3 = lista3.Count;
        int brojac4 = lista4.Count;
        int brojac5 = lista5.Count;
        int brojac6 = lista6.Count;
        int brojac7 = lista7.Count;
        int brojac8 = lista8.Count;
        int brojac9 = lista9.Count;
        int brojac10 = lista10.Count;
        int brojac11 = lista11.Count;
        int brojac12 = lista12.Count;
        int brojac13 = lista13.Count;
        int brojac14 = lista14.Count;
        int brojac15 = lista15.Count;
        int brojac16 = lista16.Count;
        int brojac17 = lista17.Count;

        int brojac6 = lista6.Count;
        int brojac7 = lista7.Count;
        int brojac8 = lista8.Count;
        int brojac9 = lista9.Count;
        int brojac10 = lista10.Count;
        int brojac11 = lista11.Count;
        int brojac12 = lista12.Count;
        int brojac13 = lista13.Count;
        int brojac14 = lista14.Count;
        int brojac15 = lista15.Count;
        int brojac16 = lista16.Count;
        int brojac17 = lista17.Count;
        int brojac18 = lista18.Count;
        int brojac19 = lista19.Count;
        int brojac20 = lista20.Count;
        int brojac21 = lista21.Count;
        int brojac22 = lista22.Count;
        int brojac23 = lista23.Count;
        int brojac24 = lista24.Count;

        double wt0 = (brojac0 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt1 = (brojac1 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt2 = (brojac2 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt3 = (brojac3 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt4 = (brojac4 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt5 = (brojac5 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt6 = (brojac6 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt7 = (brojac7 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt8 = (brojac8 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt9 = (brojac9 * 1.0) / (brojac0 + brojac1 + brojac2 +
        double wt10 = (brojac10 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt11 = (brojac11 * 1.0) / (brojac0 + brojac1 + brojac2
        double wt12 = (brojac12 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt13 = (brojac13 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt14 = (brojac14 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt15 = (brojac15 * 1.0) / (brojac0 + brojac1 + brojac2
        double wt16 = (brojac16 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt17 = (brojac17 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt18 = (brojac18 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt19 = (brojac19 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt20 = (brojac20 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt21 = (brojac21 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt22 = (brojac22 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt23 = (brojac23 * 1.0) / (brojac0 + brojac1 + brojac:
        double wt24 = (brojac24 * 1.0) / (brojac0 + brojac1 + brojac:

        double rezultat = (wt0 * gini0) + (wt1 * gini1) + (wt2 * gin:
        return rezultat;
    }
    #references
```

Slika 21. Metoda *srednjaGiniVrijednost*

Izvor: Autor

Ovaj postupak je potrebno ponavljati za svaki stupac, tj. potrebno je izračunati gini vrijednost za sve stupce. Taj postupak je napravljen u metodi *podjeliPodatke*. Koristeći metodu *srednjaGiniVrijednost*, izračunaju se gini vrijednosti za sve stupce. Te vrijednosti se međusobno uspoređuju i odabire se stupac s najmanjom gini vrijednosti za podjelu podataka.

```

1 reference
2 static podjelaPodataka podjeliPodatke(int[,] matrica, int[]
3 {
4     podjelaPodataka pp = new podjelaPodataka();
5     int brojStupaca = matrica.GetLength(1);
6     int najboljiStupac = 0;
7     double najmanjaSrednjaGiniVrijednost = double.MaxValue;
8     List<int> najboljaList0 = new List<int>();
9     List<int> najboljaList1 = new List<int>();
10    List<int> najboljaList2 = new List<int>();
11    List<int> najboljaList3 = new List<int>();
12    List<int> najboljaList4 = new List<int>();
13    List<int> najboljaList5 = new List<int>();
14    List<int> najboljaList6 = new List<int>();
15    List<int> najboljaList7 = new List<int>();
16    List<int> najboljaList8 = new List<int>();
17    List<int> najboljaList9 = new List<int>();
18    List<int> najboljaList10 = new List<int>();
19    List<int> najboljaList11 = new List<int>();
20    List<int> najboljaList12 = new List<int>();
21    List<int> najboljaList13 = new List<int>();
22    List<int> najboljaList14 = new List<int>();
23    List<int> najboljaList15 = new List<int>();
24    List<int> najboljaList16 = new List<int>();
25    List<int> najboljaList17 = new List<int>();
26    List<int> najboljaList18 = new List<int>();
27    List<int> najboljaList19 = new List<int>();
28    List<int> najboljaList20 = new List<int>();
29    List<int> najboljaList21 = new List<int>();
30    List<int> najboljaList22 = new List<int>();
31    List<int> najboljaList23 = new List<int>();
32    List<int> najboljaList24 = new List<int>();
33
34    for (int j = 0; j < brojStupaca; j++)
35    {
36        List<int> list0 = new List<int>();
37        List<int> list1 = new List<int>();
38        List<int> list2 = new List<int>();
39        List<int> list3 = new List<int>();
40        List<int> list4 = new List<int>();
41        List<int> list5 = new List<int>();
42        List<int> list6 = new List<int>();
43        List<int> list7 = new List<int>();
44        List<int> list8 = new List<int>();
45        List<int> list9 = new List<int>();
46        List<int> list10 = new List<int>();
47        List<int> list11 = new List<int>();
48        List<int> list12 = new List<int>();
49        List<int> list13 = new List<int>();
50        List<int> list14 = new List<int>();
51        List<int> list15 = new List<int>();
52        List<int> list16 = new List<int>();
53        List<int> list17 = new List<int>();
54        List<int> list18 = new List<int>();
55        List<int> list19 = new List<int>();
56        List<int> list20 = new List<int>();
57        List<int> list21 = new List<int>();
58        List<int> list22 = new List<int>();
59        List<int> list23 = new List<int>();
60        List<int> list24 = new List<int>();
61
62        foreach (int i in lista)
63        {
64            if (matrica[i, j] == 0)
65                list0.Add(i);
66            if (matrica[i, j] == 1)
67                list1.Add(i);
68            if (matrica[i, j] == 2)
69                list2.Add(i);
70            if (matrica[i, j] == 3)
71                list3.Add(i);
72            if (matrica[i, j] == 4)
73                list4.Add(i);
74            if (matrica[i, j] == 5)
75                list5.Add(i);
76            if (matrica[i, j] == 6)
77                list6.Add(i);
78            if (matrica[i, j] == 7)
79                list7.Add(i);
80            if (matrica[i, j] == 8)
81                list8.Add(i);
82            if (matrica[i, j] == 9)
83                list9.Add(i);
84            if (matrica[i, j] == 10)
85                list10.Add(i);
86            if (matrica[i, j] == 11)
87                list11.Add(i);
88            if (matrica[i, j] == 12)
89                list12.Add(i);
90            if (matrica[i, j] == 13)
91                list13.Add(i);
92            if (matrica[i, j] == 14)
93                list14.Add(i);
94            if (matrica[i, j] == 15)
95                list15.Add(i);
96            if (matrica[i, j] == 16)
97                list16.Add(i);
98            if (matrica[i, j] == 17)
99                list17.Add(i);
100           if (matrica[i, j] == 18)
101               list18.Add(i);
102           if (matrica[i, j] == 19)
103               list19.Add(i);
104           if (matrica[i, j] == 20)
105               list20.Add(i);
106           if (matrica[i, j] == 21)
107               list21.Add(i);
108           if (matrica[i, j] == 22)
109               list22.Add(i);
110           if (matrica[i, j] == 23)
111               list23.Add(i);
112           if (matrica[i, j] == 24)
113               list24.Add(i);
114
115           najboljaList0 = new List<int>(list0);
116           najboljaList1 = new List<int>(list1);
117           najboljaList2 = new List<int>(list2);
118           najboljaList3 = new List<int>(list3);
119           najboljaList4 = new List<int>(list4);
120           najboljaList5 = new List<int>(list5);
121           najboljaList6 = new List<int>(list6);
122           najboljaList7 = new List<int>(list7);
123           najboljaList8 = new List<int>(list8);
124           najboljaList9 = new List<int>(list9);
125           najboljaList10 = new List<int>(list10);
126           najboljaList11 = new List<int>(list11);
127           najboljaList12 = new List<int>(list12);
128           najboljaList13 = new List<int>(list13);
129           najboljaList14 = new List<int>(list14);
130           najboljaList15 = new List<int>(list15);
131           najboljaList16 = new List<int>(list16);
132           najboljaList17 = new List<int>(list17);
133           najboljaList18 = new List<int>(list18);
134           najboljaList19 = new List<int>(list19);
135           najboljaList20 = new List<int>(list20);
136           najboljaList21 = new List<int>(list21);
137           najboljaList22 = new List<int>(list22);
138           najboljaList23 = new List<int>(list23);
139           najboljaList24 = new List<int>(list24);
140
141           double srednjaGiniVrijednost = srednjaGiniVrijednost;
142           if (srednjaGiniVrijednost < najmanjaSrednjaGiniVrijednost)
143           {
144               najmanjaSrednjaGiniVrijednost = srednjaGiniVrijednost;
145               najboljiStupac = j;
146           }
147
148           return pp;
149       }
150   }
151
152   pp.stupac = najboljiStupac;
153   pp.srednjaGiniVrijednost = najmanjaSrednjaGiniVrijednost;
154   pp.list0 = new List<int>(najboljaList0);
155   pp.list1 = new List<int>(najboljaList1);
156   pp.list2 = new List<int>(najboljaList2);
157   pp.list3 = new List<int>(najboljaList3);
158   pp.list4 = new List<int>(najboljaList4);
159   pp.list5 = new List<int>(najboljaList5);
160   pp.list6 = new List<int>(najboljaList6);
161   pp.list7 = new List<int>(najboljaList7);
162   pp.list8 = new List<int>(najboljaList8);
163   pp.list9 = new List<int>(najboljaList9);
164   pp.list10 = new List<int>(najboljaList10);
165   pp.list11 = new List<int>(najboljaList11);
166   pp.list12 = new List<int>(najboljaList12);
167   pp.list13 = new List<int>(najboljaList13);
168   pp.list14 = new List<int>(najboljaList14);
169   pp.list15 = new List<int>(najboljaList15);
170   pp.list16 = new List<int>(najboljaList16);
171   pp.list17 = new List<int>(najboljaList17);
172   pp.list18 = new List<int>(najboljaList18);
173   pp.list19 = new List<int>(najboljaList19);
174   pp.list20 = new List<int>(najboljaList20);
175   pp.list21 = new List<int>(najboljaList21);
176   pp.list22 = new List<int>(najboljaList22);
177   pp.list23 = new List<int>(najboljaList23);
178   pp.list24 = new List<int>(najboljaList24);
179
180   return pp;
181 }

```

Slika 22. Metoda *podjeliPodatke*

Izvor: Autor

