**UNIVERSITETI I PRISHTINËS “HASAN PRISHTINA”**

**FAKULTETI I INXHINIERISË ELEKTRIKE DHE KOMPJUTERIKE**



**PUNIM DIPLOME**

**Tema: Zhvillimi i aplikacionit për detektimin e sëmundjeve të zemrës përmes algoritmit “Random Forest” me Machine Learning**

|  |  |
| --- | --- |
| **Mentori:**  Prof. Dr. Avni Rexhepi | **Kandidatja:**  Teuta Ukshini |

Shtator, 2024

# Abstrakti

Vetë koncepti i Machine Learning bazohet në analizën e një burimi të dhënash nga të cilat nxjerrë rezultate të parashikuara, pra burimi i të dhënave është pjesa kryesore e një projekti që përdorë algoritmet e Machine Learning për të nxjerrë analiza dhe rezultate të parashikuara.

Në shumicën e rasteve, zgjidhja përmes aplikimit të algoritmeve të Data Mining rezulton në kod me qartësi dhe performancë më të lartë. Ideja gjenereale e shumicës së këtyre algoritmeve është që të ndërtohet një model matematikor nga një mostër e të dhënave e pastaj ky model të shfrytëzohet për bërjen e parashikimeve.

Një ndër algoritmet më të përdorura në lëmin e Data Mining është Random Forest, për të cilin do të diskutojmë edhe më gjerësisht në këtë punim. Qëllimi kryesor i këtij punimi është thellimi i njohurive mbi lëmin e Data Mining si dhe përdorimi i algoritmit Random Forest në zgjidhjen e një problemi specifik nga jeta e përditshme.

Në këtë punim, ne do të shqyrtojmë bazat teorike të algoritmit Random Forest dhe do të eksplorojmë zbatimin e tij në skenarë të botës reale. Gjithashtu do të diskutojmë sfidat dhe kufizimet e algoritmit dhe do të propozojmë përmirësime të mundshme dhe drejtime kërkimore në të ardhmen. Për më tepër, ne do të krahasojmë algoritmin Random Forest me algoritme të tjera të njohura të mësuarit të makinës për të theksuar pikat e forta dhe të dobëta të tij.

Ky hulumtim synon të sigurojë një vlerësim gjithëpërfshirës të algoritmit Random Forest dhe potencialit të tij për adresimin e detyrave të ndryshme të parashikimit. Duke analizuar performancën e tij në fusha të ndryshme, ne mund të fitojmë njohuri mbi efektivitetin dhe zbatueshmërinë e algoritmit në kontekste të ndryshme. Kjo analizë do të ofrojë udhëzime të vlefshme për praktikuesit dhe studiuesit që kërkojnë të përdorin algoritmin Random Forest për modelimin parashikues në fushat e tyre përkatëse.

# Abstract

The very concept of Machine Learning is based on the analysis of a data source from which to derive predicted results, so the data source is the main part of a project that uses Machine Learning algorithms to derive analysis and predicted results.

In most cases, solving through the application of Data Mining algorithms results in code with higher clarity and performance. The general idea of most of these algorithms is to build a mathematical model from a sample of data and then use this model to make predictions.

One of the most used algorithms in the field of Data Mining is Random Forest, which we will discuss in more detail in this paper. The main goal of this paper is to deepen the knowledge in the field of Data Mining as well as the use of the Random Forest algorithm in solving a specific problem from everyday life.

In this thesis , we will review the theoretical foundations of the Random Forest algorithm and explore its application in real-world scenarios. We will also discuss the challenges and limitations of the algorithm, and propose potential improvements and future research directions. Additionally, we will compare the Random Forest algorithm with other popular machine learning algorithms to highlight its strengths and weaknesses.

Furthermore, this research aims to provide a comprehensive assessment of the Random Forest algorithm and its potential for addressing various prediction tasks. By analyzing its performance in different domains, we can gain insights into the algorithm's effectiveness and applicability in diverse contexts. This analysis will provide valuable guidance for practitioners and researchers seeking to leverage the Random Forest algorithm for predictive modeling in their respective fields.

# Mirënjohjet

*Dua të shpreh mirënjohjen time më të thellë për mbështetjen dhe prezencën tuaj të vazhdueshme gjatë këtij rrugëtimi. Inkurajimi dhe besimi juaj në aftësitë e mia kanë qenë të paçmueshme*

*Një falenderim i veqantë shkon për mentorin tim të nderuar Prof.Dr Avni Rrexhepi i cili me rekomandimet dhe vërejtjet e tij më ka ndihmuar që të përfundojë me sukses punimin e diplomës.*

*Profesor, ekspertiza dhe përkushtimi juaj kanë pasuruar përvojën time dhe më kanë nxitur të arrij qëllime të reja. Jam shumë mirënjohëse për çdo moment të mentorimit tuaj. Ishte një nder i madh të bashkëpunoj me ju.*

*Ky punim diplomë është një dedikim për dy prindërit e mi, të cilët nuk kanë kursyer asgjë për mua dhe më kanë mbështetur pa kushte. Mirënjohja ime për dashurinë dhe sakrificat e tyre është e pafund.*

*Familjes dhe miqve të mi, faleminderit që jeni gjithmonë aty për mua. Dashuria dhe mbështetja juaj janë burim frymëzimi dhe forcë. Besimi juaj në mua ka luajtur një rol të rëndësishëm në suksesin tim.*

*Kam një fat të jashtëzakonshëm që kam njerëz të tillë të mrekullueshëm në jetën time. Mbështetja juaj ka bërë ndryshimin, dhe nuk do të kisha arritur këtu pa ju. Faleminderit për gjithçka që keni bërë për mua.*

*Me vlerësim të përzemërt!*

**Përmbajtja**

[Abstrakti 2](#_Toc178704845)

[Abstract 3](#_Toc178704846)

[Mirënjohjet 4](#_Toc178704847)

[1 Hyrje 7](#_Toc178704848)

[1.1 Hyrje 7](#_Toc178704849)

[1.2 Motivimi 7](#_Toc178704850)

[1.3 Përshkrimi i problemit 8](#_Toc178704851)

[1.4 Puna e ngjajshme e deritanishme 8](#_Toc178704852)

[2 Algoritmet 9](#_Toc178704853)

[2.1 Koncepti i Algoritmit 9](#_Toc178704854)

[2.2 Historiku i algoritmeve 10](#_Toc178704855)

[2.3 Rëndësia e analizës së algoritmeve 10](#_Toc178704856)

[3 Data Mining: Kuptimi dhe aplikimet në analizën e të dhënave 13](#_Toc178704857)

[3.1 Të dhënat e analizueshme: Ku mund të aplikohet Data Mining 13](#_Toc178704858)

[3.2 Rëndësia strategjike e Data Mining në analizën e të dhënave 14](#_Toc178704859)

[3.3 Algoritmet dhe Data Mining: Roli i tyre në analizën e të dhënave të mëdha 15](#_Toc178704860)

[3.3.1 Shoqërimi (ang. Association) 15](#_Toc178704861)

[3.3.2 Klasifikimi (ang. Classification) 15](#_Toc178704862)

[3.3.3 Grumbullimi (ang. Clustering) 15](#_Toc178704863)

[3.3.4 Parashikimi (ang. Prediction) 15](#_Toc178704864)

[3.3.5 Pemët vendimmarrëse (ang. Decision Trees) 16](#_Toc178704865)

[4 Metodat e ansamblimit (ang. Ensemble Methods) 17](#_Toc178704866)

[4.1 Bagging (Bootstrap Aggregating) – Bagazhimi (Grumbullimi me Boostrap) 18](#_Toc178704867)

[4.2 Boosting 19](#_Toc178704868)

[4.3 Stacking 19](#_Toc178704869)

[5 Pyjet e rastësishme (ang. Random Forest) 20](#_Toc178704870)

[5.1 Çka është Random Forest? 20](#_Toc178704871)

[5.2 Historia e zhvillimit të Random Forest 21](#_Toc178704872)

[5.3 Lidhshmëria në mes Decision Tree dhe Random Forest 22](#_Toc178704873)

[5.4 Si funksionojnë Random Forest? 23](#_Toc178704874)

[5.4.1 Algoritmi Random Forest – Klasifikimi 24](#_Toc178704875)

[5.4.2 Algoritmi Random Forest – Regresioni 25](#_Toc178704876)

[5.5 Hapat e aplikimit të Random Forest në analizën e të dhënave 26](#_Toc178704877)

[5.6 Përdorimi i vetive të Random Forest në parashikimin e të dhënave 28](#_Toc178704878)

[5.7 Decision Tree vs Random Forest 29](#_Toc178704879)

[5.8 Avantazhet dhe Disavantazhet e Random Forest 32](#_Toc178704880)

[5.9 Kur të përdorni Random Forest në jetën reale 33](#_Toc178704881)

[6 Përdorimi i metrikave për zgjedhjen e karakteristikave 34](#_Toc178704882)

[6.1 Mesatarja e uljes së Papastërsisë (ang. Mean Decrease in Impurity (MDI)) 34](#_Toc178704883)

[6.2 Saktësia mesatare e uljes (ang. Mean Decrease Accuracy (MDA)) 35](#_Toc178704884)

[6.3 Matrica e Konfuzionit (ang. Confusion Matrix) 36](#_Toc178704885)

[6.4 Saktësia (ang. Accuracy) 37](#_Toc178704886)

[6.5 Preciziteti (ang. Precision) 38](#_Toc178704887)

[6.6 Rikujtesë (ang. Recall) 39](#_Toc178704888)

[6.7 Rezultati F1 (ang. F1 Score) 40](#_Toc178704889)

[6.8 Sipërfaqja nën lakore (ang. AUC-ROC) 41](#_Toc178704890)

[7 Ndërtimi i një aplikacioni web të bazuar në Random Forest 42](#_Toc178704891)

[7.1 Qëllimi i Web Aplikacionit 42](#_Toc178704892)

[7.2 Zhvillimi i Aplikacionit RF: Struktura dhe Teknikat 43](#_Toc178704893)

[7.2.1 Python 43](#_Toc178704894)

[7.2.2 Visual Studio Code 44](#_Toc178704895)

[7.2.3 Flask 46](#_Toc178704896)

[7.3 Aplikacioni në Jupyter Notebook 48](#_Toc178704897)

[7.4 Zhvillimi i Aplikacionit 54](#_Toc178704898)

[7.4.1 Ekzekutimi i Aplikacionit 55](#_Toc178704899)

[8 Përfundimet 58](#_Toc178704900)

[9 Conclusions 59](#_Toc178704901)

[10 Shtesat 60](#_Toc178704902)

[Referencat 63](#_Toc178704903)

# Hyrje

## Hyrje

Algoritmet luajnë një rol vendimtar në fushën e shkencës kompjuterike, pasi ato janë baza për zgjidhjen e problemeve komplekse në mënyrë efikase dhe efektive. Pyjet e rastësishme (ang. Random Forest), një teknikë e njohur e të mësuarit të makinës (ML-Machine Learning), përdorin një kombinim të pemëve të shumëfishta vendimmarrëse (Decision Tree) për të bërë parashikime dhe përdoren gjerësisht në aplikacione të ndryshme, si njohja e imazheve, zbulimi i mashtrimit dhe parashikimi financiar.

