

TECHNICKÁ UNIVERZITA V KOŠICIACH
FAKULTA ELEKTROTECHNIKY A INFORMATIKY

**GENEROVANIE ROZHODOVACÍCH STROMOV C4.5 –
LIETAŤ ALEBO NELIETAŤ**
Zadanie na predmet Strojové učenie

2023

Štefan Kando

PodĎakovanie

Ďakujem Leteckej škole FUTURE FLY SK.ATO.11 Bidovce (<https://leteckaskola.sk/>) za poskytnutie dát, bez ktorých by tento projekt nebol možný.

Obsah

Úvod	4
1. Teoretický popis algoritmu C4.5	5
2. Popis dát.....	7
3. Popis postupu a funkcií	10
3.1. Hlavný chod programu	10
3.2. Implementácia C4.5 algoritmu.....	12
3.2.1. Triedy.....	12
3.2.2. Implementácia.....	12
4. Vyhodnotenie.....	14
Záver.....	16
Zoznam použitej literatúry	17

Úvod

Pilot, či už dopravný alebo súkromný, je vystavený veľa rozhodnutiam, ku ktorým sa musí stavať zodpovedne. Jedno z najviac základných rozhodnutí je rozhodnutie, či ísť lietať, alebo nie.

Toto rozhodnutie závisí na veľa faktoroch, a vo veľa z nich, ako zdravotný a duševný stav pilota, technický stav lietadla a pod., nám umelá inteligencia nepomôže. Umelá inteligencia nám však môže pomôcť s týmto rozhodnutím vzhľadom na počasie, najmä pri súkromnom lietaní za vidu (Visual Flight Rules – VFR).

Cieľom tohto projektu je vytvoriť nástroj, ktorý pilotom, najmä začínajúcim, ako som ja, pomôže sa rozhodnúť, či sa na let vydať alebo nie, vzhľadom na to, aké je počasie. Dôležité je však poznamenať, že od začiatku je projekt vyvíjaný len ako pomôcka, alebo experiment, ako by sa na základe analýzy dát počítač vzhľadom na počasie rozhodoval ísť lietať alebo nie – ale samotné rozhodnutie vždy ostáva na pilotovi.

Keďže nie všetci piloti sú programátori, a chceme vytvoriť rozhodovací model, ktorému budú rozumieť všetci, použijeme jeden z algoritmov na generovanie rozhodovacích stromov. Letectvo je odvetvie, kde na generalizáciu veľmi často nie je priestor, a preto pre najväčšiu presnosť sa generalizácii vyhneme a budeme pracovať s numerickými dátami o počasí. Na prácu s numerickými dátami je z algoritmov generovania rozhodovacích stromov vhodný C4.5.

V nasledujúcich kapitolách si priblížime, ako C4.5 funguje z teoretického hľadiska, ukážeme a vysvetlíme si dáta, ktoré máme k dispozícii, následne na to popíšeme samotnú implementáciu tohto algoritmu a nakoniec ho vyhodnotíme.

1. Teoretický popis algoritmu C4.5

Algoritmus C4.5 je pokračovanie algoritmu ID3, ktorý bol navrhnutý Rossom Quinlanom. C4.5 je algoritmus strojového učenia, ktorý sa používa na generovanie rozhodovacích stromov z tréningových dát. Je neinkrementálny (po natrénovaní sa neprispôsobuje novým dátam) a klasifikačný (výstupom z neho je trieda – v našom prípade rozhodnutie – lietať/nelietat).

Algoritmus C4.5 pracuje rekurzívne tak, že strom rozširuje, kým nedosiahne kritérium zastavenia. Kritérium zastavenia môže byť napríklad dosiahnutie určitej hĺbky stromu alebo počet príkladov v uzle. C4.5 je schopný spracovávať záznamy s chýbajúcimi hodnotami a dokáže pracovať s numerickými a kategorickými atribútmi.

Pri generovaní rozhodovacích stromov algoritmus C4.5 postupuje nasledovne:

1. Ak sú všetky príklady v aktuálnom uzle rovnakého typu (triedy), vytvorí sa listový uzol s priradenou triedou a rekúzia končí.
2. Ak sú atribúty vyčerpané, vytvorí sa listový uzol s triedou, ktorá je v aktuálnom uzle najčastejšia.
3. Ak žiadny z predchádzajúcich prípadov nenastal, vyberie sa atribút, ktorý poskytuje najlepšie delenie podľa kritéria informačného zisku alebo pomerného zisku informácií. Potom sa pre každú hodnotu vybraného atribútu vytvorí nový uzol.
4. Pre každý nový uzol sa rekurzívne opakujú kroky 1 až 3.