```

1     list15.Add(i);
2     if (matrica[i, j] == 16)
3         list16.Add(i);
4     if (matrica[i, j] == 17)
5         list17.Add(i);
6     if (matrica[i, j] == 18)
7         list18.Add(i);
8     if (matrica[i, j] == 19)
9         list19.Add(i);
10    if (matrica[i, j] == 20)
11        list20.Add(i);
12    if (matrica[i, j] == 21)
13        list21.Add(i);
14    if (matrica[i, j] == 22)
15        list22.Add(i);
16    if (matrica[i, j] == 23)
17        list23.Add(i);
18    if (matrica[i, j] == 24)
19        list24.Add(i);
20
21    }
22
23    double srednja_gini_vrijednost = srednjaGiniVrijednost;
24    if (srednja_gini_vrijednost < najmanjaSrednjaGiniVrijednost)
25    {
26        najmanjaSrednjaGiniVrijednost = srednja_gini_vrijednost;
27        najboljiStupac = j;
28    }
29
30    najboljaList0 = new List<int>(list0);
31    najboljaList1 = new List<int>(list1);
32    najboljaList2 = new List<int>(list2);
33    najboljaList3 = new List<int>(list3);
34    najboljaList4 = new List<int>(list4);
35    najboljaList5 = new List<int>(list5);
36    najboljaList6 = new List<int>(list6);
37    najboljaList7 = new List<int>(list7);
38    najboljaList8 = new List<int>(list8);
39    najboljaList9 = new List<int>(list9);
40    najboljaList10 = new List<int>(list10);
41    najboljaList11 = new List<int>(list11);
42    najboljaList12 = new List<int>(list12);
43    najboljaList13 = new List<int>(list13);
44    najboljaList14 = new List<int>(list14);
45    najboljaList15 = new List<int>(list15);
46    najboljaList16 = new List<int>(list16);
47    najboljaList17 = new List<int>(list17);
48    najboljaList18 = new List<int>(list18);
49    najboljaList19 = new List<int>(list19);
50    najboljaList20 = new List<int>(list20);
51    najboljaList21 = new List<int>(list21);
52    najboljaList22 = new List<int>(list22);
53    najboljaList23 = new List<int>(list23);
54    najboljaList24 = new List<int>(list24);
55
56    }
57
58    pp.stupac = najboljiStupac;
59    pp.srednjaGiniVrijednost = najmanjaSrednjaGiniVrijednost;
60    pp.list0 = new List<int>(najboljaList0);
61    pp.list1 = new List<int>(najboljaList1);
62    pp.list2 = new List<int>(najboljaList2);
63    pp.list3 = new List<int>(najboljaList3);
64    pp.list4 = new List<int>(najboljaList4);
65    pp.list5 = new List<int>(najboljaList5);
66    pp.list6 = new List<int>(najboljaList6);
67    pp.list7 = new List<int>(najboljaList7);
68    pp.list8 = new List<int>(najboljaList8);
69    pp.list9 = new List<int>(najboljaList9);
70    pp.list10 = new List<int>(najboljaList10);
71    pp.list11 = new List<int>(najboljaList11);
72    pp.list12 = new List<int>(najboljaList12);
73    pp.list13 = new List<int>(najboljaList13);
74    pp.list14 = new List<int>(najboljaList14);
75    pp.list15 = new List<int>(najboljaList15);
76    pp.list16 = new List<int>(najboljaList16);
77    pp.list17 = new List<int>(najboljaList17);
78    pp.list18 = new List<int>(najboljaList18);
79    pp.list19 = new List<int>(najboljaList19);
80    pp.list20 = new List<int>(najboljaList20);
81    pp.list21 = new List<int>(najboljaList21);
82    pp.list22 = new List<int>(najboljaList22);
83    pp.list23 = new List<int>(najboljaList23);
84    pp.list24 = new List<int>(najboljaList24);
85
86    return pp;
87 }

```

Slika 23. Metoda *podjeliPodatke*

Izvor: Autor

Atributi klase *čvor* su: stupac na temelju kojeg se obavlja podjela podataka, *gini* vrijednost stupca, liste različitih vrijednosti u stupcu, redni broj čvora te broj primjera za svaki konačni ishod (da li se oprema koristi ili ne koristi). Prva tri atributa su ista kao i kod klase *podjela podataka* te se računaju pomoću istih metoda. Metoda *brojIshoda* računa za koliko primjera se pojedina oprema koristi ili ne koristi.