Në këtë punim do të trajtojmë performancën e algoritmeve në ndërtimin e një modeli pyjor të rastësishëm. Pyetja kryesore e kërkimit rrotullohet rreth përcaktimit të algoritmit(eve) që japin rezultatet më të sakta dhe më të besueshme kur zbatohet në një model i pyjeve të rastësishme. Për të adresuar këtë pyetje, do të përshkruajmë metodologjinë e përdorur në studim. Kjo mund të përfshijë një përshkrim të grupit të të dhënave të përdorur, hapat e ndërmarrë para përpunimit, algoritmet e testuara dhe metrikat e vlerësimit të përdorura për të vlerësuar performancën e modeleve. Duke ofruar një metodologji të qartë dhe të detajuar, lexuesit mund të kuptojnë se si është kryer studimi dhe si janë marrë rezultatet. Duke nënvizuar implikimet e hulumtimit, demonstrohet rëndësia e studimit në avancimin e njohurive në fushën e algoritmeve dhe pyjeve të rastësishme.

## Motivimi

Zgjedhja e një teme për projektin tim të diplomës nuk ishte një vendim i lehtë. Doja të zgjidhja diçka që jo vetëm të ngjallte interesin tim, por të kishte edhe aplikime në botën reale. Pas shumë mendimeve dhe konsideratave, përfundimisht vendosa të përqendrohem në algoritmin Random Forest, një teknikë e fuqishme dhe popullore e të mësuarit të makinës që ka fituar vëmendje vitet e fundit për aftësinë e saj për të trajtuar të dhëna komplekse dhe për të prodhuar parashikime të sakta. Një nga arsyet kryesore ishte shkathtësia dhe efektiviteti i algoritmit në një gamë të gjerë aplikimesh. Random Forest është përdorur me sukses në fusha të ndryshme, si financa, kujdesi shëndetësor dhe marketingu, për të zgjidhur probleme komplekse dhe për të marrë vendime të informuara. Aftësia e tij për të trajtuar të dhëna kategorike dhe numerike, për t'u marrë me vlerat që mungojnë dhe për të reduktuar mbipërshtatjen e bën atë një mjet të vlefshëm për shkencëtarët e të dhënave dhe praktikuesit e të mësuarit të makinës. Unë besoj se duke studiuar Random Forest, jo vetëm që do të fitoj njohuri dhe aftësi të vlefshme, por gjithashtu do të bëj një ndikim domethënës në fushën e shkencës së të dhënave dhe inteligjencës artificiale.

## Përshkrimi i problemit

Algoritmet e të mësuarit të makinës, të tilla si algoritmi i pyjeve të rastësishme, janë përdorur për të analizuar grupe të mëdha të dhënash dhe për të identifikuar modele që mund të ndihmojnë në parashikimin dhe parandalimin e sëmundjeve të zemrës. Një nga sfidat kryesore në trajtimin e sëmundjeve të zemrës është kompleksiteti i faktorëve të ndryshëm që kontribuojnë në zhvillimin e saj. Faktorët e rrezikut për sëmundjet e zemrës përfshijnë presionin e lartë të gjakut, kolesterolin e lartë, diabetin, duhanin, obezitetin dhe mungesën e aktivitetit fizik. Identifikimi i individëve që janë në rrezik të lartë për sëmundje të zemrës mund të jetë i vështirë për shkak të ndërveprimit të faktorëve të shumtë dhe ndërveprimeve të tyre.

Algoritmi i pyllit të rastësishëm funksionon duke krijuar një "pyll" të pemëve vendimmarrëse, ku çdo pemë ndërtohet duke përdorur një nëngrup të rastësishëm të të dhënave dhe një nëngrup të rastësishëm të veçorive. Më pas, algoritmi kombinon parashikimet e secilës pemë për të prodhuar një parashikim përfundimtar. Kjo qasje ndihmon në reduktimin e përshtatjes së tepërt (ang. Overfitting) dhe përmirësimin e saktësisë së modelit. Në kontekstin e sëmundjeve të zemrës, algoritmi i pyjeve të rastësishme mund të përdoret për të analizuar faktorë të ndryshëm rreziku dhe ndërveprimet e tyre për të identifikuar individët që janë në rrezik të lartë për zhvillimin e gjendjes. Duke ekzaminuar modelet në të dhëna, algoritmi mund të gjenerojë njohuri që mund të mos jenë të dukshme përmes analizave tradicionale statistikore. Duke shfrytëzuar aftësitë e algoritmeve të mësuarit të makinës, ofruesit e kujdesit shëndetësor mund të kuptojnë më mirë faktorët kompleksë që kontribuojnë në sëmundjet e zemrës dhe të zhvillojnë strategji më efektive për menaxhimin dhe trajtimin e gjendjes.

## Puna e ngjajshme e deritanishme

Në temën e diplomës sime, kam zgjedhur të eksploroj algoritmin Random Forest për të adresuar një problem të ngjashëm me atë të punimit të mëparshëm në lendën Dizajni dhe Analiza e Algoritmeve (DAA). Në atë punim grupor, kemi përdorur algoritmin Decision Tree për të parashikuar personat me nivele të larta të kolesterolit.

Arsyeja për zgjedhjen time të Random Forest si temë e diplomës është e përqendruar në përmirësimin dhe zgjerimin e punës së mëparshme. Edhe pse algoritmi Decision Tree është një zgjedhje e mirë për modelimin e të dhënave të tilla, Random Forest ofron një nivel më të avancuar të kompleksitetit dhe performancës së modelit. Në këtë rast, synimi është të identifikojmë më saktësisht personat që janë në rrezik ose që kanë për momentin sëmundje të zemrës. Me Random Forest do të konsideroj një gamë më të gjerë të karakteristikave dhe ndërveprimeve të mundshme midis tyre për të përmirësuar performancën e modelit në parashikimin e rrezikut të sëmundjes së zemrës.

# Algoritmet

## Koncepti i Algoritmit

Një algoritëm mund të përdoret për shumë gjëra, duke përfshirë përpunimin e të dhënave, kërkimin, renditjen, optimizimin, dhe shumë operacione të tjera. Për shembull, një algoritem i kërkimit mund të jetë i përdorur për të gjetur një element të caktuar në një listë, ndërsa një algoritem i renditjes mund të përdoret për të organizuar një listë në një rradhë të caktuar.

Një algoritëm është një grup hapash ose rregullash të përcaktuara qartë që duhen ndjekur për të zgjidhur një problem ose për të përfunduar një detyrë. Është një proces ose formulë që përdoret për të zgjidhur një grup problemesh bazuar në formula matematikore dhe/ose hapa logjikë. Algoritmet janë shtylla kurrizore e shkencës kompjuterike dhe përdoren pothuajse në çdo aspekt të jetës sonë, nga mënyra se si ne kërkojmë informacion në internet deri në mënyrën se si përdorim mediat sociale. Ata janë forca lëvizëse pas teknologjisë që fuqizon botën moderne.

A diagram of a lamp

Description automatically generatedNjë nga parimet kryesore të algoritmeve është se ato janë krijuar për të qenë efikas, që do të thotë se ata mund të zgjidhin problemet në mënyrën më efektive dhe në kohë të mundshme. Kjo shpesh përfshin zbërthimin e një problemi më të madh në pjesë më të vogla, më të menaxhueshme dhe më pas zgjidhjen e secilës pjesë veç e veç përpara se të kombinohen rezultatet në një zgjidhje përfundimtare. Algoritmet mund të përdoren gjithashtu për të optimizuar proceset dhe për të përmirësuar rezultatet. Për shembull, ato përdoren zakonisht në tregjet financiare për të identifikuar dhe ekzekutuar tregti fitimprurëse, në mjekësi për të diagnostikuar dhe trajtuar sëmundjet dhe në transport për të optimizuar rrugët dhe oraret.

Algoritmet gjithmonë çojnë në një zgjidhje dhe synojnë të jenë sa më efikas në këtë proces.

Njerëzit shpesh përdorin:

* Pseudokodin për hapat e algoritmave
* Flowchart [1] (Diagrami i rrjedhës) për vizualizimin e proceseve

Këto mjete ndihmojnë në vizualizimin e hapave dhe vendimeve brenda algoritmit, duke e bërë më të lehtë kuptimin dhe zbatimin e tij.

*Figura 1 Figurë e thjeshtë e algoritmit*

Në bazë të përkufizimit të mësipërm, përfitojmë katër vetitë themelore të algoritmit:

* Vijimi i udhëzimeve duhet të jetë **i kufizuar**
* Ai duhet të sjellë një **përfundim**
* Udhëzimet duhen të jenë të **zbatueshme**
* Udhëzimet nuk duhet të jenë të **paqarta**

Si përfundim, algoritmet janë në zemër të shkencës kompjuterike dhe luajnë një rol jetik në zgjidhjen e problemeve komplekse dhe optimizimin e proceseve. Ato përdoren në një gamë të gjerë aplikimesh, nga detyrat e përditshme deri te teknologjitë më të avancuara dhe kanë potencialin të kenë një ndikim të rëndësishëm në shoqëri. Derisa teknologjia vazhdon të evoluojë, rëndësia e algoritmeve do të vazhdojë të rritet, duke i bërë ato një fushë thelbësore studimi dhe zhvillimi për të ardhmen. Me kompleksitetin në rritje të problemeve, nevoja për zgjidhje inovative dhe efikase do të bëhet edhe më kritike. Prandaj, kuptimi dhe shfrytëzimi i fuqisë së algoritmeve do të jetë thelbësor në formësimin e së ardhmes së teknologjisë

## Historiku i algoritmeve

Emri "algoritëm" ka marrë një rëndësi simbolike, duke u lidhur me konceptin e zgjidhjes së problemeve dhe llogaritjes. Ai përfaqëson qasjen e strukturuar, hap pas hapi për zgjidhjen e problemeve dhe arritjen e një rezultati të dëshiruar.

Fjala algoritëm është një kombinim i fjalës latine *algorismus* dhe fjala greke *arithmos*, që do të thotë “numër”[2]. Termi algoritëm konsiderohet të ketë ardhur nga emri i dijetarit islam, matematikanit arab Abū Ja’far ʿAbdallāh Muḥammad ibn Mūsā al-Khwārizmī, i cili jetoi në vitet 780-850 në Bagdad. Ai ishte një matematikan që shkroi për numrat indo-arab dhe ishte ndër të parët që e përdori zeron si “pozicion” në notacionin bazë të pozicioneve për numrat. Nga punimi i tij “Hisab al-jabr wa’l-muqabala”, që konsiderohet si libri i parë i shkruar për algjebrën, e ka prejardhjen termi algjebër.