Informačný zisk sa vypočíta nasledovne [1]:

$$I(S, A) = p(A_k)(H(S) - H(S, A)) - p(A_{unk})$$

Informačný zisk = pravdepodobnosť, že sa atribút A_k objaví v rozhodovacom strome * (entrópia celého datasetu – pomerová entrópia (entrópia s ohľadom na atribút A)) – pravdepodobnosť neznámych hodnôt

Modifikovaná pomerová entrópia sa počíta nasledovne:

$$H_p(S, A) = -\sum p(a_j) \log_2 p(a_j) - p(a_{unk}) \log_2 p(a_{unk})$$

Algoritmus C4.5 je schopný riešiť problémy s preučeníím pomocou postupu, ktorý sa nazýva "orezávanie". Orezávanie stromu spočíva v tom, že sa zredukujú niektoré vetvy stromu, ktoré nespĺňajú určité kritérium, napríklad minimálny počet príkladov v uzle. Tým sa znižuje zložitosť modelu, predchádza sa overfittingu a zvyšuje sa schopnosť generalizácie.

C4.5 je obľúbený a široko používaný algoritmus, ktorý má výbornú schopnosť klasifikovať neznáme dáta na základe trénovacích dát. Jeho implementácie sú dostupné v rôznych programovacích jazykoch a knižniciach. Výhodou rozhodovacieho stromu je jeho ilustratívnosť, tým pádom je zrozumiteľný aj pre laikov.

2. Popis dát

Keďže chceme skúmať rozhodovanie súkromných pilotov, ktorí lietajú za vidu (VFR), niet lepšieho zdroju dát, ako je súkromné/športové letisko. Letecká škola FUTURE FLY SK.ATO.11 Bidovce, ktorá je spojená s aeroklubom Bidovce a prevádzkuje na súkromnom letisku Bidovce, si udržuje svojich žiakov, pilotov a inštruktorov organizovaných pomocou online kalendára: <https://leteckaskola.sk/kalendar-letov/>.

Tento kalendár má svoju bázu dát v Google kalendári. Všetky údaje sa nahrávajú do Google kalendára, a tie sa pomocou API (Application Programming Interface – programátorské rozhranie) prenášajú na webovú stránku. Avšak na tomto online kalendári je dostupná história len 7 dní / 168 hodín. Potrebovali by sme teda pracovať priamo s bazou dát v Google kalendári.

Letecká škola FUTURE FLY SK.ATO.11 Bidovce bola natoľko dobrá, že mi tieto dáta poskytla vo forme súboru ICS. V tomto súbore sa v čase prijatia tohto súboru nachádzalo 9046 záznamov o letoch, servisoch, školeniach, a hlavne – oznamoch o zrušení lietania z dôvodu nepriaznivého počasia - od roku 2017 do roku 2023. **Tento súbor obsahuje mená pilotov a inštruktorov, ako aj telefónne čísla a iné citlivé informácie. Vzhľadom na citlivosť týchto informácií tieto dáta nebudem ďalej šíriť a nebudú súčasťou odovzdania môjho zadania.** Funkčnosť zadania s týmito dátami však bude prezentovaná pri osobnej prezentácii.

Pre naše potreby však žiadne tieto citlivé informácie nepotrebujeme. Pre nás je podstatné, kedy sa lietalo, a kedy nie. K tomu nám pomôže Regular Expression (RegEx), ktorý tieto dáta vyfiltruje pomocou popisu danej aktivity v Google kalendároch. Ak popis bude obsahovať reťazec „zrušené“, budeme predpokladať, že sa tento deň nelietalo. Pokiaľ bude popis začínať na reťazec „FF“ (takto sú označené lety leteckej školy, predpokladám, že FF znamená Future Fly), vieme, že sa tento deň lietalo. Zvyšné dni necháme tak, pretože nevieme, či sa nelietalo kvôli nepriaznivému počasiu, alebo o lietanie nebol záujem.