```

1282     pp.list5 = new List<int>(najboljaList5);
1283     pp.list6 = new List<int>(najboljaList6);
1284     pp.list7 = new List<int>(najboljaList7);
1285     return pp;
1286 }
1287
1288 1 reference
1289 static int[] brojIshoda(int[] polje, List<int> lista, int brojRazlicitihVrijednosti)
1290 {
1291     int[] rezultat = new int[brojRazlicitihVrijednosti];
1292     foreach (int i in lista)
1293     {
1294         int c = polje[i];
1295         rezultat[c]++;
1296     }
1297     return rezultat;
1298 }
1299
1300 references

```

Slika 24. Metoda brojIshoda

Izvor: Autor

Atributi klase *stabloOdlučivanja* su: broj konačnih rezultata, broj čvorova u stablu te lista čvorova. Metoda *napraviStablo* oblikuje stablo odlučivanja na način da se u njoj provode sve prethodno opisane metode. Metode se provode posebno za svaki čvor u stablu odlučivanja.

```

}
1 reference
public void napraviStablo(int[,] matrica, int[] polje)
{
    int n = matrica.GetLength(0);
    List<int> sviRedovi = new List<int>();
    for (int i = 0; i < n; i++)
    {
        sviRedovi.Add(i);
    }
    this.stablo[0].redovi = new List<int>(sviRedovi);
    for (int i = 0; i < this.brojČvorova; i++)
    {
        this.stablo[i].IDčvor = i;
        podjelaPodataka pp = podjelaPodatke(matrica, polje, this.stablo[i].redovi, this.brojHogučihIshoda);
        stablo[i].atribut = pp.atribut;
        stablo[i].srednjaVrijednost = pp.srednjaVrijednost;
        stablo[i].ishod = brojIshoda(polje, this.stablo[i].redovi, this.brojHogučihIshoda);
        int prvi = (25 * i) + 1;
        int drugi = (25 * i) + 2;
        int treci = (25 * i) + 3;
        int cetvrti = (25 * i) + 4;
        int peti = (25 * i) + 5;
        int sestii = (25 * i) + 6;
        int sedmi = (25 * i) + 7;
        int osmi = (25 * i) + 8;
        int deveti = (25 * i) + 9;
        int deseti = (25 * i) + 10;
        int jedanaesti = (25 * i) + 11;
        int dvanaesti = (25 * i) + 12;
        int trinaesti = (25 * i) + 13;
        int četrnaesti = (25 * i) + 14;
        int petnaesti = (25 * i) + 15;
        int šestnaesti = (25 * i) + 16;
        int sedamnaesti = (25 * i) + 17;
        int osamnaesti = (25 * i) + 18;
        int devetnaesti = (25 * i) + 19;
        int dvadeseti = (25 * i) + 20;
        int dvadeset_prvi = (25 * i) + 21;
        int dvadeset_drugi = (25 * i) + 22;
        int dvadeset_treci = (25 * i) + 23;
        int dvadeset_cetvrti = (25 * i) + 24;
        int dvadeset_peti = (25 * i) + 25;
        if (prvi < brojČvorova)
        {
            stablo[prvi].redovi = new List<int>(pp.list0);
        }
        if (drugi < brojČvorova)
        {
            stablo[drugi].redovi = new List<int>(pp.list1);
        }
        if (treci < brojČvorova)
        {
            stablo[treci].redovi = new List<int>(pp.list2);
        }
        if (cetvrti < brojČvorova)
        {
            stablo[cetvrti].redovi = new List<int>(pp.list3);
        }
        if (peti < brojČvorova)
        {
            stablo[peti].redovi = new List<int>(pp.list4);
        }
        if (sesti < brojČvorova)
        {
            stablo[sesti].redovi = new List<int>(pp.list5);
        }
        if (sedmi < brojČvorova)
        {
            stablo[sedmi].redovi = new List<int>(pp.list6);
        }
        if (osmi < brojČvorova)
        {
            stablo[osmi].redovi = new List<int>(pp.list7);
        }
        if (deveti < brojČvorova)
        {
            stablo[deveti].redovi = new List<int>(pp.list8);
        }
        if (deseti < brojČvorova)
        {
            stablo[deseti].redovi = new List<int>(pp.list9);
        }
        if (jedanaesti < brojČvorova)
        {
            stablo[jedanaesti].redovi = new List<int>(pp.list10);
        }
        if (dvanaesti < brojČvorova)
        {
            stablo[dvanaesti].redovi = new List<int>(pp.list11);
        }
        if (trinaesti < brojČvorova)
        {
            stablo[trinaesti].redovi = new List<int>(pp.list12);
        }
        if (četrnaesti < brojČvorova)
        {
            stablo[četrnaesti].redovi = new List<int>(pp.list13);
        }
    }
}

```

Slika 25. Metoda napraviStablo

Izvor: Autor

```

if (četnaesti < brojčvorova)
{
    stablo[četnaesti].redovi = new List<int>(pp.list13);
}
if (petnaesti < brojčvorova)
{
    stablo[petnaesti].redovi = new List<int>(pp.list14);
}
if (šesnaesti < brojčvorova)
{
    stablo[šesnaesti].redovi = new List<int>(pp.list15);
}
if (sedamnaesti < brojčvorova)
{
    stablo[sedamnaesti].redovi = new List<int>(pp.list16);
}
if (osamnaesti < brojčvorova)
{
    stablo[osamnaesti].redovi = new List<int>(pp.list17);
}
if (devetnaesti < brojčvorova)
{
    stablo[devetnaesti].redovi = new List<int>(pp.list18);
}
if (dvadeseti < brojčvorova)
{
    stablo[dvadeseti].redovi = new List<int>(pp.list19);
}
if (dvadeset_prvi < brojčvorova)
{
    stablo[dvadeset_prvi].redovi = new List<int>(pp.list20);
}
if (dvadeset_drugi < brojčvorova)
{
    stablo[dvadeset_drugi].redovi = new List<int>(pp.list21);
}
if (dvadeset_treci < brojčvorova)
{
    stablo[dvadeset_treci].redovi = new List<int>(pp.list22);
}
if (dvadeset_cetvrti < brojčvorova)
{
    stablo[dvadeset_cetvrti].redovi = new List<int>(pp.list23);
}
if (dvadeset_peti < brojčvorova)
{
    stablo[dvadeset_peti].redovi = new List<int>(pp.list24);
}
}

```

Slika 26. Metoda *napraviStablo*

Izvor: Autor

Metode *pokažiČvor* i *pokaži* se koriste za prikaz pojedinog čvora ili cjelokupnog stabla odlučivanja. Ove metode se koriste i za obrezivanje stabla. Na ekranu se neće prikazati grane koje se ponavljaju te grane koje nemaju važnost za cjelokupno stablo odlučivanja.

4.2. Rezultati primjene stabla odlučivanja u analizi opreme za prihvat i otpremu zrakoplova

Stablo odlučivanja se ne može primijeniti za svaki tip opreme. Za pojedinu opremu konačna stabla odlučivanja su previše jednostavna, tj. imaju mali broj grana. U te primjere spadaju: zračni most, viličar, cisterna za opskrbu gorivom, podmetači, agregat, zračni starter, kolica za prijevoz prtljage i druga oprema. U ovom radu stablo odlučivanja je izrađeno za samohodne stepenice, vučene stepenice, catering vozilo, cisternu za opskrbu vodom i utovarivač.

4.2.1. Stablo odlučivanja za samohodne stepenice

Prvo stablo odlučivanja će analizirati upotrebu samohodnih stepenica. Ukupan broj primjera u kojima se koriste samohodne stepenice iznosi 96, dok je ukupan broj primjera u kojima se ne koriste samohodne stepenice iznosi 672. Za samohodne stepenice čvor 0 je korijenski čvor te se grana na temelju stupca: razina kvalitete usluge zračne luke. Gini vrijednost čvora je 0,1875. Čvor 1 predstavlja zračne luke niže razine usluge te uključuje 384 primjera. Broj primjera niže razine usluge u kojima se samohodne stepenice ne koriste iznosi 288, dok je broj primjera u kojima se koriste 96. Ovaj čvor se grana prema stupcu: da li zrakoplov ima ugrađene stepenice. Gini vrijednost ovog čvora iznosi 0,25. Čvor 2 je lisni čvor te predstavlja zračne luke više razine kvalitete usluge. Uključuje 384 primjera. Zračne luke više

razine usluge posjeduju pozicije s izguravanjem, te ne koriste samohodne stepenice, već koriste zračni most. Samohodne stepenice ne koriste u svih 384 primjera. Čvor 26 predstavlja zrakoplove koji nemaju ugrađene stepenice te uključuje 192 primjera. Broj primjera u kojima se samohodne stepenice koriste iznosi 96, dok je broj primjera u kojima se ne koriste 96. Ovaj čvor se dalje grana prema stupcu: da li zračna luka ima dopuštenje zračnog prijevoznika za upotrebu samohodnih stepenica. Gini vrijednost ovog čvora iznosi 0. Čvor 27 je lisni čvor te predstavlja zrakoplove koji imaju ugrađene stepenice. Uključuje 192 primjera. Samohodne stepenice se ne koriste u sva 192 primjera. Gini vrijednost ovog čvora iznosi 0. Čvor 651 je lisni čvor, te predstavlja zračne luke koje nemaju dopuštenje zračnih prijevoznika za korištenje samohodnih stepenice. Uključuje 96 primjera. Samohodne stepenice se ne koriste u svih 96 primjera. Gini vrijednost čvora je 0. Čvor 652 je lisni čvor, te predstavlja zračne luke koje imaju dopuštenje zračnog prijevoznika za korištenje samohodnih stepenica. Uključuje 96 primjera. Samohodne stepenice se koriste u svih 96 primjera. Gini vrijednost čvora je 0.