Al-Khwarizmi, i përkthyer në latinisht si “Algoritmi” ose “Algaurizin”, ishte matematikan, astronom dhe gjeograf gjatë perandorise Abaside (Kalifati Abasid, ishte kalifati i tretë islam që pasoi Profetin Muhamed) dhe ishte dijetar, studiues dhe shkencëtar në “Shtëpinë e diturisë/urtësisë” (Dār al-Ḥikma), në Bagdad. Në kohën e renesansës evropiane, ai konsiderohej si zbuluesi origjinal i algjebrës, edhe pse tash dihet se puna e tij bazohej në burime më të vjetra indiane dhe të greqisë antike. Edhe fjalët e mbetura prej punimeve të tij flasin për kontributin e tij në matematikë. Fjala algjebër, që rrjedhë prej fjalës “al-jabr”, që ishte njëri prej dy operacioneve që ai përdori për të zgjidhur ekuacionet kuadratike. Poashtu, termi “Algorism” dhe “Algorithm”, buron prej formës latine të emrit të tij. Punimi i tij në latinisht ishte quajtur “Algoritmi de numero indorum”. [0]

## Rëndësia e analizës së algoritmeve

Zgjedhja e algoritmit më të mirë është një proces i ndërlikuar që shpesh kërkon analizë të detajuar matematikore. Analiza e algoritmeve është procesi i vlerësimit të efikasitetit dhe efektivitetit të algoritmeve në zgjidhjen e problemeve llogaritëse. Ai përfshin matjen e kompleksitetit të kohës dhe hapësirës së një algoritmi për të përcaktuar performancën e tij. Në fushën e shkencës kompjuterike, analiza e algoritmeve është vendimmarrëse për të kuptuar sjelljen e algoritmeve të ndryshme dhe për zgjedhjen e algoritmit më të përshtatshëm për një problem specifik. Si përfundim, **analiza e algoritmeve** është një degë kritike e shkencës kompjuterike që luan një rol themelor në zhvillimin e algoritmeve efikase për zgjidhjen e problemeve llogaritëse. Qëllimi i tij është të kuptojë efikasitetin e algoritmeve, të vlerësojë performancën e tyre, të identifikojë kompensimet e përfshira në hartimin e algoritmeve dhe të optimizojë algoritmet ekzistuese[3]. Kjo fushë është thelbësore për avancimin e shkencës kompjuterike dhe zhvillimin e algoritmeve që mund të trajtojnë sasitë gjithnjë në rritje të të dhënave dhe problemeve llogaritëse në botën e sotme.

**Analiza e Algoritmeve: Pse është e Domosdoshme për Zhvillimin e Efikasitetit**  
Duke shqyrtuar një algoritëm për një problem specifik, ne mund të fillojmë të zhvillojmë njohjen e modelit në mënyrë që lloje të ngjashme problemesh të mund të zgjidhen me ndihmën e këtij algoritmi.

Algoritmet shpesh janë mjaft të ndryshëm nga njëri-tjetri, megjithëse objektivi i këtyre algoritmeve është i njëjtë. Për shembull, ne e dimë se një grup numrash mund të renditen duke përdorur algoritme të ndryshme. Numri i krahasimeve të kryera nga një algoritëm mund të ndryshojë me të tjerët për të njëjtën hyrje. Prandaj, kompleksiteti kohor i këtyre algoritmeve mund të ndryshojë.

Arsyeja më e drejtë për drejtë për të analizuar një algoritëm është të zbulojmë karakteristikat e tij në mënyrë që të vlerësojmë përshtatshmërinë e tij për aplikime të ndryshme ose të krahasojmë atë me algoritme të tjera për të njëjtin aplikim. Për më tepër, analiza e një algoritmi mund të na ndihmojë ta kuptojmë më mirë, dhe të sugjerojmë përmirësime të informuara. Algoritmet priren të bëhen më të shkurtëra, më të thjeshta dhe më elegante gjatë procesit të analizës. Nga kjo mund të shtojmë se dy atributet kryesore të algoritmeve janë:

**Saktësia e Algoritmeve: Rëndësia dhe Strategjitë për Sigurimin e Rezultateve të Besueshme**

Algoritmi duhet të jap rezultat të saktë për të gjitha rastet e mundshme. Mund të përcaktohet në mënyrë matematikore por shpesh nuk dihet. Një algoritëm i saktë duhet të prodhojë vazhdimisht rezultate të sakta dhe të minimizojë gabimet ose mospërputhjet. Në fusha si kujdesi shëndetësor, ku algoritmet përdoren për diagnoza mjekësore dhe rekomandime trajtimi, saktësia është thelbësore për të siguruar mirëqenien e pacientëve.

Arritja e saktësisë në algoritme kërkon testim dhe vërtetim të plotë për të siguruar që algoritmi prodhon rezultate të besueshme në skenarë dhe grupe të dhënash të ndryshme. Për më tepër, zgjedhja e të dhënave të përshtatshme hyrëse, përdorimi i modeleve të sakta matematikore dhe marrja në konsideratë e burimeve të mundshme të gabimit janë thelbësore për të siguruar saktësinë e një algoritmi.

**Analiza e Efikasitetit të Algoritmëve: Krahasimi dhe Vlerësimi**

Efikasiteti është një atribut themelor i një algoritmi sepse përcakton shpejtësinë dhe përdorimin e burimeve të nevojshme për të përmbushur një detyrë. Një algoritëm efikas duhet të jetë në gjendje të përfundojë një detyrë në një kohë të arsyeshme dhe me përdorim minimal të burimeve. Kjo është veçanërisht e rëndësishme në botën e sotme me ritëm të shpejtë, ku përdoruesit presin përgjigje të shpejta dhe kohë minimale pritjeje. Ndersa për të arritur efikasitet, algoritmet shpesh janë të dizajnuara për të minimizuar numrin e hapave të nevojshëm për të përfunduar një detyrë, si dhe sasinë e memories dhe fuqisë përpunuese të nevojshme[4].

* **Analiza kohore:** Sa instruksione mund të ekzekutoj një algoritëm? Për shembull, motorët e kërkimit mbështeten në algoritme efikase për të përpunuar dhe rikuperuar rezultatet e kërkimit brenda disa milisekondave. Kjo u mundëson përdoruesve të gjejnë shpejt informacionin që kërkojnë, duke optimizuar kështu përvojën e tyre të kërkimit. Algoritmet e zhvilluara për këtë qëllim janë të dizajnuara për të minimizuar kohën e ekzekutimit, duke siguruar rezultate të sakta dhe të shpejta
* **Analiza e hapësinore (e hapësirës memoruese):** Për sa hapësirë memoruese ka nevojë algoritmi ?

Për shembull, algoritmet e renditjes (qortimit) si renditja e shpejtë (Quick Sort) dhe bashkimi (Merge) janë krijuar për të rregulluar në mënyrë efikase një listë artikujsh në një renditje specifike, duke reduktuar kompleksitetin kohor të procesit të renditjes.

Si përfundim, rëndësia e analizës së algoritmit nuk duhet të nënvlerësohet. Duke vlerësuar me përpikëri performancën dhe sjelljen e algoritmeve, ne jo vetëm që mund të përmirësojmë softuerin që përdorim, por edhe të kontribuojmë në kuptimin më të gjerë shkencor të llogaritjes. Ndërsa teknologjia vazhdon të evoluojë dhe integrohet në çdo aspekt të jetës sonë, rëndësia e analizës së algoritmit do të vazhdojë vetëm të rritet, duke formësuar të ardhmen e informatikës dhe shoqërisë në tërësi.

# Data Mining: Kuptimi dhe aplikimet në analizën e të dhënave

Data Mining (Gërmimi i të dhënave) është procesi i analizimit të grupeve të mëdha të të dhënave për të zbuluar modele, tendenca dhe njohuri që mund të përdoren për të marrë vendime të informuara biznesi dhe për të nxitur veprime strategjike. Qëllimi kryesor i gërmimit të të dhënave është nxjerrja e informacionit të vlefshëm dhe të zbatueshëm nga sasi të mëdha të dhënash[5]. Ky proces përfshin disa teknika dhe mjete, të tilla si algoritmet e ML, analizat statistikore dhe vizualizimi i të dhënave, për të zbuluar modele dhe marrëdhënie të fshehura brenda të dhënave. Duke vepruar kështu, bizneset mund të fitojnë një kuptim më të thellë të klientëve të tyre, tendencave të tregut dhe efikasitetit operacional, gjë që përfundimisht mund të çojë në një vendimmarrje më të mirë dhe performancë të përmirësuar. Për më tepër, nxjerrja e të dhënave mund të përdoret gjithashtu për të zbuluar mashtrimet dhe anomalitë brenda një grupi të dhënash. Në financë, për shembull, teknikat e nxjerrjes së të dhënave mund të përdoren për të shënuar transaksione potencialisht mashtruese ose për të identifikuar modele të pazakonta në sjelljen e klientit. Kjo mund të ndihmojë institucionet financiare për të zbutur rrezikun dhe për t'u mbrojtur nga aktivitetet mashtruese.

Data Mining është proces i selektimit, eksplorimit dhe modelimit të një sasie shumë të madhe të të dhënave për identifikimin e rregullave ose relacioneve që fillimisht janë të panjohura me qëllim të përfitimit të rezultateve të qarta dhe të dobishme për pronaret e bazës së të dhënave.

## Të dhënat e analizueshme: Ku mund të aplikohet Data Mining

Në parim Data Mining mund të përdoret në të gjitha llojet e depove (angl. repository) të të dhënave digjitale. Natyrisht, nga disa depo të të dhënave më lehtë vijmë deri te rezultatet e kërkuara. Në vazhdim po i përmendim disa depo të dhënave që sot përdoren për mbledhjen dhe ruajtjen e të dhënave.

**Baza e të Dhënave Relacionale (angl. Relational Database)** – paraqet një bashkësi të tabelave, ku secila tabelë ka emër unik në kuadër të bazës së të dhënave. Tabelat përbëhen prej atributeve (kolonave ose fushave) dhe zakonisht ruajnë numër të madh të informatave.

Këto sisteme përdoren në industri të ndryshme si: në sistemin bankar dhe atë financiar, në ERP sisteme, sistemet për përkujdesje ndaj klientëve, në sisteme të ndryshme shkencore etj[6]. Bazat relacionale janë sisteme që më së shumti përdoren për ruajtjen e të dhënave. Në këtë punim, do të fokusohemi kryesisht në këtë lloj të bazave të të dhënave.

**Depo e të Dhënave (angl. Data Warehouse)** – janë depo të të dhënave që shërbejnë për mbledhjen dhe ruajtjen e të dhënave nga burime të ndryshme. Ta marrim një firmë e cila i ka disa sisteme të ndara: Sistemin për menaxhimin financiar, sistemin për menaxhimin e relacioneve me klient dhe sistemin për menaxhimin e shitjeve. Në mënyrë që kompania e caktuar të ketë mundësi të bëjë analiza të mirëfillta, duhet që të ketë shënime të konsoliduara nga të tri sistemet. Me anë të Data Warehouse të dhënat nga tri sistemet mund të konsolidohen në një bazë dhe pastaj mbi këto shënime të bëhen analizat e nevojshme. Pra, Data Warehouse mundëson mbledhjen e të dhënave nga burime të ndryshme, konsolidimin e tyre dhe analizën pa ndikuar në performancën e sistemeve të cilat ju shërbejnë klientëve në kohë reale[7].

**Bazat e të Dhënave Transaksionale** - në përgjithësi bazat transaksionale përbëhen nga fajllat, ku secili rresht paraqet një transaksion. Transaksioni përmban numrin unik dhe listën e elementeve tjera që e përbëjnë transaksionin.

Llojet tjera të bazave – me zhvillimin e mëtutjeshëm të teknologjisë se data bazave, janë zhvilluar sisteme të avancuara të data bazave dhe informacioneve. Sistemet e reja për menaxhimin e shënimeve, bëjnë përpunimin dhe ruajtën e të dhënave të ndryshme si: të dhënat gjeografike (hartat), të dhënat e dizajnerëve, të dhënat multimediale (të dhënat video, audio foto dhe tekst) etj.