Ďalej potrebujeme dáta o počasí. Predtým, než budeme hľadať zdroje, si potrebujeme určiť, aké dáta potrebujeme. Na to si musíme rozanalyzovať, podľa čoho sa pilot rozhoduje a aké informácie má k dispozícii.

Limity počasia pre lietanie vieme rozdeliť do troch kategórií:

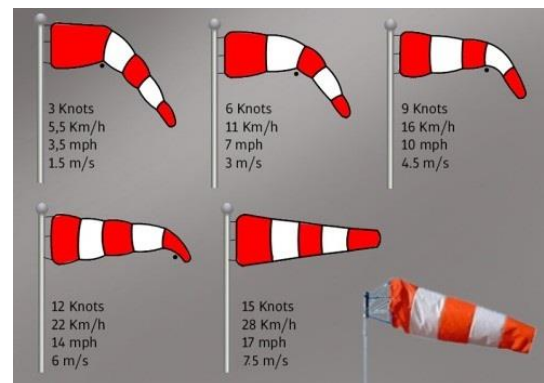
1. Predpísané limity počasia letu za vidu (VFR) [2]:
 - Viditeľnosť musí byť minimálne 5000 m
 - Oblačnosť musí byť minimálne 300 m (1000 stôp) vertikálne a 1500 m horizontálne od lietadla

2. Predpísané limity daného lietadla: Vzhľadom na dané lietadlo existujú limity, v ktorých lietadlo môže lietať, a tie sa vzťahujú:
 - Na teplotu: Lietadlo nesmie lietať v podmienkach vzniku námrazy ak na to nie je vybavené. Taktiež, vzhľadom na krátku dráhu v Bidovciach (400 m), lietadlo nemôže vzlietnuť ak je veľmi horúco (teplý vzduch je redší, tým pádom je vztlak menší)
 - Na silu a smer vetra: Lietadlo by malo vždy ideálne vzlietať a pristávať proti vetru. Niekedy to však nie je možné, ak vietor fúka kolmo na dráhu. Každé lietadlo má limit, v akom silnom bočnom vetre dokáže vzlietnuť a pristáť. Taktiež existujú limity na vzlet pri zadnom vetre, ak si to okolnosti vyžadujú.
3. Osobné limity pilota: Tie sú pre každého pilota individuálne. Môžu sa vzťahovať napríklad k sile bočného vetra. Vzlet a pristávanie pri bočnom vetre sú zložité manévry, a každý pilot má limit, aký silný bočný vietor zvládne.

Keď pilot na zemi kontroluje počasie, má k dispozícii letecké informačné služby. Tieto služby platia pre Košické letisko, ale keďže sú Bidovce pomerne blízko, sú dobrým smerodajcom. Tieto dáta obsahujú:

- Smer a silu vetra
- Viditeľnosť a zrážky
- Stav oblačnosti
- Teplota a rosný bod
- Tlak vzduchu vzhľadom na hladinu mora (pre nastavenie výškomerov)

Smer a sila vetra sa dajú odčítať z veterného rukáva priamo na letisku.



Veterný rukáv

Aby sme natrénovali relatívne spoľahlivý rozhodovací strom, musíme ho trénovať na rozmedzí minimálne dva roky, aby sme sa vyhli anomáliám, ktoré by mohli nastať len počas jedného roka. Pre naše potreby som sa rozhodol použiť interval 1/1/2021 – 1/1/2023. Na získanie dát o počasí by sme tiež mohli využiť API, avšak API o počasí s takou ďalekou históriou by bolo drahé a ťažké na zohnanie, preto som sa rozhodol získať dáta jednorazovo. Pre naše potreby mi vyhovovala stránka www.meteomatics.com/en/download-weather-data. Do tejto stránky som zadal presné súradnice

vzťažného bodu letiska, a nechal si generovať dáta od 1/1/2021 12:00 do 1/1/2023 12:00 v 24 hodinových intervaloch.

Locations

Latitude: 48.7428028, Longitude: 21.448325
Lat: 48.7428028 - Lon: 21.448325

Date & Time

01/01/2021 12:00 – 01/01/2023 12:00
Time zone: Europe/Bratislava Resolution: daily

Parameters

Temperature (mean, min or max over selected time interval)
Height: 2m Interval: 24h Measure: max Unit: Celsius

Effective cloud cover Unit: %

Amount of precipitation Interval: 24h Unit: mm

Visibility Unit: m

Wind direction Height: 10m Unit: d

Wind speed Height: 10m Unit: kn

Locations

Enter, select or specify the locations for which you would like weather data.