```

C:\Users\Drobac\OneDrive\Radna površina\za
čvor:0
ne koristi se:672 koristi se:96
kvaliteta usluge
gini vrijednost: 0,1875
čvor:1
niža razina kvalitete
ne koristi se:288 koristi se:96
ugrađene stepenice
gini vrijednost: 0,25
čvor:2
viša razina kvalitete
ne koristi se:384 koristi se:0
gini vrijednost: 0
čvor:26
nema ugrađene stepenice
ne koristi se:96 koristi se:96
dopuštenje zračnog prijevoznika
gini vrijednost: 0
čvor:27
ima ugrađene stepenice
ne koristi se:192 koristi se:0
gini vrijednost: 0
čvor:651
nema dopuštenje zračnog prijevoznika
ne koristi se:96 koristi se:0
gini vrijednost: 0
čvor:652
ima dopuštenje zračnog prijevoznika
ne koristi se:0 koristi se:96
gini vrijednost: 0

```

Slika 27. Stablo odlučivanja za samohodne stepenice

Izvor: Autor

4.2.2. Stablo odlučivanja za vučene stepenice

Sljedeći stablo odlučivanja analizira upotrebu vučenih stepenice. Ukupan broj primjera za koji se vučene stepenice koriste iznosi 96, a broj primjera za koji se ne koriste iznosi 672. Čvor 0 je korijenski čvor, te se grana prema stupcu: razina kvalitete usluge zračne luke. Gini vrijednost čvora je 0,1875. Čvor 1 predstavlja zračne luke niže razine kvalitete usluge. Uključuje 384 primjera. Broj primjera za koje se vučene stepenice koriste iznosi 96, a za koje se ne koriste iznosi 288. Ovaj čvor se grana prema stupcu: da li zrakoplov ima ugrađene stepenice. Gini vrijednost čvora iznosi 0,25. Čvor 2 je lisni čvor te predstavlja zračne luke više razine kvalitete usluge. Uključuje 384 primjera. Zračne luke više razine kvalitete usluge posjeduju pozicije s izguravanjem te ne koriste vučene stepenice već zračni most. Vučene stepenice se ne koriste u sva 384 primjera. Vrijednost ovog čvora je 0. Čvor 26 predstavlja

zrakoplove koji nemaju ugrađene stepenice te uključuje 192 primjera. Broj primjera u kojima se vučene stepenice koriste iznosi 96, a broj primjera za koji se ne koriste iznosi 96. Ovaj čvor se grana prema stupcu: da li zračna luka ima dopuštenje zračnog prijevoznika za korištenje samohodnih stepenica. Gini vrijednost čvora je 0. Čvor 27 je lisni čvor te predstavlja zrakoplove koji imaju ugrađene stepenice. Uključuje 192 primjera. Vučene stepenice se ne koriste u sva 192 primjera. Gini vrijednost čvora iznosi 0. Čvor 651 je lisni čvor te predstavlja zračne luke koje nemaju dopuštenje zračnog prijevoznika za korištenje samohodnih stepenica. Uključuje 96 primjera. Vučene stepenice se koriste u svih 96 primjera. Gini vrijednost čvora iznosi 0. Čvor 652 je lisni čvor te predstavlja zračne luke koje imaju dopuštenje prijevoznika za korištenje samohodnih stepenica. Uključuje 96 primjera. Vučene stepenice se ne koriste u svih 96 primjera. Gini vrijednost čvora iznosi 0.

```

C:\Users\Drobac\OneDrive\Radna površina\zav
čvor:0
ne koristi se:672   koristi se:96
kvaliteta usluge
gini vrijednost: 0,1875

čvor:1
niža razina kvalitete
ne koristi se:288   koristi se:96
ugrađene stepenice
gini vrijednost: 0,25

čvor:2
viša razina kvalitete
ne koristi se:384   koristi se:0
gini vrijednost: 0

čvor:26
nema ugrađene stepenice
ne koristi se:96   koristi se:96
dopuštenje zračnog prijevoznika
gini vrijednost: 0

čvor:27
ima ugrađene stepenice
ne koristi se:192   koristi se:0
gini vrijednost: 0

čvor:651
nema dopuštenje zračnog prijevoznika
ne koristi se:0   koristi se:96
gini vrijednost: 0

čvor:652
ima dopuštenje zračnog prijevoznika
ne koristi se:96   koristi se:0
gini vrijednost: 0

```

Slika 28. Stablo odlučivanja za vučene stepenice

Izvor: Autor

4.2.3. Stablo odlučivanja za catering vozilo

Sljedeće stablo odlučivanja analizira upotrebu catering vozila. Ukupan broj primjera za koje se catering vozilo koristi iznosi 352 primjera, dok se ne koristi u 416 primjera. Čvor 0 je korijenski čvor te se grana prema stupcu: tip zračnog prijevoznika. Gini vrijednost prvog čvora je 0,2815. Čvor 1 je lisni čvor, te predstavlja teretne zračne prijevoznike. Uključuje 256 primjera. Catering vozilo se ne koristi za svih 256 primjera. Gini vrijednost čvora iznosi 0. Čvor 2 predstavlja tradicionalne zračne prijevoznike te uključuje 256 primjera. Broj primjera za koje se catering vozilo koristi iznosi 192, a za koje se ne koristi iznosi 64. Ovaj čvor se dalje grana prema stupcu: duljina leta. Gini vrijednost čvora iznosi 0,25. Čvor 3 predstavlja niskotarifne zračne prijevoznike te uključuje 256 primjera. Broj primjera u kojima se koristi catering vozilo

iznosi 160, a u kojima se ne koriste iznosi 96. Ovaj čvor se dalje grana prema stupcu: duljina leta. Gini vrijednost ovog čvora iznosi 0,1875. Čvor 51 predstavlja kratkolinijski let tradicionalnog prijevoznika te uključuje 128 primjera. Broj primjera u kojima se catering vozilo koristi iznosi 64, a u kojima se ne koristi iznosi 64. Čvor se dalje grana prema stupcu: tip leta. Gini vrijednost čvora iznosi 0. Čvor 52 je lisni čvor te predstavlja dugolinijski let tradicionalnog prijevoznika. Uključuje 128 primjera. Catering vozilo se koristi u svih 128 primjera. Gini vrijednost čvora je 0.

Čvor 76 predstavlja kratkolinijski let niskotarifnog prijevoznika te uključuje 128 primjera. Broj primjera u kojima se koristi catering vozilo iznosi 32, a u kojima se ne koristi iznosi 96. Čvor se dalje grana prema stupcu tip leta. Gini vrijednost čvora je 0,25. Čvor 77 je lisni čvor te predstavlja dugolinijski let niskotarifnog prijevoznika. Uključuje 128 primjera. Catering vozilo se koristi u svih 128 primjera. Gini vrijednost čvora je 0. Čvor 1276 je lisni čvor te predstavlja tuzemni kratkolinijski let tradicionalnog zračnog prijevoznika. Uključuje 64 primjera. Catering vozilo se ne koristi u sva 64 primjera. Gini vrijednost čvora iznosi 0. Čvor 1277 je lisni čvor te predstavlja inozemni kratkolinijski let tradicionalnog prijevoznika. Uključuje 64 primjera. Catering vozilo se koristi u sva 64 primjera. Gini vrijednost čvora je 0. Čvor 1901 predstavlja je lisni čvor te predstavlja tuzemni kratkolinijski let niskotarifnog prijevoznika. Uključuje 64 primjera. Catering vozilo se ne koristi u sva 64 primjera. Gini vrijednost čvora je 0. Čvor 1902 predstavlja inozemni kratkolinijski let niskotarifnog prijevoznika te uključuje 64 primjera. Broj primjera za koje se catering vozilo koristi iznosi 32, a za koje se ne koristi iznosi 32. Čvor se dalje grana prema stupcu ugovorna obveza. Gini vrijednost čvora iznosi 0. Čvor 47551 je lisni čvor te predstavlja niskotarifne prijevoznike koji nemaju ugovornu obvezu obavljanja usluge cateringa. Uključuje 32 primjera. Catering vozilo se ne koristi u sva 32 primjera. Gini vrijednost čvora je 0. Čvor 47552 je lisni čvor te predstavlja niskotarifne prijevoznike koji imaju ugovornu obvezu obavljanja usluge cateringa. Uključuje 32 primjera. Catering vozilo se koristi u sva 32 primjera. Gini vrijednost čvora iznosi 0.