## 3.2 Rëndësia strategjike e Data Mining në analizën e të dhënave

Data Mining (Gërmimi i të dhënave) është një fushë shumëdisiplinore që bazohet në statistikat, ML, menaxhimin e bazës së të dhënave dhe ekspertizën specifike të fushës. Është një mjet i vlefshëm për organizatat që kërkojnë të fitojnë njohuri të zbatueshme nga të dhënat e tyre dhe të marrin vendime të bazuara në të dhëna.

Duke analizuar vëllime të mëdha të dhënash, organizatat mund të identifikojnë rreziqet e mundshme dhe modelet e mashtrimit, duke u mundësuar atyre të marrin masa proaktive për të zbutur këto rreziqe dhe për të parandaluar aktivitetet mashtruese[8]. Kjo është veçanërisht e rëndësishme në industri të tilla si financat, sigurimet dhe kujdesi shëndetësor, ku aftësia për të zbuluar dhe parandaluar mashtrimin mund të ketë një ndikim të rëndësishëm në fund.

Për më tepër, nxjerrja e të dhënave është thelbësore për rritjen e efikasitetit operacional dhe produktivitetit. Duke analizuar të dhënat operacionale, organizatat mund të identifikojnë joefikasitetet, pengesat dhe fushat për përmirësim, duke u mundësuar atyre të thjeshtojnë proceset dhe të optimizojnë shpërndarjen e burimeve. Kjo mund të rezultojë në kursime të kostos dhe përmirësim të produktivitetit, duke çuar përfundimisht në një biznes më konkurrues dhe fitimprurës. Derisa përdorimi i të dhënave vazhdon të rritet, data mining do të bëhet më e rëndësishme për t'u mundësuar bizneseve të qëndrojnë konkurrues dhe të nxisin suksesin.

## 3.3 Algoritmet dhe Data Mining: Roli i tyre në analizën e të dhënave të mëdha

Data Mining është një lëmi relativisht e re, por që kërkesa për aplikimin e saj janë aq të mëdha sa që është zhvilluar dhe vazhdon të zhvillohet me ritme shumë të shpejta. Data Mining algoritmet (modelet) janë procedura të definuara mirë, të cilat si madhësi hyrëse i marrin të dhënat që do të analizohen dhe si rezultat i gjenerojnë trendët apo rregullat. Në sistemet e data mining, përdoren një numër i madh i algoritmeve, varësisht nga qëllimi i analizës. Është shumë me rëndësi që të njihen vetit e këtyre algoritmeve në mënyrë që të përdoret modeli adekuat për detyrën e caktuar.

Në vazhdim shkurtimisht do i përshkruajmë disa nga metodat[9] që kanë përdorim më të madh.

### 3.3.1 Shoqërimi (ang. Association)

Ky algoritëm është dizajnuar në mënyrë specifike për përdorim në analiza të tregut. Lidhja në të dhënat e minierave mund të aplikohet në fusha dhe industri të ndryshme. Për shembull, në kujdesin shëndetësor, Association mund të përdoret për të identifikuar modelet e gjendjeve mjekësore që ndodhin bashkë ose për të zbuluar marrëdhëniet midis opsioneve të ndryshme të trajtimit dhe rezultateve të pacientit. Në financë, shoqërimi (angl. Association) mund të ndihmojë në identifikimin e modeleve të transaksioneve mashtruese ose të parashikojë tendencat e tregut të aksioneve bazuar në të dhënat historike.

### 3.3.2 Klasifikimi (ang. Classification)

Klasifikimi përdoret për të analizuar një grup të dhënash të caktuar dhe merr çdo shembull të tij. Ai ia cakton këtë shembull një klase të caktuar. I tillë që gabimi i klasifikimit do të jetë më i vogël. Përdoret për nxjerrjen e modeleve që përcaktojnë klasa të rëndësishme të të dhënave brenda grupit të të dhënave të dhëna.

### 3.3.3 Grumbullimi (ang. Clustering)

Grumbullimi (grupimi) është pothuajse i ngjashëm me klasifikimin, por në këtë grupim bëhen në varësi të ngjashmërive të artikujve të të dhënave. Grupe të ndryshme kanë objekte të ndryshme ose të palidhura. Quhet gjithashtu segmentimi i të dhënave pasi ndan grupe të mëdha të të dhënave në grupe sipas ngjashmërive.

### 3.3.4 Parashikimi (ang. Prediction)

Kjo metodë përdoret për të parashikuar të ardhmen bazuar në tendencat ose grupin e të dhënave të së shkuarës dhe të tashmes. Parashikimi përdoret më së shumti për të kombinuar metoda të tjera të minierave si klasifikimi, përputhja e modelit, analiza e trendit dhe relacioni. Për këtë do të flasim edhe më vonë në shqyrtim të zgjedhjes së metodës përkatese.

* + - **3.3.4.1 Analiza e Klasifikimit (ang. Classification Analysis)**

Teknika e klasifikimit përdoret për të kategorizuar të dhënat, në varësi të ngjashmërive të tyre dhe për të identifikuar klasën.

* + - **3.3.4.2 Analiza e Regresionit (ang. Regression Analysis)**

Analiza e regresionit është një teknikë statistikore për përcaktimin e marrëdhënies ndërmjet një ndryshoreje të vetme të varur (kriterore) dhe një ose më shumë variablave të pavarur (parashikues).

### 3.3.5 Pemët vendimmarrëse (ang. Decision Trees)

Pemët e vendimmarrëse[10] përdoren për të klasifikuar dhe parashikuar të dhënat duke i ndarë ato në segmente bazuar në një sërë vendimesh. Ky proces quhet ndarje rekursive dhe përfshin ndarjen e të dhënave në nënbashkësi gjithnjë e më të vogla bazuar në atribute ose kritere të ndryshme, derisa të plotësohet një prag ose kusht i caktuar.

A diagram of a variety of components

Description automatically generated with medium confidence

*Figura 2 Përmbledhje e algoritmeve në përgjithësi*

# Metodat e ansamblimit (ang. Ensemble Methods)

Metodat e ansamblimit janë teknika që përdoren për të kombinuar parashikimet e disa modeleve individuale për të krijuar një model më të fuqishëm dhe më të stabilizuar. Këto metoda janë të dizajnuara për të përmirësuar performancën e parashikuesve duke adresuar dobësitë dhe variabilitetin e tyre.

Ideja bazë e metodave të ansamblimit është të merret një koleksion modelesh të ndryshme dhe të kombinohen parashikimet e tyre për të bërë një parashikim më të saktë dhe më të forte [11]. Kjo bazohet në konceptin që ndërsa modelet individuale mund të kenë pika të forta dhe të dobëta, duke i kombinuar ato së bashku, dobësitë e tyre mund të zbuten dhe pikat e tyre të forta mund të përforcohen. Si rezultat, metodat e ansamblit janë në gjendje të prodhojnë parashikime më të besueshme dhe të sakta se çdo model individual më vete.

A diagram of a tree

Description automatically generated

*Figura 3 Modelet bazë*

Metodat e ansamblimit bazohen në idenë se shumë nxënës të dobët mund të kombinohen për të krijuar një nxënës të fortë. Një nxënës i dobët është një model që performon pak më mirë se rastësia e rastësishme, ndërsa një nxënës i fortë është një model që performon dukshëm më mirë se shansi i rastësishëm. Duke kombinuar parashikimet e nxënësve të dobët në një mënyrë strategjike, metodat e grupit mund të prodhojnë një nxënës të fortë që është i aftë të bëjë parashikime shumë të sakta.

Ekzistojnë disa lloje të ndryshme të metodave të ansamblit, secila me qasjen e vet unike për kombinimin e modeleve individuale. Një nga llojet më të njohura është metoda e grumbullimit të bagazhit ose bootstrap, e cila përfshin trajnimin e shembujve të shumtë të të njëjtit model në nëngrupe të ndryshme të të dhënave të trajnimit dhe më pas kombinimin e parashikimeve të tyre përmes mesatares ose votimit. Kjo metodë ndihmon në reduktimin e variancës në parashikime dhe përmirëson saktësinë e përgjithshme të modelit.

Një lloj tjetër i metodës së ansamblit është rritja, e cila funksionon duke trajnuar një sekuencë modelesh ku çdo model i ri fokusohet në mostrat që janë klasifikuar gabimisht nga modeli i mëparshëm.

A diagram of a diagram

Description automatically generatedPërveç grumbullimit dhe rritjes, ka edhe lloje të tjera të bashkimit, si grumbullimi, ku parashikimet e modeleve të shumta përdoren si tipare hyrëse për një meta-mësues, dhe pyjet e rastësishme.

Disa nga metodat e ansamblimit [12] më të njohura përfshijnë:

1. **Përforcimi (ang. Boosting)**
2. **Grumbullimi me Bootstrap-Bagazhim (ang. Bootstrap Aggregating => Bagging)**
3. **Grumbullimi (ang. Stacking)**
4. **Dhe të tjera (më pak të përdorura)**

*Figura 4 Metodat e Ansamblimit*

## Bagging (Bootstrap Aggregating) – Bagazhimi (Grumbullimi me Boostrap)

Vlen të përmendet se Bootstrap Aggregating ose Bagging është një teknikë mjaft e thjeshtë por vërtet e fuqishme. Të kuptuarit e konceptit të përgjithshëm të Bagging është vërtet thelbësor për ne pasi është baza e algoritmit Random Forest (RF). Rreth Random Forest do të flasim më shumë në pikën e tretë në vazhdim.

|  |
| --- |
| A diagram of a classifier  Description automatically generated  *Fig 5 Modeli i Bagging (Bootstrap Aggregating)* |

## Boosting

Këto algoritme synojnë të përmirësojnë saktësinë e një modeli të caktuar duke kombinuar fuqinë e modeleve të shumta më të dobëta. Siç u përmend më lart, rritja përdor qasjen sekuenciale. Ideja kryesore e algoritmit përforcues është ndërtimi në rritje i një ansambli duke trajnuar çdo shembull të ri modeli për të theksuar rastet e trajnimit që modelet e mëparshme i kanë klasifikuar gabimisht. Pra, një model po mëson nga gabimet e një tjetri, gjë që nxit mësimin.

|  |
| --- |
| A diagram of a model test  Description automatically generated  *Fig 6 Predikimi në bazë te Boosting* |

## Stacking

Në vend që të mbështetet në një model të vetëm për të bërë parashikime, stacking kombinon parashikimet e modeleve të shumta dhe përdor një meta-model për të mësuar nga parashikimet e kombinuara. Kjo i lejon meta-modelit të kapë pikat e forta dhe të dobëta të modeleve individuale dhe të prodhojë një parashikim më të saktë dhe më të besueshëm.

Stacking është një teknikë pak më interesante. Ai përfshin trajnimin e një modeli (të quajtur Meta Learner) për të kombinuar parashikimet e shumë algoritmeve të tjera të ML (Nxënësit bazë). Së pari, nxënësit bazë trajnohen duke përdorur të dhënat e disponueshme. Së dyti, Meta model është trajnuar për të bërë një parashikim përfundimtar duke përdorur parashikimet e modelit bazë si të dhëna hyrëse.

|  |
| --- |
| A diagram of a method  Description automatically generated  *Fig 7 Parashikimet e modeleve të stacking* |

Si përfundim, metodat e ansamblit janë një mjet i vlefshëm në ML që mund të përmirësojë ndjeshëm performancën e modeleve parashikuese. Duke kombinuar parashikimet e shumë nxënësve të dobët, metodat e grupit mund të krijojnë nxënës të fortë që janë më të saktë dhe më të fortë. Ndërsa fusha e ML vazhdon të rritet, metodat e ansamblimit do të mbeten një teknikë e rëndësishme për përmirësimin e saktësisë dhe besueshmërisë së modeleve parashikuese.