Latitude: 48.7428028, Longitude: 21.448325

Date & Time

Choose a date or date range, and time, as well as the time zone.

Start Date * 01/01/2021 Start Time * 12:00
End Date * 01/01/2023 End Time * 12:00
Time Zone * Europe/Bratislava (CET)
Resolution * daily

Nastavenie stránky meteomatics.com pred exportom dát

V týchto intervaloch som si dal exportovať tieto dáta:

- Maximálna teplota za 24 hodín
- Pokrytie oblohy oblačnosťou v % (výška oblačnosti nebola dostupná)
- Zrážky za 24 hodín v mm
- Viditeľnosť v m
- Smer a sila vetra v uzloch

Finálny dataset teda vznikol prienikom dát o lietaní a o počasí medzi 1/1/2021 a 1/1/2023. Tento prienik obsahuje 552 záznamov, z toho 451 pozitívnych a 101 negatívnych.

3. Popis postupu a funkcií

3.1. Hlavný chod programu

Naša implementácia algoritmu C4.5 je navrhnutá tak, aby bola čo najjednoduchšia a najzrozumiteľnejšia. Program je robený ako konzolová aplikácia v programovacom jazyku C#. Dáta program číta z predurčeného priečinku „data“ vo vnútri projektového priečinku, kde program očakáva dáta o počasí a ICS dáta z kalendára. Kód je dobre okomentovaný s komentármi v angličtine. Hlavnou časťou kódu je funkcia Main, ktorá riadi celý proces generovania rozhodovacieho stromu:

1. Na začiatku program vytvorí hlavičku a potom načíta súbor s údajmi o počasí a údajmi z kalendára. Ak je jeden z týchto súborov chýbajúci alebo poškodený, program sa ukončí s chybovou správou.
2. Nasleduje proces vytvárania dátového setu pre tréning. Program najskôr kombinuje údaje o počasí s údajmi z kalendára. Potom vyberá pozitívne a negatívne príklady, ktoré budú použité na tréning a testovanie.
3. Vytvára sa tréningový set a testovací set. Tieto sady sú náhodne zamiešané, aby sa zabezpečilo, že strom nebude pristranný vzhľadom na poradie príkladov.
4. Generuje sa rozhodovací strom pomocou algoritmu C4.5. Tento proces je monitorovaný stopkou, aby sme mohli vidieť, ako dlho trvá.
5. Vytvára sa diagnostický reťazec, ktorý obsahuje informácie o nastaveniach generovania a výkonnosti algoritmu.
6. Potom sa vytlačia diagnostické informácie o presnosti, správnosti, návratnosti a F1 skóre pre tréningové, testovacie a všetky dáta.
7. Nakoniec sa generujú výstupné súbory. Tieto súbory sú uložené do novej zložky v priečinku „data“, ktorá má názov „dátum-čas“. Obsahujú diagnostický reťazec, Mermaid syntax pre vizualizáciu stromu a Excel súbory s dátami pre tréning, testovanie a všetky dáta. Mermaid syntax môže byť použitý na vizualizáciu stromu na www.mermaid.live. Excel súbory obsahujú dataset s triedou a vypočítanou triedou. Riadky, kde trieda a vypočítaná trieda nesedia sú zvýraznené.

Algoritmus C4.5 je implementovaný v triede C45Algorithm. Táto trieda obsahuje metódu BuildTree, ktorá vytvára rozhodovací strom z tréningových dát. Strom je reprezentovaný triedou DecisionNode, ktorá obsahuje atribút, hodnotu a odkazy na potomkov.

```
-----  
Generator rozhodovacich stromov C4.5  
Lietat/nelietat  
Naprogramoval Stefan Kando  
-----
```

```
-----  
Nacitam data o pocasi...
```

```
Data o pocasi uspesne nacistane.  
Nacitanych 732 zaznamov od 01/01/2021 12:00:00 do 01/01/2023 12:00:00.  
-----
```

```
-----  
Nacitam data z kalendara...
```

```
Data z kalendara uspesne nacistane.  
Nacitanych 9046 zaznamov od 01/01/0001 00:00:00 do 17/06/2023 11:00:00.  
-----
```

```
Vyfiltrovaný prienik obsahuje 552 zaznamov,  
z toho 451 pozitivnych a 101 negativnych.
```

```
Zadajte pocet pozitivnych prikladov do trenovacej mnoziny (1-451): 100
```

```
Zadajte pocet negativnych prikladov do trenovacej mnoziny (1-101): 50  
-----
```

```
Generujem strom...  
Strom uspesne vygenerovany za 20 ms.  
-----
```