```

C:\Users\Drobac\OneDr
čvor:0
ne koristi se:416 koristi se:352
zračni prijevoznik
gini vrijednost: 0,28125
čvor:1
teretni zračni prijevoznik
ne koristi se:256 koristi se:0
gini vrijednost: 0
čvor:2
tradicionalni zračni prijevoznik
ne koristi se:64 koristi se:192
duljina leta
gini vrijednost: 0,25
čvor:3
niskotarifni zračni prijevoznik
ne koristi se:96 koristi se:160
duljina leta
gini vrijednost: 0,1875
čvor:51
kratkolinijski let
ne koristi se:64 koristi se:64
tip leta
gini vrijednost: 0
čvor:52
dugolinijski let
ne koristi se:0 koristi se:128
gini vrijednost: 0
čvor:76
kratkolinijski let
ne koristi se:96 koristi se:32
tip leta
gini vrijednost: 0,25
čvor:77
dugolinijski let
ne koristi se:0 koristi se:128
gini vrijednost: 0
čvor:1276
tuzemni let
ne koristi se:64 koristi se:0
gini vrijednost: 0
čvor:1277
inozemni let
ne koristi se:0 koristi se:64
gini vrijednost: 0
čvor:1901
tuzemni let
ne koristi se:64 koristi se:0
gini vrijednost: 0
čvor:1902
inozemni let
ne koristi se:32 koristi se:32
ugovorna obveza
gini vrijednost: 0
čvor:47551
nema ugovornu obvezu
ne koristi se:32 koristi se:0
gini vrijednost: 0
čvor:47552
ima ugovornu obvezu
ne koristi se:0 koristi se:32
gini vrijednost: 0

```

Slika 29. Stablo odlučivanja za catering vozilo

Izvor: Autor

4.2.4. Stablo odlučivanja za cisternu za vodu

Sljedeće stablo odlučivanja analizira upotrebu cisterne za vodu. Ukupan broj primjera za koji se cisterna za vodu koristi iznosi 192, a broj primjera za koji se ne koristi iznosi 576. Čvor 0 je korijenski čvor te se grana prema stupcu: razina kvalitete usluge zračne luke. Gini vrijednost ovog čvora iznosi 0,25. Čvor 1 je lisni čvor te predstavlja zračne luke niže razine kvalitete usluge. Uključuje 384 primjera. Cisterna za vodu se ne koristi u sva 384 primjera. Gini vrijednost čvora iznosi 0. Čvor 2 predstavlja zračne luke više razine kvalitete usluge te uključuje 384 primjera. Broj primjera za koje se cisterna za vodu koristi iznosi 192, a broj primjera za koje se ne koristi iznosi 192. Ovaj čvor se dalje grana prema stupcu: tip zračnog prijevoznika. Gini vrijednost ovog čvora iznosi 0,167. Čvor 51 je lisni čvor te predstavlja teretne zračne prijevoznike. Uključuje 128 primjera. Cisterna za vodu se ne koristi u svih 128 primjera. Gini vrijednost čvora je 0. Čvor 52 je lisni čvor te predstavlja tradicionalne zračne prijevoznike. Uključuje 128 primjera. Cisterna za vodu se koristi u svih 128 primjera. Gini vrijednost čvora iznosi 0. Čvor 53 predstavlja niskotarifne zračne prijevoznike te uključuje 128 primjera. Broj primjera za koje se koristi cisterna za vodu iznosi 64, a za koje se ne koristi iznosi 64. Ovaj čvor se grana prema stupcu: ugovorna obveza. Gini vrijednost ovog čvora iznosi 0. Čvor 1326 je lisni čvor te predstavlja niskotarifne zračne prijevoznike koji nemaju ugovornu obvezu. Uključuje

64 primjera. Cisterna za vodu se ne koristi u sva 64 primjera. Gini vrijednost čvora je 0. Čvor 1327 je lisni čvor te predstavlja niskotarifne zračne prijevoznike koji imaju ugovornu obvezu. Uključuje 64 primjera. Cisterna za vodu se koristi za sva 64 primjera. Gini vrijednost čvora je 0.

```

C:\Users\Drobac\OneDrive\Radna površina\z:
čvor:0
ne koristi se:576   koristi se:192
kvaliteta usluge
gini vrijednost: 0,25

čvor:1
niža razina kvalitete
ne koristi se:384   koristi se:0
gini vrijednost: 0

čvor:2
viša razina kvalitete
ne koristi se:192   koristi se:192
zračni prijevoznik
gini vrijednost: 0,166666666666667

čvor:51
teretni zračni prijevoznik
ne koristi se:128   koristi se:0
gini vrijednost: 0

čvor:52
tradicionalni zračni prijevoznik
ne koristi se:0     koristi se:128
gini vrijednost: 0

čvor:53
niskotarifni zračni prijevoznik
ne koristi se:64    koristi se:64
ugovorna obveza
gini vrijednost: 0

čvor:1326
nema ugovornu obvezu
ne koristi se:64    koristi se:0
gini vrijednost: 0