# Pylli i rastësishëm (ang. Random Forest)

## Çka është Random Forest?

Random Forest është një algoritëm i të mësuarit të mbikëqyrur që bazohet në metodën e mësimit të ansamblimit dhe në shumë pemëve vendimmarrëse. Pyjet e rastësishme është një teknikë Bagging, kështu që të gjitha llogaritjet kryhen paralelisht dhe nuk ka ndërveprim midis pemëve të vendimit gjatë ndërtimit të tyre. RF mund të përdoret për të zgjidhur detyrat e Klasifikimit dhe Regresionit.

Random Forest është një algoritëm i përdorur gjerësisht ML i zhvilluar nga Leo Breiman dhe Adele Cutler, i cili kombinon prodhimin e pemëve të shumëfishta vendimesh për të arritur një rezultat të vetëm[13]. Lehtësia e përdorimit dhe fleksibiliteti i tij kanë nxitur miratimin e tij, pasi trajton problemet e klasifikimit dhe regresionit. Emri "*Pyjet e rastësishme*" vjen nga ideja grumbullimin e të dhënave të rastësishme (ang. Random) dhe ndërtimit të Pemëve të shumëfishta Vendimi (Decision Trees). Në përgjithësi, është një algoritëm i fuqishëm ML që kufizon disavantazhet e një modeli të pemëve vendimmarrëse.

Algoritmi është i njohur për aftësinë e tij për të trajtuar të dhëna me dimensione të larta, grupe të dhënash të mëdha dhe është rezistent ndaj përshtatjes së tepërt. Në thelbin e tij, Pyjet e rastësishme (RF) është një koleksion i pemëve vendimmarrëse, ku çdo pemë është trajnuar në një nëngrup të rastësishëm të të dhënave dhe veçorive të trajnimit. Rastësia në të dhënat dhe veçoritë e trajnimit ndihmon në reduktimin e korrelacionit midis pemëve, duke e bërë modelin më të fortë dhe më pak të prirur ndaj përshtatjes së tepërt. Për më tepër, çdo pemë e pyjeve të rastësishme trajnohet në mënyrë të pavarur, dhe parashikimi përfundimtar bëhet duke mesatarizuar parashikimet nga të gjitha pemët, ose duke përdorur një mekanizëm votimi.

|  |
| --- |
| A diagram of a tree  Description automatically generated  *Fig 8 Figurë e thjeshtë e Random Forest Algoritmit* |

## Historia e zhvillimit të Random Forest

Pemët e vendimmarrjes në formën e tyre jo-ansamblike kanë qenë të pranishme në forma të ndryshme të paktën që nga vitet 1950, megjithëse asnjë studiues nuk mund të thuhet se është zbuluesi dhe ka të ngjarë të jenë zbuluar disa herë. Një nga punimet më të hershme për të përmendur një pemë vendimmarrëse është një artikull i 1959 nga statisticieni britanik mbi një algoritëm për të ndërtuar një pemë vendimmarrëse për klasifikimin e organizmave biologjikë[14].

Gjatë dekadave në vijim, pemët vendimmarrëse u rafinuan gradualisht nga komuniteti i statistikave. Algoritmet më të njohura ishin C4.5, ID3 dhe CART. Megjithatë, asnjëra nga metodat nuk përfshinte metoda ansambleje, dhe kështu pemët vendimmarrëse ishin ende të prirura për t'u përshtatur, siç e kemi parë me shembullin e Boston Housing më lart.

Në vitin 1995, studiuesja Hong Kong-Amerikane Ho Tin-Kam zhvilloi algoritmin e parë për pyjet e rastësishme, ndërsa ajo ishte duke punuar në Bell Labs në New Jersey[15]. Ajo përdori metodën e nënhapësirës së rastësishme për të reduktuar korrelacionin midis vlerësuesve, duke ekspozuar çdo vlerësues në një nëngrup të të gjithë grupit të veçorive, por duke përdorur ende të gjithë grupin e trajnimit për të trajnuar çdo vlerësues.

Në vitin 2001, shkencëtari amerikan i kompjuterave Leo Breiman (një nga zbuluesit origjinalë të algoritmit CART) rafinoi algoritmin e ansamblit të Ho për të prezantuar grumbullimin e bagazheve ose bootstrap. Ideja e Breiman ishte të merrte nën-mostra nga grupi origjinal i trajnimit për të trajnuar çdo vlerësues[16].

Megjithëse mekanizmi duket i thjeshtë, ai përfshin shumë forca të ndryshme lëvizëse që e bëjnë të vështirë analizimin. Në fakt, vetitë e tij matematikore mbeten deri më sot kryesisht të panjohura deri më tani, shumica e studimeve teorike janë përqendruar në pjesë të izoluara ose versione të stilizuara të algoritmit. Përpjekje interesante në këtë drejtim janë nga Lin dhe Jeon (2006), të cilët vendosin një lidhje midis pyjeve të rastësishme dhe metodave adaptive të fqinjit më të afërt (shih gjithashtu Biau dhe Devroye, 2010, për rezultate të mëtejshme); Meinshausen (2006), i cili studion konsistencën e pyjeve të rastësishme në kontekstin e parashikimit kuantil të kushtëzuar; dhe Biau et al. (2008), të cilët ofrojnë teorema të konsistencës për versione të ndryshme të thjeshtuara të pyjeve të rastësishëm dhe parashikuesve të tjerë të rastësishëm të ansamblit. Megjithatë, mekanizmi statistikor i pyjeve "të vërteta" të rastësishme nuk është kuptuar ende plotësisht dhe është ende nën hetim aktiv.

Shumica e zbatimeve moderne të algoritmit të pyjeve të rastësishme, siç është zbatimi në bibliotekën Scikit-Learn të Python, bazohen në versionin e Breiman.

## Lidhshmëria mes Decision Tree dhe Random Forest

Përpara se të gërmojmë në ndërlikimet e pyjeve të rastësishme, është thelbësore të kemi një kuptim gjithëpërfshirës të pemëve vendimmarrëse. Pemët vendimmarrëse shërbejnë si themeli mbi të cilin ndërtohet pyjet e rastësishme dhe të kesh një zotërim të fortë të strukturës, funksionalitetit dhe kufizimeve të tyre është thelbësore përpara se të kalosh në modele më komplekse. Para së gjithash, pemët vendimmarrëse janë një mjet i njohur dhe i fuqishëm në fushën e ML dhe analizës së të dhënave. Ato janë të gjithanshme dhe mund të përdoren si për detyra klasifikimi ashtu edhe për regresion. Pemët vendimmarrëse funksionojnë duke ndarë në mënyrë rekursive të dhënat në nënbashkësi më të vogla bazuar në veçori të caktuara dhe përfundimisht duke arritur në një vendim ose parashikim[18]. Ky proces vizualizohet si një strukturë e ngjashme me pemën, me nyje që përfaqësojnë pikat e vendimit dhe degët që përfaqësojnë rezultatet e mundshme.

Megjithatë, pemët vendimmarrëse kanë gjithashtu kufizimet e tyre. Ata janë të prirur për t'u mbivendosur, veçanërisht kur thellësia e pemës nuk kontrollohet siç duhet. Mbi përshtatja ndodh kur modeli bëhet tepër kompleks dhe fillon të kapë zhurmë në të dhëna, gjë që mund të çojë në përgjithësim të dobët në shembujt e paraparë. Për më tepër, pemët vendimmarrëse janë të ndjeshme ndaj ndryshimeve të vogla në të dhëna, të cilat mund të rezultojnë në ndërtimin e pemëve të ndryshme nga grupe trajnimi paksa të ndryshme. Me këtë kuptim të pemëve vendimmarrëse në mendje, tani mund të eksplorojmë pyjet e rastësishme, i cili është një metodë e të mësuarit të grupit që përdor fuqinë e pemëve vendimmarrëse. Pyjet e rastësishme funksionon duke trajnuar pemë të shumta vendimmarrëse në nënbashkësi të rastësishme të të dhënave dhe më pas duke kombinuar parashikimet e tyre për të marrë një vendim përfundimtar. Kjo qasje e ansamblimit ndihmon në zbutjen e çështjeve të mbipërshtatjes që lidhen me pemët individuale vendimmarrëse dhe në përgjithësi çon në performancë më të mirë në të dhënat e padukshme.

Për më tepër, Pyjet e rastësishme trashëgon interpretueshmërinë e pemëve vendimmarrëse, pasi është ende e mundur të analizohen pemët individuale në ansambël dhe të fitohen njohuri në procesin e vendimmarrjes. Pyjet e rastësishme ofron gjithashtu një matje të rëndësisë së veçorive, e cila tregon rëndësinë relative të veçorive të ndryshme në bërjen e parashikimeve.

Si përfundim, të kesh një kuptim të fortë të pemëve vendimmarrëse është thelbësore përpara se të futesh në pyll të rastësishëm. Pemët e vendimmarrjes shërbejnë si blloqe ndërtimi të pyjeve të rastësishme dhe një kuptim i qartë i pikave të forta dhe kufizimeve të tyre është thelbësor për përdorimin efektiv të pyjeve të rastësishme. Duke zotëruar së pari pemët vendimmarrëse, ne mund të vendosim një bazë solide për shfrytëzimin e fuqisë së pyjeve të rastësishme në aplikacione të ndryshme të ML dhe analizës së të dhënave.

## Si funksionojnë Random Forest?

Algoritmi Random Forest - Ky algoritëm është ndërtuar mbi konceptin e të mësuarit në grup, ku kombinon parashikimet e modeleve të shumta individuale për të krijuar një parashikim më të saktë dhe më të fortë. Algoritmi Random Forest funksionon duke krijuar një koleksion të pemëve vendimmarrëse, ku secila pemë trajnohet në mënyrë të pavarur në një nëngrup të rastësishëm të të dhënave. Rastësia në përzgjedhjen e nëngrupit të të dhënave për secilën pemë ndihmon në futjen e diversitetit në pemë, duke i bërë ato më pak të korreluara me njëra-tjetrën dhe më të sakta në parashikimet e tyre.

Për të kuptuar se si funksionon Algoritmi Random Forest, është thelbësore të kuptohet koncepti i pemëve vendimmarrëse. Çdo degë e pemës përfaqëson një vendim të bazuar në një veçori specifike, dhe gjethet e pemës përfaqësojnë vendimin ose rezultatin përfundimtar. Në Algoritmin Random Forest, krijohen një numër i madh pemësh vendimmarrëse, secila duke përdorur një nëngrup të rastësishëm të veçorive dhe të dhënave të trajnimit. Ky proces njihet si bagging (shkurt për grumbullimin e bootstrap-it) dhe ndihmon në reduktimin e përshtatjes së tepërt duke futur diversitet në pemë. Për më tepër, zgjedhja e rastësishme e veçorive përdoret për të siguruar që pemët të mos lidhen me njëra-tjetrën, duke përmirësuar më tej saktësinë e modelit. Pasi të trajnohen pemët individuale vendimmarrëse, Algoritmi Random Forest kombinon parashikimet e tyre për të bërë parashikimin përfundimtar[19]. Në rastin e detyrave të klasifikimit, algoritmi përdor një sistem votimi me shumicë, ku klasa që merr më shumë vota nga pemët individuale zgjidhet si parashikimi përfundimtar. Për detyrat e regresionit, algoritmi mesatarizon parashikimet e pemëve individuale për të bërë parashikimin përfundimtar.