DIAGNOSTIKA

```
Trenovacie data:  
Presnost: 96.67%  
Spravnost: 96.12%  
Navratnost: 99%  
F1: 97.54%
```

```
Testovacie data:  
Presnost: 76.12%  
Spravnost: 93.22%  
Navratnost: 78.35%  
F1: 85.14%
```

```
Vsetky data:  
Presnost: 81.7%  
Spravnost: 93.97%  
Navratnost: 82.93%  
F1: 88.1%  
-----
```

```
Generujem subory...  
Vytvorených 5 suborov v "data/230510-015144".  
-----
```

```
Generovanie dokoncene.
```

```
Pre vizualizaciu stromu skopirujte obsah "Mermaid.txt" na stranku "https://mermaid.live".
```

```
Stlacte akukolvek klavesu pre ukoncenie...
```

Výstup z chodu programu

3.2. Implementácia C4.5 algoritmu

3.2.1. Triedy

Naša implementácia C4.5 algoritmu používa nasledovné triedy:

- **DecisionNode:** Toto je abstraktná trieda, ktorá predstavuje uzol v strome rozhodnutí. Má metódu `Predict()`, ktorá je abstraktná a musí byť prekrytá v odvodených triedach.
- **DecisionLeaf:** Toto je trieda, ktorá predstavuje koncový uzol (list) v strome rozhodnutí. Obsahuje atribút `Class`, ktorý predstavuje rozhodnutie (klasifikáciu) tohto uzla. Metóda `Predict()` vráti hodnotu tohto atribútu ako reťazec.
- **DecisionSplit:** Toto je trieda, ktorá predstavuje interný (rozhodovací) uzol v strome rozhodnutí. Obsahuje atribúty `AttributeIndex` (index atribútu, podľa ktorého sa má rozhodnúť), `SplitValue` (hodnota, pri ktorej sa má rozhodnúť) a `LeftChild` a `RightChild` (ľavé a pravé dieťa tohto uzla). Metóda `Predict()` zistí hodnotu príslušného atribútu vstupných dát, porovná ju so `SplitValue` a podľa výsledku prenesie predikciu na ľavé alebo pravé dieťa.
- **C45Algorithm:** Toto je hlavná trieda, ktorá obsahuje logiku algoritmu C4.5. Obsahuje metódu `BuildTree()`, ktorá vytvorí strom rozhodnutí zo zadaných tréningových dát.

3.2.2. Implementácia

Naša implementácia C4.5 algoritmu prebieha nasledovne:

1. Ak je počet tréningových príkladov menší ako určitý prah (`MinSamplesSplit` – na predídenie overfittingu) alebo sú všetky príklady rovnakej triedy (`IsPure(data)`), vytvorí sa list s najčastejšou triedou (`MajorityClass(data)`).
2. Inak sa pre každý atribút vypočíta informačný zisk pri delení podľa tohto atribútu. Najlepšie delenie (t.j. to, ktoré prináša najväčší informačný zisk) sa použije na rozdelenie dát do dvoch skupín: `leftData` a `rightData`.
3. Tento proces sa rekurzívne opakuje na ľavej a pravej skupine dát, čím sa vytvoria ľavé a pravé podstromy. Ak sú oba podstromy rovnaké (t.j. oba sú listy rovnakej triedy), vytvorí sa len jeden list s touto triedou.
4. Inak sa vytvorí rozhodovací uzol s indexom najlepšieho atribútu, hodnotou delenia a ľavým a pravým podstromom.

Metóda `FindBestSplit(data)` je zodpovedná za nájdenie najlepšieho atribútu a hodnoty delenia.

Pre každý atribút v dátach sa vypočíta informačný zisk pri delení podľa tohto atribútu, a ten, ktorý prináša najväčší zisk, sa vyberie.

Metóda `GetBestSplitValueForAttribute(data, attributeIndex)` zistí, ktorá hodnota daného atribútu prináša najväčší informačný zisk pri delení dát.