čvor:1327
ima ugovornu obvezu
ne koristi se:0     koristi se:64
gini vrijednost: 0

```

Slika 30. Stablo odlučivanja cisterna za vodu
Izvor: Autor

4.2.5. Stablo odlučivanja za utovarivač

Sljedeće stablo odlučivanja analizira upotrebu utovarivača. Ukupan broj primjera stabla odlučivanja za koji se utovarivač koristi iznosi 210, a za koji se ne koristi iznosi 558. Čvor 0 je korijenski čvor te se grana prema stupcu: kvaliteta usluge. Gini vrijednost ovog čvora iznosi 0,2478. Čvor 1 je lisni čvor te predstavlja zračne luke niže razine kvalitete usluge. Uključuje 384 primjera. Utovarivač se ne koristi za sva 384 primjera. Gini vrijednost čvora iznosi 0. Čvor 2 predstavlja zračne luke više razine kvalitete usluge te se sastoji od 384 primjera. Broj primjera za koje se utovarivač koristi iznosi 210, a za koje se ne koristi iznosi 174. Čvor se dalje grana prema stupcu zračni prijevoznik. Gini vrijednost čvora iznosi 0,1535. Čvor 51 je lisni čvor te predstavlja teretne zračne prijevoznike. Uključuje 128 primjera. Utovarivač se koristi za svih 128 primjera. Gini vrijednost čvora iznosi 0. Čvor 52 predstavlja tradicionalne zračne prijevoznike te uključuje 128 primjera. Broj primjera za koje se utovarivač koristi iznosi 82, a

za koje se ne koristi iznosi 46. Čvor se dalje grana prema stupcu: tip zrakoplova. Gini vrijednost čvora iznosi 0,2021. Čvor 53 je lisni čvor te predstavlja niskotarifne zračne prijevoznike. Uključuje 128 primjera. Utovarivač se ne koristi u svih 128 primjera. Gini vrijednost čvora iznosi 0.

Čvor 1301 predstavlja uskotrupne zrakoplove tradicionalnih prijevoznika uključuje 64 primjera. Broj primjera za koje se utovarivač ne koristi iznosi 46, a za koje se koristi iznosi 18. Čvor se dalje grana prema stupcu model zrakoplova. Gini vrijednost čvora iznosi 0. Čvor 1302 je lisni čvor te predstavlja širokotrupne zrakoplove tradicionalnih prijevoznike. Uključuje 64 primjera. Utovarivač se koristi u sva 64 primjera. Gini vrijednost čvora iznosi 0. Čvor 32526 je lisni čvor koji predstavlja zrakoplov A318. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32527 je lisni čvor koji predstavlja zrakoplov A319ceo. Uključuje tri primjera. Utovarivač se koristi u sva tri primjera. Gini vrijednost je 0. Čvor 32528 je lisni čvor koji predstavlja zrakoplov A319neo. Uključuje dva primjera. Utovarivač se koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32529 je lisni čvor koji predstavlja zrakoplov A320ceo. Uključuje tri primjera. Utovarivač se koristi u sva tri primjera. Gini vrijednost čvora je 0. Čvor 32530 je lisni čvor koji predstavlja zrakoplov A320neo. Uključuje dva primjera. Utovarivač se koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32531 je lisni čvor koji predstavlja zrakoplov A321ceo. Uključuje tri primjera. Utovarivač se koristi u sva tri primjera. Gini vrijednost čvora je 0. Čvor 32532 je lisni čvor koji predstavlja zrakoplov A321neo. Uključuje tri primjera. Utovarivač se koristi za sva tri primjera. Gini vrijednost čvora je 0. Čvor 32533 je lisni čvor koji predstavlja zrakoplov A220 - 100. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32534 je lisni čvor koji predstavlja zrakoplov A220 - 300. Uključuje tri primjera. Utovarivač se ne koristi u sva tri primjera. Gini vrijednost čvora je 0.

Čvor 32535 je lisni čvor koji predstavlja zrakoplov B737 - 700. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32536 je lisni čvor koji predstavlja zrakoplov B737 - 800. Uključuje tri primjera. Utovarivač se ne koristi u sva tri primjera. Gini vrijednost čvora je 0. Čvor 32537 je lisni čvor koji predstavlja zrakoplov B737 - 900. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32538 je lisni čvor koji predstavlja zrakoplov B737 MAX7. Uključuje tri primjera. Utovarivač se ne koristi u sva tri primjera. Gini vrijednost čvora je 0. Čvor 32539 je lisni čvor koji predstavlja zrakoplov B737 MAX8. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32540 je lisni čvor koji predstavlja zrakoplov B737 MAX9. Uključuje tri primjera. Utovarivač se ne koristi u sva tri primjera. Gini vrijednost čvora je 0. Čvor 32541 je lisni čvor koji predstavlja zrakoplov B737 MAX10. Uključuje tri primjera. Utovarivač se ne koristi za sva tri primjera. Gini vrijednost čvora je 0.

Čvor 32542 je lisni čvor koji predstavlja zrakoplov E170. Uključuje tri primjera. Utovarivač se ne koristi za sva tri primjera. Gini vrijednost čvora je 0. Čvor 32543 je lisni čvor koji predstavlja zrakoplov E175. Uključuje tri primjera. Utovarivač se ne koristi za sva tri primjera. Gini vrijednost čvora je 0. Čvor 32544 je lisni čvor koji predstavlja zrakoplov E190. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0. Čvor

32545 je lisni čvor koji predstavlja zrakoplov E195. Uključuje tri primjera. Utovarivač se ne koristi u sva tri primjera. Gini vrijednost čvora je 0. Čvor 32546 je lisni čvor koji predstavlja zrakoplov E175 - E2. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32547 je lisni čvor koji predstavlja zrakoplov E190 – E2. Uključuje tri primjera. Utovarivač se ne koristi u sva tri primjera. Gini vrijednost čvora je 0. Čvor 32548 je lisni čvor koji predstavlja zrakoplov E195 – E2. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0.

Čvor 32549 je lisni čvor koji predstavlja zrakoplov Tu - 204. Uključuje dva primjera. Utovarivač se ne koristi u oba primjera. Gini vrijednost čvora je 0. Čvor 32550 je lisni čvor koji predstavlja zrakoplov Tu - 214. Uključuje dva primjera. Utovarivač se koristi u oba primjera. Gini vrijednost čvora je 0.