Një nga avantazhet kryesore të Algoritmit Random Forest është aftësia e tij për të trajtuar grupe të mëdha të dhënash dhe hapësira të veçorive me dimensione të larta. Algoritmi është gjithashtu i qëndrueshëm ndaj të dhënave të zhurmshme dhe të jashtme, duke e bërë atë të përshtatshëm për një gamë të gjerë aplikimesh praktike. Për më tepër, Algoritmi Random Forest ofron një masë të rëndësisë së veçorive, duke i lejuar përdoruesit të kuptojnë se cilat veçori janë më me ndikim në bërjen e parashikimeve.

A diagram of a data processing process

Description automatically generated

*Fig 9 Funksioni i pemëve vendimmarrëse në pyjet e rastësishme*

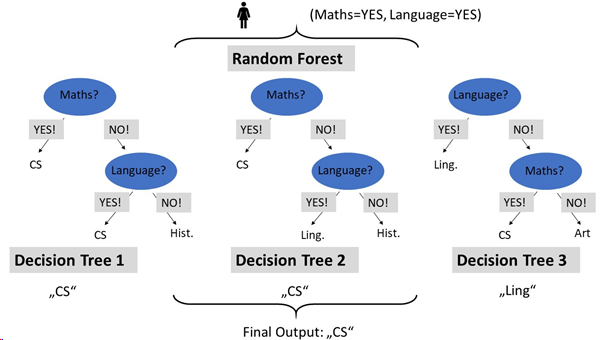
**Ndarja e Algoritmit Random Forest**

* + **Klasifikimi** - Në kontekstin e klasifikimit, Pyjet e rastësishme funksionon duke krijuar një mori pemësh vendimmarrëse, secila e trajnuar në një nëngrup të rastësishëm të të dhënave të trajnimit..
  + **Regresioni** - Në kontekstin e regresionit, Pyjet e rastësishme përdor një qasje të ngjashme për të parashikuar vlera të vazhdueshme dhe jo klasa diskrete.

### Algoritmi Random Forest – Klasifikimi

Klasifikimi i rastësishëm i pyjeve është një algoritëm i fuqishëm dhe popullor i ML që përdoret si për detyrat e klasifikimit ashtu edhe për regresionin. Është një lloj metode e të mësuarit të ansamblimit që kombinon shumë pemë vendimmarrëse për të krijuar një model të fortë dhe të saktë. Në thelb, klasifikimi i rastësishëm i pyjeve është ndërtuar mbi bazën e pemëve vendimmarrëse. Pemët vendimmarrëse janë struktura të ngjashme me pemën që kanë nyje që përfaqësojnë veçori/atribute, degë që përfaqësojnë vendime dhe gjethe që përfaqësojnë rezultatin ose klasifikimin. Megjithatë, pemët individuale vendimmarrëse janë të prirura ndaj përshtatjes së tepërt dhe mund të mos përgjithësohen gjithmonë mirë me të dhënat e padukshme. Këtu hyn në lojë klasifikimi i pyjeve të rastësishme - duke krijuar një grup pemësh vendimmarrëse, ai zbut problemin e mbipërshtatjes dhe përmirëson saktësinë e parashikimit.

Një nga avantazhet kryesore të klasifikimit të pyjeve të rastësishme është aftësia e tij për të trajtuar grupe të dhënash të mëdha me dimensionalitet të lartë. Është rezistent ndaj të dhënave të tepërta dhe të zhurmshme dhe ofron një mekanizëm të integruar të përzgjedhjes së veçorive duke identifikuar variabla të rëndësishëm për klasifikim. Për më tepër, klasifikimi i pyjeve të rastësishme nuk kërkon akordim të gjerë të hiperparametrit dhe është relativisht më pak i ndjeshëm ndaj të dhënave të jashtme. Përdoret zakonisht në fusha të tilla si financa, kujdesi shëndetësor, marketingu dhe shkenca mjedisore për detyra të tilla si segmentimi i klientëve, diagnoza e sëmundjeve, parashikimi i goditjes dhe klasifikimi i mbulesës së tokës. Në financë, mund të përdoret për të parashikuar lëvizjet e çmimeve të aksioneve ose për të zbuluar transaksione mashtruese. Në kujdesin shëndetësor, ai mund të ndihmojë në diagnostikimin e sëmundjeve dhe përcaktimin e prognozës së pacientëve. Në marketing, mund të ndihmojë në shënjestrimin e segmenteve specifike të klientëve për fushata të personalizuara. Në shkencën mjedisore, mund të përdoret për të klasifikuar lloje të ndryshme të mbulesës së tokës nga imazhet satelitore.



*Fig 10 Klasifikimi i i pyjeve të rastësishme*

### Algoritmi Random Forest – Regresioni

Regresioni i pyjeve të rastësishme është një algoritëm i fuqishëm dhe i gjithanshëm i ML që përdoret si për detyrat e klasifikimit ashtu edhe për regresionin. Ai funksionon duke ndërtuar pemë të shumëfishta vendimmarrëse gjatë fazës së trajnimit dhe duke nxjerrë parashikimin mesatar të pemëve individuale për detyrat e regresionit. Ky algoritëm ka fituar popullaritet në fusha të ndryshme si financa, kujdesi shëndetësor dhe marketingu për shkak të aftësisë së tij për të trajtuar grupe të dhënash komplekse dhe për të prodhuar parashikime të sakta.

Një nga avantazhet kryesore të regresionit të pyjeve të rastësishme është aftësia e tij për të trajtuar grupe të dhënash të mëdha me dimensionalitet të lartë. Algoritmet tradicionale të regresionit si regresioni linear ose pemët vendimmarrëse mund të luftojnë me grupet e të dhënave që përmbajnë një numër të madh karakteristikash, pasi ato mund të përshtaten më shumë me të dhënat e trajnimit ose të mos arrijnë të kapin marrëdhëniet komplekse midis veçorive dhe ndryshores së synuar. Regresioni i pyjeve të rastësishme, nga ana tjetër, është i pajisur mirë për të trajtuar të dhëna me dimensione të larta pasi merr parasysh vetëm një nëngrup të rastësishëm të veçorive për secilën pemë, duke reduktuar rrezikun e përshtatjes së tepërt dhe kapjen e veçorive më të rëndësishme për të bërë parashikime të sakta.

Një veçori tjetër kryesore e Regresionit të pyjeve të rastësishme është aftësia e tij për të ofruar pikë të rëndësisë së veçorive. Ky informacion mund të jetë jashtëzakonisht i vlefshëm për të kuptuar marrëdhëniet themelore midis veçorive dhe ndryshores së synuar.

Pavarësisht përparësive të shumta, Regresioni i pyjeve të rastësishme ka gjithashtu disa kufizime. Një kufizim i tillë është mungesa e interpretueshmërisë. Meqenëse Regresioni i pyjeve të rastësishme është një metodë ansamblimi që kombinon shumë pemë vendimmarrëse, mund të jetë sfiduese të interpretosh arsyetimin pas parashikimeve individuale.

|  |
| --- |
| A diagram of a graph  Description automatically generated  *Fig 11 Regresioni i pyjeve të rastësishme* |

## Hapat e aplikimit të Random Forest në analizën e të dhënave

Pyjet e rastësishme u zhvilluan posaçërisht për të trajtuar problemin e variancës së lartë në Pemët vendimmarrëse.

Në një nivel të lartë, në pseudo-kod, algoritmi Random Forests ndjek këto hapa[19]:

* Merrni grupin origjinal të të dhënave dhe krijoni N mostra të grumbulluara me madhësi *n*, me *n* më të vogël se grupi i të dhënave origjinale.
* Trajnoni një Pemë vendimmarrëse me secilën prej N grupeve të të dhënave të grumbulluara si hyrje. Por, kur bëni një ndarje të nyjeve, mos i eksploroni të gjitha veçoritë në grupin e të dhënave. Zgjidh rastësisht një numër më të vogël, M tipare, nga të gjitha veçoritë në grupin e trajnimit. Pastaj zgjidhni ndarjen më të mirë duke përdorur masat e papastërtisë, si Gini Impurity ose Entropy (Papastërtia Gini ose Entropia).
* Përmbledhni rezultatet e pemëve individuale vendimmarrëse në një prodhim të vetëm.
* Mesatarja e vlerave për çdo vëzhgim, të prodhuar nga secila pemë.
* Bëni një shumicë votash në të gjitha pemët, për çdo vëzhgim.

Ndërsa pjesa e Pyjeve të Random Forest i referohet trajnimit të pemëve të shumta, pjesa Random është e pranishme në dy pika të ndryshme në algoritëm. Është rastësia e përfshirë në procesin e Bagazhimit. Por më pas, ju gjithashtu zgjidhni një nëngrup të rastësishëm të veçorive për të vlerësuar ndarjen e nyjeve. Kjo është ajo që garanton se çdo pemë është e ndryshme dhe për rrjedhojë, siguron që secili model të prodhojë një rezultat paksa të ndryshëm. Dhe ndërkohë që mund të mendoni se veçoritë e marrjes së mostrave në mënyrë të rastësishme në çdo ndarje prezanton një hiperparametër tjetër që mund t'ju duhet të sintonizoni, nuk është kështu.

|  |
| --- |
| A diagram of a algorithm  Description automatically generated  *Fig 12 Pseudokodi i RF* |

Për t'i bërë gjërat të qarta, le t'i hedhim një sy algoritmit të saktë të Pyllit të Rastit:

1. Ju keni të dhënat tuaja origjinale D, dëshironi të keni K Decision Trees në ansamblimin tonë. Për më tepër, ju keni një numër N - do të ndërtoni një Pemë derisa të ketë më pak ose të barabartë me N mostra në secilën nyje. Keni një numër F – numri i veçorive që do të zgjidhen rastësisht në secilën nyje të Pemës vendimmarrëse. Tipari që do të përdoret për të ndarë nyjen zgjidhet nga këto veçori F.
2. Çdo gjë tjetër është mjaft e thjeshtë. Random Forest krijon K nëngrupe të të dhënave nga grupi origjinal i të dhënave D. Mostrat që nuk shfaqen në asnjë nëngrup quhen mostra “out-of-bag” (“jashtë grupit”).
3. K pemët ndërtohen duke përdorur vetëm një nëngrup të vetëm. Gjithashtu, çdo pemë ndërtohet derisa të ketë më pak ose të barabartë me N mostra në secilën nyje. Për më tepër, në çdo nyje zgjidhen tiparet F në mënyrë të rastësishme. Njëri prej tyre përdoret për të ndarë nyjen
4. Modelet e trajnuara K formojnë një ansamblim dhe rezultati përfundimtar për detyrën e Regresionit prodhohet duke mesatarizuar parashikimet e pemëve individuale
5. Në foton më poshtë mund të shihni algoritmin Random Forest për Klasifikimin[20].

A diagram of a voting process

Description automatically generated

*Fig 13 Hapat e RF*

## Përdorimi i vetive të Random Forest në parashikimin e të dhënave

Pyjet e studiuara këtu konsistojnë në përdorimin e inputeve të zgjedhura rastësisht ose kombinimeve të inputeve në secilën nyje për të rritur secilën pemë.