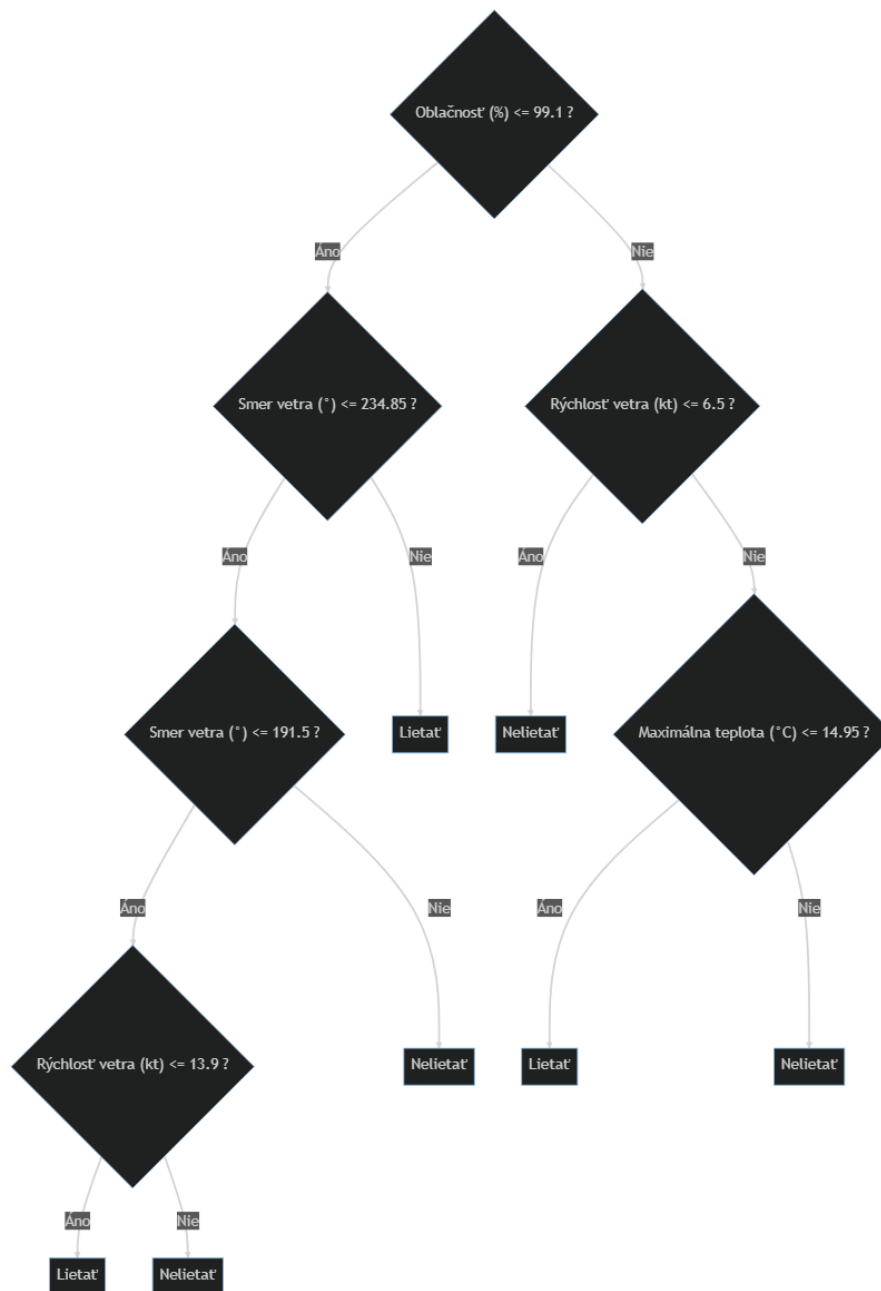
Metóda `CalculateInfoGain(data, attributeIndex, splitValue)` vypočíta informačný zisk pri delení dát podľa daného atribútu a hodnoty. Najprv sa vypočíta pôvodná entropia dát, potom sa dáta rozdelia na základe hodnoty delenia a vypočíta sa nová (vážená) entropia. Informačný zisk je rozdiel medzi pôvodnou a novou entropiou.

Metóda `CalculateEntropy(data)` vypočíta entropiu dát na základe počtu príkladov každej triedy. Entropia je miera neistoty (alebo rozmanitosti) v dátach; ak sú všetky príklady rovnakej triedy, entropia je nulová, ak je polovica príkladov jednej triedy a polovica druhej, entropia je maximálna.