```

C:\Users\Drobac\OneDrive\Re... C:\Users\Drobac\One
čvor:0 gini vrijednost: 0
ne koristi se:558 koristi se:210 čvor:32533
kvaliteta usluge A220-100
gini vrijednost: 0,247802734375 ne koristi se:3 koristi se:0
čvor:1 gini vrijednost: 0
niža razina kvalitete čvor:32534
ne koristi se:384 koristi se:0 A220-300
gini vrijednost: 0 ne koristi se:3 koristi se:0
čvor:2 gini vrijednost: 0
viša razina kvalitete čvor:32535
ne koristi se:174 koristi se:210 B737-700
zračni prijevoznik ne koristi se:2 koristi se:0
gini vrijednost: 0,153483072916667 gini vrijednost: 0
čvor:51 čvor:32536
teretni zračni prijevoznik B737-800
ne koristi se:0 koristi se:128 ne koristi se:3 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:52 čvor:32537
tradicionalni zračni prijevoznik B737-900
ne koristi se:46 koristi se:82 ne koristi se:2 koristi se:0
gini vrijednost: 0,2021484375 gini vrijednost: 0
čvor:53 čvor:32538
niskotarifni zračni prijevoznik B737 MAX7
ne koristi se:128 koristi se:0 ne koristi se:3 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:1301 čvor:32539
uskotrupni zrakoplov B737 MAX8
ne koristi se:46 koristi se:18 ne koristi se:2 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:1302 čvor:32540
širokotrupni zrakoplov B737 MAX9
ne koristi se:0 koristi se:64 ne koristi se:3 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32526 čvor:32541
A318 B737 MAX10
ne koristi se:2 koristi se:0 ne koristi se:3 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32527 čvor:32542
A319ceo E170
ne koristi se:0 koristi se:3 ne koristi se:3 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32528 čvor:32543
A319neo E195
ne koristi se:0 koristi se:2 ne koristi se:3 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32529 čvor:32544
A320ceo E190
ne koristi se:0 koristi se:3 ne koristi se:2 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32530 čvor:32545
A320neo E195
ne koristi se:0 koristi se:2 ne koristi se:3 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32531 čvor:32546
A321ceo E175-E2
ne koristi se:0 koristi se:3 ne koristi se:2 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32532 čvor:32547
A321neo E190-E2
ne koristi se:0 koristi se:3 ne koristi se:3 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32549 čvor:32548
Tu-204 E195-E2
ne koristi se:2 koristi se:0 ne koristi se:2 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32550 čvor:32549
Tu-214 Tu-204
ne koristi se:0 koristi se:2 ne koristi se:2 koristi se:0
gini vrijednost: 0 gini vrijednost: 0
čvor:32550 čvor:32550
Tu-214 Tu-214
ne koristi se:0 koristi se:2 ne koristi se:0 koristi se:2
gini vrijednost: 0 gini vrijednost: 0

```

Slika 31. Stablo odlučivanja utovarivač

Izvor: Autor

5. ZAKLJUČAK

Metodom stabla odlučivanja se izvršava klasifikacija znanja na temelju poznatih primjera. Prije upotrebe stabla odlučivanja potrebno je pripremiti potrebne podatke. Metoda se temelji na rekurzivnoj podjeli podataka sve dok je to moguće, tj. dok se ne dođe do primjera koji predstavlja konačnu klasifikaciju znanja. Podjelu podataka obavlja algoritam. Postoji više algoritama koji izvršavaju podjelu podataka. Stabla odlučivanja su lako primjenjiva te se koriste u raznim disciplinama i znanostima.

Prihvat i otprema zrakoplova je ključan proces za zračne prijevoznike i zračne luke. Sigurnost putnika, osoblja i zrakoplova je najvažnija pri obavljanju procesa prijvata i otpreme zrakoplova. Proces se sastoji od niza aktivnosti koje je potrebno obaviti prema propisanim pravilima i procedurama. Zračne luke i prijevoznici nastoje skratiti vrijeme prijvata i otpreme zrakoplova. Sva kašnjenja u obavljanju aktivnosti donose gubitke zračnim lukama i prijevoznicima. Aktivnosti nisu iste za svaki tip zrakoplova ili zračnog prijevoznika. Svaka aktivnost se izvodi pomoću specijalizirane opreme kojom upravlja posebno obučeno osoblje. Korištenje opreme ovisi karakteristikama zrakoplova, prijevoznika, zračne luke, vremenskim uvjetima...

Svrha ovog rada je izraditi stablo odlučivanja koje će pokazati o kojim karakteristikama zrakoplova, zračnog prijevoznika ili zračne luke ovisi da li će se pojedini tip opreme upotrijebiti. Utvrđeno je da korištenje većine opreme ovisi o jednom ili dva uvjeta, te da izrada stabla odlučivanja za tu opremu nije potrebna. Za manji dio opreme, konačna stabla odlučivanja su složena, te je izrada stabla odlučivanja za tu opremu primjenjiva. Analiza je pokazala da su najvažnije karakteristika kvaliteta usluge zračne luke, tj. da li zračna luka posjeduje potrebnu opremu i tip zračnog prijevoznika. Zračne luke mogu koristiti ovu analizu za nabavu i planiranje korištenja opreme na temelju tipova zrakoplova i zračnih prijevoznika koji su zastupljeni na pojedinoj zračnoj luci.

POPIS LITERATURE

- [1] Towards Data Science. Preuzeto sa: <https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052> [Pristupljeno: srpanj 2021.]
- [2] KDnuggets. Preuzeto sa: <https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html> [Pristupljeno: srpanj 2021.]
- [3] Great Learning . Preuzeto sa: <https://www.mygreatlearning.com/blog/decision-tree-algorithm/> [Pristupljeno: srpanj 2021.]
- [4] Towards Data Science. Preuzeto sa: <https://towardsdatascience.com/decision-trees-for-classification-id3-algorithm-explained-89df76e72df1> [Pristupljeno: srpanj 2021.]
- [5] Core. Preuzeto sa: <https://core.ac.uk/download/pdf/295600592.pdf> [Pristupljeno: srpanj 2021.]
- [6] Open Genius. Preuzeto sa: <https://iq.opengenus.org/id3-algorithm/> [Pristupljeno: srpanj 2021.]
- [7] Towards Data Science. Preuzeto sa: <https://towardsdatascience.com/from-a-single-decision-tree-to-a-random-forest-b9523be65147> [Pristupljeno: srpanj 2021.]
- [8] Analytics Steps. Preuzeto sa: <https://www.analyticssteps.com/blogs/what-gini-index-and-information-gain-decision-trees> [Pristupljeno: srpanj 2021.]
- [9] Towards Data Science. Preuzeto sa: <https://towardsdatascience.com/what-is-the-c4-5-algorithm-and-how-does-it-work-2b971a9e7db0> [Pristupljeno: srpanj 2021.]
- [10] Machine Learning Mastery. Preuzeto sa: <https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/> [Pristupljeno: srpanj 2021.]
- [11] Open Genius. Preuzeto sa: <https://iq.opengenus.org/cart-algorithm/> [Pristupljeno: srpanj 2021.]
- [12] PLOS. Preuzeto sa: <https://journals.plos.org> [Pristupljeno: srpanj 2021.]
- [13] Towards Data Science. Preuzeto sa: <https://towardsdatascience.com/clearly-explained-top-2-types-of-decision-trees-chaid-cart-8695e441e73e> [Pristupljeno: srpanj 2021.]
- [14] Select statistical services. Preuzeto sa: <https://select-statistics.co.uk/blog/chaid-chi-square-automatic-interaction-detector/> [Pristupljeno: srpanj 2021.]
- [15] Analytics Vidhya. Preuzeto sa: <https://www.analyticsvidhya.com/blog/2021/05/implement-of-decision-tree-using-chaid/>
- [16] Machine Learning Mastery. Preuzeto sa: <https://machinelearningmastery.com/multivariate-adaptive-regression-splines-mars-in-python/> [Pristupljeno: srpanj 2021.]
- [17] UC Business Analytics R Programming Guide. Preuzeto sa: <http://uc-r.github.io/mars> [Pristupljeno: srpanj 2021.]
- [18] DisplayR. Preuzeto sa: <https://www.displayr.com/machine-learning-pruning-decision-trees/> [Pristupljeno: srpanj 2021.]
- [19] what – when – how. Preuzeto sa: <http://what-when-how.com/artificial-intelligence/decision-tree-applications-for-data-modelling-artificial-intelligence/> [Pristupljeno: srpanj 2021.]
- [20] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/ground-handling-ramp-operations-aircraft-ground-support-equipment-gse-machines-that-supplement-the-airplane/> [Pristupljeno: kolovoz 2021.]