Pyjet që rezultojnë japin saktësi që krahasohen në mënyrë të favorshme me Adaboost. Kjo klasë e procedurave ka karakteristikat e dëshirueshme:

* Saktësia e tij është po aq e mirë sa Adaboost dhe ndonjëherë më e mirë.
* Është relativisht i qëndrueshëm ndaj pikave të jashtme dhe zhurmës.
* Është më i shpejtë se sa futja në thasë ose rritja.
* Ai jep vlerësime të brendshme të dobishme të gabimit, forcës, korrelacionit dhe rëndësisë së ndryshueshme.
* Është e thjeshtë dhe e paralelizuar lehtë.

**Karakteristikat e rëndësishme të Pyjet e rastësishme**

* **Diversiteti:** Jo të gjitha atributet/ndryshoret/tiparet merren parasysh gjatë krijimit të një peme individuale; çdo pemë është e ndryshme.
* **Imun ndaj mallkimit të dimensionalitetit:** Meqenëse çdo pemë nuk i merr parasysh të gjitha tiparet, hapësira e veçorive zvogëlohet.
* **Paralelizimi:** Çdo pemë krijohet në mënyrë të pavarur nga të dhëna dhe atribute të ndryshme. Kjo do të thotë që ne mund të përdorim plotësisht CPU-në për të ndërtuar pyje të rastësishme.
* **Ndarja Train-Test:** Në një pyll të rastësishëm, ne nuk kemi nevojë të veçojmë të dhënat për tren dhe testim pasi gjithmonë do të ketë 30% të të dhënave që nuk shihen nga pema e vendimmarrjes.
* **Stabiliteti:** Stabiliteti lind sepse rezultati bazohet në votimin e shumicës/mesatarja.

Amit dhe Geman (1997) rritën pemë të cekëta për njohjen e karaktereve të shkruara me dorë duke përdorur përzgjedhje të rastësishme nga një numër i madh karakteristikash të përcaktuara gjeometrikisht për të përcaktuar ndarjen në secilën nyje[21].

## Decision Tree vs Random Forest

Në fushën e ML, pema vendimmarrëse dhe pyjet e rastësishme janë dy algoritme me përdorim të gjerë. Si pema vendimmarrse ashtu edhe pyjet e rastësishme kanë pikat e forta dhe të dobëta të tyre, dhe ato përdoren në skenarë të ndryshëm bazuar në natyrën e grupit të të dhënave dhe kërkesat specifike të një problemi të caktuar.

Një pemë vendimmarrëse është një strukturë e ngjashme me diagramin e rrjedhës në të cilën nyjet e brendshme përfaqësojnë tiparet e një grupi të dhënash, degët përfaqësojnë rregullat e vendimit dhe nyjet e gjetheve përfaqësojnë rezultatin.

A diagram of a tree

Description automatically generated

*Fig 14 Struktura e Decision Tree*

Nga ana tjetër, Pyjet e rastësishme është një metodë e të mësuarit të ansamblit që përbëhet nga pemë të shumta vendimmarrëse. Ai funksionon duke ndërtuar një mori pemësh vendimmarrëse në kohën e trajnimit dhe duke nxjerrë klasën që është mënyra e klasave (klasifikimi) ose parashikimi mesatar (regresioni) i pemëve individuale. Pyjet e rastësishme përmirëson performancën e një peme të vetme vendimmarrëse duke reduktuar mbipërshtatjen dhe duke rritur saktësinë dhe qëndrueshmërinë e modelit.

A diagram of a process

Description automatically generated

*Fig 15 Struktura e Random Forest*

Një nga ndryshimet kryesore midis pemës së vendimmarrëse dhe Pyjet e rastësishme qëndron në pikat e forta dhe të dobëta të tyre individuale. Nga ana tjetër, Pyjet e rastësishme është më pak i prirur ndaj përshtatjes së tepërt, sepse përdor mençurinë e turmës për të bërë parashikime. Pyjet e rastësishme e arrin këtë duke mesatarizuar parashikimet e pemëve të shumëfishta vendimmarrëse, gjë që ndihmon në reduktimin e variancës dhe përmirësimin e aftësisë së përgjithësimit të modelit.

Një faktor tjetër dallues midis pemës vendimmarrëse dhe Pyjet e rastësishme është interpretueshmëria e tyre[22]. Pema vendimmarrëse është e njohur për interpretueshmërinë e saj të lartë, pasi ajo mund të vizualizohet dhe kuptohet lehtësisht nga njerëzit. Çdo rregull vendimi në një pemë vendimi mund të shpjegohet lehtësisht dhe të përkthehet në njohuri të zbatueshme. Përkundrazi, Pyjet e rastësishme është më pak i interpretueshëm për shkak të natyrës së tij komplekse, pasi përfshin një numër të madh pemësh vendimmarrëse. Ndërsa mund të jetë sfiduese të interpretosh një model pyjesh të rastësishme, ai shpesh e tejkalon pemën vendimmarrëse për sa i përket saktësisë dhe qëndrueshmërisë.

A diagram of a tree and random form

Description automatically generated

*Fig 16 Decision Tree vs Random Forest*

Për sa i përket aplikimeve, pema vendimmarrëse përdoret shpesh në skenarë ku interpretueshmëria dhe lehtësia e të kuptuarit janë vendimmarrëse, si për shembull në diagnozën mjekësore, vlerësimin e kreditit dhe vlerësimin e rrezikut. Nga ana tjetër, Pyjet e rastësishme përdoret gjerësisht në grupe të dhënash komplekse dhe me dimensione të larta, të tilla si në bioinformatikë, njohjen e imazheve dhe parashikimin financiar, ku theksi vihet në arritjen e saktësisë së lartë parashikuese dhe zbutjen e përshtatjes së tepërt.

Të dy algoritmet kanë meritat e tyre dhe zgjidhen bazuar në kërkesat specifike të një problemi të caktuar. Derisa fusha e ML vazhdon të evoluojë, pema vendimmarrëse dhe Pyjet e rastësishme do të vazhdojnë të luajnë një rol vendimtar në zgjidhjen e problemeve të botës reale dhe nxitjen e inovacionit në fusha të ndryshme.

|  |  |
| --- | --- |
| **Decision Tree** | **Random Forest** |
| 1. Pemët vendimmarrëse zakonisht vuajnë nga problemi i përshtatjes së tepërt nëse lejohet të rritet pa asnjë kontroll. | 1. Pyjet e rastësishme krijohen nga nëngrupe të dhënash, dhe rezultati përfundimtar bazohet në renditjen mesatare ose shumicë; prandaj është kujdesur për problemin e mbipërshtatjes. |
| 2. Një pemë e vetme vendimmarrëse është më e shpejtë në llogaritje. | 2. Është relativisht më i ngadalshëm. |
| 3. Kur një grup të dhënash me veçori merret si hyrje nga një pemë vendimmarrëse, ajo do të formulojë disa rregulla për të bërë parashikime. | 3. Pyjet e rastësishme zgjedh rastësisht vëzhgimet, ndërton një pemë vendimmarrëse dhe merr rezultatin mesatar. Nuk përdor asnjë grup formulash. |

*Tabelë 1. Decision Tree vs Random Forest*

**Analogjia e jetës reale e Pyjet e rastësishme**

Një analogji të jetës reale për ta kuptuar më tej këtë koncept. Një student i quajtur X dëshiron të zgjedhë një kurs dhe është i hutuar për zgjedhjen e kursit bazuar në grupin e aftësive të tij. Kështu ai vendos të konsultohet me njerëz të ndryshëm si kushërinjtë e tij, mësuesit, prindërit, studentët e diplomës dhe njerëzit që punojnë. Ai u bën pyetje të ndryshme si pse duhet të zgjedhë, mundësitë e punës me atë kurs, tarifa e kursit, etj. Më në fund, pasi konsultohet me njerëz të ndryshëm për kursin, vendos të ndjekë kursin e sugjeruar nga shumica e njerëzve.

A diagram of a tree

Description automatically generated

*Fig 17 Analogjia e RF në jetën reale*

## Avantazhet dhe Disavantazhet e Random Forest

**Avantazhet:**

* Algoritmi Random Forest eliminon mbipërshtatjen pasi rezultati bazohet në shumicën e votave ose mesataren. Çdo pemë vendimmarrëse e formuar është e pavarur nga të tjerat, duke demonstruar vetinë e paralelizimit
* Për shkak se përdoren përgjigjet mesatare nga një numër i madh pemësh, ai është shumë i qëndrueshëm
* Ai ruan diversitetin duke mos marrë parasysh të gjitha tiparet gjatë krijimit të çdo peme vendimmarrëse, megjithëse kjo nuk është e vërtetë në të gjitha rrethanat
* Nuk ndikohet dimensionaliteti. Hapësira e veçorive është minimizuar sepse çdo pemë nuk i merr parasysh të gjitha vetitë.

Edhe pse Random Forest është një nga algoritmet më efektive për problemet e klasifikimit dhe regresionit, ka disa aspekte që duhet të keni parasysh përpara se ta përdorni.

* Kur krahasohet me pemët vendimmarrëse, ku vendimet përcaktohen duke ndjekur rrugën e pemës, Pyjet e rastësishme është shumë më kompleks.
* Për shkak të kompleksitetit të tij, koha e trajnimit është më e gjatë se për modelet e tjera. Çdo pemë vendimmarrëse duhet të gjenerojë dalje për të dhënat hyrëse të ofruara sa herë që duhet të bëjë një parashikim.

**Disavantazhet:**

Megjithatë, pavarësisht nga avantazhet e tij të shumta, Pyjet e rastësishme ka gjithashtu disa disavantazhe që duhet të merren parasysh kur përdoret ky algoritëm.

* Një nga disavantazhet kryesore të Pyjet e rastësishme është kompleksiteti dhe kostoja llogaritëse e trajnimit dhe akordimit të modelit. Pyjet e rastësishme përfshin ndërtimin dhe kombinimin e një numri të madh të pemëve vendimmarrëse, të cilat mund të jenë llogaritëse të shtrenjta, veçanërisht kur kemi të bëjmë me një grup të madh të dhënash me shumë veçori. Procesi i trajnimit dhe akordimit të modelit kërkon një sasi të konsiderueshme burimesh llogaritëse dhe kohë, gjë që mund të jetë sfiduese për përdoruesit me burime të kufizuara.
* Një tjetër disavantazh i Pyjet e rastësishme është mungesa e interpretueshmërisë së tij. Modeli prodhon parashikime të sakta, por është e vështirë të interpretohen dhe të kuptohen marrëdhëniet midis veçorive të hyrjes dhe parashikimeve të daljes. Kjo mungesë interpretueshmërie mund të jetë një pengesë e rëndësishme kur bëhet fjalë për shpjegimin e rezultateve të modelit për palët e interesuara dhe marrjen e vendimeve të zbatueshme bazuar në parashikimet e modelit[23].

## Kur të përdorni Random Forest në jetën reale

Pyjet e rastësishme përdoret kryesisht për të zgjidhur problemet e klasifikimit.