Metóda `Log2(x)` je pomocná funkcia na výpočet logaritmu so základom 2.

Metóda `GetAttributeValueByIndex(data, index)` je ďalšia pomocná funkcia, ktorá vráti hodnotu atribútu tréningových dát na základe jeho indexu. Táto funkcia je potrebná, pretože algoritmus C4.5 pracuje s indexmi atribútov, nie s ich konkrétnymi menami.

4. Vyhodnotenie



Jeden z vygenerovaných stromov znázornený na www.mermaid.live
 Parametre: 30 pozitívnych, 20 negatívnych príkladov v testovacej množine

Testovacie dáta:
 Presnosť: 68.13%
 Správnosť: 87.83%
 Návratnosť: 71.97%
 F1: 79.11%

Všetky dáta:
 Presnosť: 70.65%
 Správnosť: 88.53%
 Návratnosť: 73.61%
 F1: 80.39%

Na mojej implementácii je zaujímavé to, že je možné meniť počet príkladov v pozitívnej a negatívnej testovacej množine. Následne sa z celej množiny dát tieto dáta uniformne náhodne vyberajú, takže aj pri rovnakom počte príkladov v testovacej množine je vygenerovaný strom stále iný.

Na tomto prípade vidíme, že algoritmus nerozumie relevancii porovnávacích atribútov; a to je na strojovom učení najzaujímavejšie. Ak je zamračené, algoritmus by neodporúčal lietať, ak je vietor medzi 191.5 stupňov a 234.85 stupňov, pretože štatisticky keď fúkal takýto vietor, tak bolo lietanie zrušené (jedná sa o bočný vietor v uhle 21 až 64 stupňov, čo pri nie veľmi silnom vetre nebýva nič hrozné).

Najzaujímavejšie je však sledovať veľké stromy, ktoré sú však bohužiaľ príliš veľké na zahrnutie do dokumentácie. Najzaujímavejšie a najlepšie výsledky získavam so 100 pozitívnymi a 50 negatívnymi príkladmi. Pri týchto hodnotách sa všetky metriky pohybujú v hodnotách 80-90%. Pri použití viac dát sú stromy veľké a neprehľadné.

Pri prezeraní dát v Exceli vidíme, že sú dáta zašumené. Pomerne často sa vyskytujú prípady, kedy sa lieta v podmienkach, ktoré sú mimo limit lietania podľa VFR. Toto môže byť spôsobené buď tým, že sa takéto zarezervované lety nevymazali, alebo nám 24 hodinový interval počasia nestačí, keďže sa občas zvykne počasie prudko meniť.

Záver

Z pohľadu pilota som bol z týchto rozhodovacích stromov zaskočený. Je fascinujúce vidieť, ako môžeme pristupovať k rozhodovacím procesom z perspektívy počítača. Napriek tomu, akokoľvek je táto implementácia zaujímavá, súčasne si uvedomujem, že slúži len ako nástroj na podporu rozhodovania a nemôže nahradiť skúsenosť a intuíciu skutočného pilota.

V rámci tohto projektu som sa naučil hodnotiť a interpretovať dáta na úplne novej úrovni. V budúcnosti by som chcel pokračovať v tejto práci a vylepšiť niektoré aspekty tohto modelu. Napríklad by som chcel preskúmať iné metódy strojového učenia, ktoré by mohli byť v tejto situácii ešte efektívnejšie, alebo by som chcel zlepšiť spôsob, akým model berie do úvahy viacero rôznych faktorov, ktoré ovplyvňujú počasie.

Pilotovanie je pre mňa vášnivá záľuba a spojenie toho s umelou inteligenciou - mojou ďalšou záľubou - bola zaujímavá skúsenosť. Tento projekt mi otvoril dvere k novým možnostiam a som nadšený z toho, čo ďalšie môže priniesť spojenie týchto dvoch svetov.

Zoznam použitej literatúry

- [1]. Rozhodovacie stromy, Strojové učenie, KKUI TU Košice, Kristína Machová
- [2]. ICAO Annex 2 to the Convention on International Civil Aviation: Rules of the Air. Dostupné na internete: https://www.icao.int/Meetings/anconf12/Document%20Archive/an02_cons%5B1%5D.pdf