- [21] Pavlin S, Bračić M. Tehnologija prihvata i otpreme zrakoplova, Sveučilište u Zagrebu Fakultet prometnih znanosti, 2017.
- [22] SKYbrary. Preuzeto sa: [https://www.skybrary.aero/index.php/Passenger Boarding Bridge](https://www.skybrary.aero/index.php/Passenger_Boarding_Bridge) [Pristupljeno: kolovoz 2021.]
- [23] TK Elevator China. Preuzeto sa: <https://www.tkelevator.com.cn/cn-en/products/airport-solutions/> [Pristupljeno: kolovoz 2021.]
- [24] AeroExpo. Preuzeto sa: <https://www.aeroexpo.online/aeronautic-manufacturer/towed-stairs-3934.html> [Pristupljeno: kolovoz 2021.]
- [25] Airport Suppliers. Preuzeto sa: <https://www.airport-suppliers.com/product/self-propelled-passenger-stairs/> [Pristupljeno: kolovoz 2021.]
- [26] Wayback Machine. Preuzeto sa: https://web.archive.org/web/20130312074201/http://www.used-buses.net/bustypes/airport_bus.asp [Pristupljeno: kolovoz 2021.]
- [27] AeroMobiles. Preuzeto sa: <https://www.aeromobiles.com.sg/news/118-cimc-electric-apron-bus> [Pristupljeno: kolovoz 2021.]
- [28] Ambulift Bulmor. Preuzeto sa: <https://www.bulmor-airground.com/uk/special-prm-boarding-ambulift/> [Pristupljeno: kolovoz 2021.]
- [29] Aero Corner. Preuzeto sa: <https://aerocorner.com/blog/best-aircraft-wheel-chocks/> [Pristupljeno: kolovoz 2021.]
- [30] Daily Mail Online. Preuzeto sa: <https://www.dailymail.co.uk/news/article-4696004/Worker-chases-rolling-plane-removing-wheel-chocks.html> [Pristupljeno: kolovoz 2021.]
- [31] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/aircraft-ground-power-unit/> [Pristupljeno: kolovoz 2021.]
- [32] Airport Suppliers. Preuzeto sa: <https://www.airport-suppliers.com/product/solid-state-ground-power-unit-400-hz/> [Pristupljeno: kolovoz 2021.]
- [33] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/how-aircraft-catering-works/> [Pristupljeno: kolovoz 2021.]
- [34] AeroExpo. Preuzeto sa: <https://www.aeroexpo.online/prod/mallaghan/product-168807-12291.html> [Pristupljeno: kolovoz 2021.]
- [35] LAS – 1. Preuzeto sa: <https://las1.lv/en/equipment/ground-support-equipment/baggage-handling-on-apron/baggage-carts.html> [Pristupljeno: kolovoz 2021.]
- [36] Orientitan. Preuzeto sa: <http://uld-equipment.com/2-1-open-airport-baggage-cart.html> [Pristupljeno: kolovoz 2021.]
- [37] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/how-cargo-is-loaded-and-unloaded-from-an-airplane/> [Pristupljeno: kolovoz 2021.]
- [38] TEMG. Preuzeto sa: <http://www.temg.es/equiposEN/autopropulsados/TG-6000.htm> [Pristupljeno: kolovoz 2021.]
- [39] Orientitan. Preuzeto sa: <http://uld-equipment.com/1-1-10ft-pallet-dolly.html> [Pristupljeno: kolovoz 2021.]
- [40] JBT AeroTech. Preuzeto sa: <https://www.jbtc.com/aerotech/products-and-services/ground-support-equipment/cargo-loaders/commander> [Pristupljeno: kolovoz 2021.]

- [41] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/aircraft-towing-tractor-pushback-tractor-or-tug-the-machine-that-wrestles-with-the-giant-airplane/> [Pristupljeno: kolovoz 2021.]
- [42] Airport Industry – News. Preuzeto sa: <https://airportindustry-news.com/all-electric-aircraft-pushback-tug-in-operation-at-munich/> [Pristupljeno: kolovoz 2021.]
- [43] Raguz V. Analiza prihvata i otpreme zrakoplova prema poslovnim modelima zračnog prijevoznika, Sveučilište u Zagrebu, Fakultet prometnih znanosti 2020.
- [44] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/aircraft-lavatory-service-truck-the-vehicle-that-gets-the-dirty-job-done/> [Pristupljeno: kolovoz 2021.]
- [45] Shutterstock. Preuzeto sa: <https://www.shutterstock.com/search/lavatory+service+vehicles> [Pristupljeno: kolovoz 2021.]
- [46] Wikimedia Commons. Preuzeto sa: https://commons.wikimedia.org/wiki/File:Potable_Water_Trucks.JPG [Pristupljeno: kolovoz 2021.]
- [47] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/how-airplane-refueling-works/> [Pristupljeno: kolovoz 2021.]
- [48] Wikipedia. Preuzeto sa: https://en.wikipedia.org/wiki/File:Shell_aviation_fuel_truck_at_V%C3%A6rnes.jpg [Pristupljeno: kolovoz 2021.]
- [49] Guinault. Preuzeto sa: <http://www.guinault.com/en/aviation/acu/> [Pristupljeno: kolovoz 2021.]
- [50] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/what-is-an-aircraft-air-start-unit-asu/> [Pristupljeno: kolovoz 2021.]
- [51] Guinault. Preuzeto sa: <http://www.guinault.com/en/aviation/asu/> [Pristupljeno: kolovoz 2021.]
- [52] Aviation Learnings. Preuzeto sa: <https://aviationlearnings.com/aircraft-de-icing-truck-the-machine-that-saves-airplanes-from-freezing-in-winters/> [Pristupljeno: kolovoz 2021.]
- [53] Aero Expo. Preuzeto sa: <https://www.aeroexpo.online/aeronautic-manufacturer/aircraft-de-icing-vehicle-1148.html> [Pristupljeno: kolovoz 2021.]

POPIS KRATICA

| | |
|-------|---|
| ID3 | (Iterative Dichtomiser 3) |
| CART | (Classification and Regression Trees) |
| MARS | (Multivariate Adaptive Regression Splines) |
| CHAID | (Chi – Square Automatic Interaction Detection) |
| TRT | (Turn Round Time) Vrijeme prihvata i otpreme zrakoplova |
| PRM | (Passengers with Restricted Mobility) Putnici sa smanjenom pokretljivošću |
| GPU | (Ground Power Unit) Zemaljski izvor električne energije |
| ULD | (Unit Load Device) Jedinično sredstvo utovara |

POPIS SLIKA

| | |
|--|----|
| Slika 1. Zračni most..... | 11 |
| Slika 2. Vučene stepenice..... | 12 |
| Slika 3. Samohodne stepenice..... | 12 |
| Slika 4. Autobus..... | 13 |
| Slika 5. Liftmobil..... | 14 |
| Slika 6. Podmetač..... | 14 |
| Slika 7. GPU..... | 15 |
| Slika 8. Catering vozilo..... | 16 |
| Slika 9. Kolica za prijevoz prtljage..... | 16 |
| Slika 10. Mobilna pokretna traka..... | 17 |
| Slika 11. ULD kolica..... | 18 |
| Slika 12. Utovarivač..... | 18 |
| Slika 13. Vozilo za izguravanje..... | 19 |
| Slika 14. Vozilo za servis otpadnih voda..... | 21 |
| Slika 15. Vozilo za opskrbu zrakoplova vodom..... | 21 |
| Slika 16. Cisterna za opskrbu zrakoplova gorivom..... | 22 |
| Slika 17. Uređaj za klimatizaciju..... | 23 |
| Slika 18. Zračni starter..... | 24 |
| Slika 19. Vozilo za odleđivanje i zaštitu zrakoplova od zaleđivanja..... | 24 |
| Slika 20. Metoda <i>giniVrijednost</i> | 26 |
| Slika 21. Metoda <i>srednjaGiniVrijednost</i> | 26 |
| Slika 22. Metoda <i>podjeliPodatke</i> | 27 |
| Slika 23. Metoda <i>podjeliPodatke</i> | 27 |
| Slika 24. Metoda <i>brojIshoda</i> | 28 |
| Slika 25. Metoda <i>napraviStablo</i> | 28 |
| Slika 26. Metoda <i>napraviStablo</i> | 29 |
| Slika 27. Stablo odlučivanja za samohodne stepenice..... | 30 |
| Slika 28. Stablo odlučivanja za vučene stepenice..... | 31 |
| Slika 29. Stablo odlučivanja za catering vozilo..... | 33 |
| Slika 30. Stablo odlučivanja cisterna za vodu..... | 34 |
| Slika 31. Stablo odlučivanja utovarivač..... | 36 |

POPIS TABLICA

| | |
|--|----|
| Tablica 1. Prikaz jednog primjera skupa vrijednosti svih uvjeta..... | 25 |
|--|----|



Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj završni rad
isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na
objavljenu literaturu što pokazuju korištene bilješke i bibliografija.
Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz
necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.
Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj
visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.
Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu završnog rada
pod naslovom **Analiza opreme za prihvat i otpremu zrakoplova temeljena na stablu**
odlučivanja
na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom
repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

U Zagrebu, 6.9.2021

Student/ica:

L.D.

(potpis)