Mund të dëshironi të provoni Random Forest si algoritmin tuaj të Klasifikimit ML për të zgjidhur probleme të tilla si:

1. Zbulimi i mashtrimit (Klasifikimi) –Tregon se si modelet e thjeshta ML mund të mposhtin rrjetet nervore komplekse në një detyrë të padukshme
2. Vlerësimi i Kreditit (Klasifikimi) – një zgjidhje e rëndësishme në sektorin bankar. Disa banka ndërtojnë rrjete të mëdha nervore për të përmirësuar këtë detyrë. Megjithatë, qasjet e thjeshta mund të japin të njëjtin rezultat
3. Rasti i tregtisë elektronike (Klasifikimi) – për shembull, ne mund të përpiqemi të parashikojmë nëse klienti do ta pëlqejë produktin apo jo
4. Çdo problem klasifikimi me të dhënat e tabelës, për shembull- Kaggle

Në rastin e Regresionit, duhet të përdorni Random Forest nëse:

1. Nuk është problem i serisë kohore
2. Të dhënat kanë një tendencë jolineare dhe ekstrapolimi nuk është vendimtar

Për shembull, Random Forest përdoret shpesh në parashikimin e vlerës (vlera e një shtëpie ose një pako qumështi nga një markë e re).

**Aplikimet e pyjeve të rastësishme**

Pyjet e rastësishme përdoren gjerësisht në fusha dhe industri falë qëndrueshmërisë dhe aftësisë së tyre për t'u marrë me tipare të ndryshme. Me disa parapërpunime, ata janë shumë të mirë në përpunimin e të dhënave që përmbajnë variabla të vazhdueshme dhe kategorike. Me kusht që përgjegjshmëria dhe transparenca, ose shpejtësia, të mos jenë kërkesa të rrepta të çështjes së biznesit, pyjet e rastësishme janë një zgjedhje popullore në shumë aplikacione biznesi dhe kërkimore.

# Përdorimi i metrikave për zgjedhjen e karakteristikave

Random Forest është një algoritëm i njohur të mësuarit të makinës që njihet për aftësinë e tij për të trajtuar grupe të dhënash komplekse dhe për të prodhuar parashikime të sakta. Për të vlerësuar performancën e një modeli Random Forest, është e rëndësishme të analizohen metrika të ndryshme kyçe që mund të ndihmojnë për të kuptuar se sa mirë po performon modeli.

Kur përdorni Random Forest, zgjedhja e veçorive është një hap kritik në procesin e modelimit. Algoritmi funksionon duke krijuar një mori pemësh vendimmarrëse, ku çdo pemë është ndërtuar duke përdorur një nëngrup karakteristikash. Prandaj, zgjedhja e veçorive të duhura do të ndikojë ndjeshëm në saktësinë dhe performancën e modelit.

Ka disa metrika [24] që mund të përdoren për të vlerësuar rëndësinë e veçorive në Random Forest:

## Mesatarja e uljes së Papastërsisë (ang. Mean Decrease in Impurity (MDI))

Në një model pylli të rastësishëm, çdo pemë vendimmarrëse kërkon veçorinë më të mirë për të ndarë të dhënat në secilën nyje. Pema vendimmarrëse i ndan të dhënat në atë mënyrë që të maksimizojë pastërtinë ose homogjenitetin e nëngrupeve që rezultojnë. Papastërtia e një nyje llogaritet duke përdorur metrikë si entropia ose papastërtia Gini, të cilat matin rastësinë ose çrregullimin e të dhënave në atë nyje. Ulja e papastërtisë që arrihet duke ndarë të dhënat në një veçori të veçantë është një masë se sa e rëndësishme është ajo veçori për të bërë parashikime në model.

Për të llogaritur MDI për një veçori, algoritmi i Pyjeve të rastësishme fillimisht ndërton një pyll me pemë vendimmarrëse duke përdorur një nëngrup të të dhënave të trajnimit dhe një nëngrup të rastësishëm të veçorive në çdo ndarje. Më pas, për çdo veçori, algoritmi llogarit uljen mesatare të papastërtisë që shkaktohet nga përdorimi i kësaj veçorie për të ndarë nyjet në të gjitha pemët në pyll.

A comparison of a graph

Description automatically generated

*Fig 18 Mean Decrease in impurity(MDI)*

Në Pyjet e rastësishme (Random Forest), Zvogëlimi Mesatar i Papastërtisë (MDI) mat rëndësinë e një karakteristike duke përllogaritur se sa shumë ajo zvogëlon papastërtinë (p.sh., Gini ose Entropia) në të gjitha pemët.

**Formula për MDI:**

Ku:

* T = Numri total i pemëve.
* N*t* = Seti i nyjeve në pemën t.
* I (X*j*, *n*) = Nëse atributi X*j* përdoret në nyjen n.
* Δ*i*(*n*) = Ulja e papastërtisë në nyjen *n*.

Formula MDI mund të shpjegohet më tej si më poshtë: - Papastërtia e një nyje është një masë se sa mirë janë klasifikuar mostrat në atë nyje. Papastërtia mund të llogaritet duke përdorur metoda të ndryshme, si papastërtia Gini ose entropia.

* Kur një veçori përdoret për të ndarë një nyje, papastërtia e nyjeve të fëmijëve që rezultojnë reduktohet. Ulja e papastërtisë, ΔI(nyja), është një masë se sa ka kontribuar veçoria në performancën e modelit.
* Përqindja e mostrave që arrijnë një nyje, p(nyje), përdoret për të peshuar uljen e papastërtisë. Kjo siguron që MDI të marrë parasysh rëndësinë relative të secilës nyje në modelin e përgjithshëm.
* Shuma e uljeve të peshëzuara të papastërtisë në të gjitha nyjet e modelit të Pyjeve të rastësishme jep vlerën totale të MDI për atë karakteristikë.

## Saktësia mesatare e uljes (ang. Mean Decrease Accuracy (MDA))

Një nga metrikat më të përdorura për vlerësimin e performancës së një modeli Random Forest është saktësia. Saktësia mat përqindjen e rasteve të klasifikuara saktë nga numri i përgjithshëm i rasteve në grupin e të dhënave. Saktësia mesatare e uljes (MDA) është një metrikë thelbësore e përdorur në të mësuarit të makinës për të vlerësuar rëndësinë e veçorive në një model. Është një masë se sa zvogëlohet saktësia e një modeli kur një veçori e veçantë hiqet nga modeli. Sa më e lartë të jetë vlera MDA për një veçori, aq më e rëndësishme është ajo veçori për performancën e modelit.

A graph of a number of data

Description automatically generated with medium confidence

*Fig 19 Mean Decrease Accuracy (MDA)*

MDA llogaritet duke trajnuar fillimisht një model me të gjitha veçoritë e disponueshme dhe më pas duke llogaritur saktësinë e modelit. Më pas, modeli trajnohet sërish, por këtë herë me një veçori të hequr. Dallimi midis saktësisë origjinale dhe saktësisë së trajnur është MDA për atë veçori.

**Formula për MDA:**

Ku:

* T = numri i përgjithshëm i pemëve në pyll.
* = saktësia e pemës t në grupin e të dhënave origjinale (pa ngatërruar veçorinë Xj)
* = saktësia e pemës t pas ndryshimit të rastësishëm të veçorisë Xj.

Duke hequr veçoritë me vlera të ulëta MDA, modeli mund të thjeshtohet, gjë që mund të çojë në kohë më të shpejta të trajnimit, ulje të përshtatjes së tepërt dhe përgjithësim të përmirësuar.

MDA përdoret gjithashtu në kontekstin e pyjeve të rastësishme, një algoritëm i njohur i mësimit të ansamblit. Në pyjet e rastësishme, MDA për çdo veçori llogaritet dhe tiparet me vlerat më të larta MDA konsiderohen më të rëndësishmet në model.

## Matrica e Konfuzionit (ang. Confusion Matrix)

Matrica e konfuzionit është një matricë katrore që ofron një pasqyrë gjithëpërfshirëse të performancës së një modeli. Ai shfaq numrin e pozitivëve të vërtetë, negativëve të vërtetë, pozitivëve të rremë dhe negativëve të rremë për një problem të caktuar klasifikimi.

Vetë matrica e konfuzionit nuk përcaktohet nga një formulë, por përkundrazi është një tabelë që përmbledh performancën e një modeli klasifikimi duke treguar numërimin e pozitiveve të vërteta, negative të vërteta, pozitive të rreme dhe negative të rreme.

Për një problem klasifikimi binar, matrica e konfuzionit [25] është e strukturuar si më poshtë:

|  |  |  |
| --- | --- | --- |
|  | Predicted Positive | Predicted Negative |
| Actual Positive | TP | FN |
| Actual Negative | FP | TN |

*Tabelë 2 Tabela e metrikave*

Ku:

* TP (True Positives) - Numri i rasteve pozitive të parashikuara saktë si pozitive.
* TN (True Negatives - Numri i rasteve negative të parashikuara saktë si negative.
* FP (False Positives) - Numri i rasteve negative të parashikuara gabimisht si pozitive.
* FN (False Negatives) - Numri i rasteve pozitive të parashikuara gabimisht si negative.

Mund të përdorni matricën e konfuzionit për të llogaritur metrika të ndryshme të performancës si saktësia, preciziteti, rikujtimi dhe rezultati F1.

Një nga avantazhet kryesore të matricës së konfuzionit është aftësia e saj për të theksuar llojet specifike të gabimeve të bëra nga modeli. Duke analizuar shpërndarjen e gabimeve, studiuesit dhe praktikuesit mund të fitojnë njohuri të vlefshme për pikat e forta dhe të dobëta të algoritmit të pyjeve të rastësishëm. Ky informacion mund të përdoret më pas për të rregulluar modelin, për të rregulluar hiperparametrat ose për të përfshirë veçori shtesë për të përmirësuar performancën e tij të përgjithshme.

Për më tepër, matrica e konfuzionit mund të përdoret për të krahasuar performancën e modeleve të ndryshme të mësuarit të makinës, duke përfshirë Pyjet e rastësishme, në të njëjtin grup të dhënash.

## Saktësia (ang. Accuracy)

A graph showing a green line

Description automatically generatedSaktësia e një modeli Random Forest është një faktor vendimtar në përcaktimin e efektivitetit dhe besueshmërisë së tij. Ky ndikohet nga disa faktorë, duke përfshirë cilësinë dhe diversitetin e të dhënave të trajnimit, numrin e pemëve vendimmarrëse në ansamblim dhe hiperparametrat e përdorur për të konfiguruar modelin. Një nga avantazhet kryesore të Random Forest është aftësia e tij për të trajtuar një gamë të gjerë të llojeve të të dhënave, duke përfshirë të dhëna numerike, kategorike dhe madje edhe tekste.

*Fig 20 Accuracy*

Formula për llogaritjen e saktësisë në kontekstin e një pylli të rastësishëm (ose ndonjë modeli klasifikimi) është:

Ku:

* TP (True Positives) - Numri i rasteve pozitive të parashikuara saktë nga modeli.
* TN (True Negatives) - Numri i rasteve negative të parashikuara saktë nga modeli.
* FP (False Positives) - Numri i rasteve negative të parashikuara gabimisht si pozitive.
* FN (False Negatives) - Numri i rasteve pozitive të parashikuara gabimisht si negative.

## Preciziteti (ang. Precision)

Precisioni matë proporcionin e parashikimeve të vërteta pozitive nga të gjitha parashikimet pozitive të bëra nga modeli. Është një metrikë e dobishme për të vlerësuar aftësinë e modelit për të identifikuar saktë rastet pozitive. Rritja e numrit të pemëve mund të përmirësojë saktësinë e modelit, pasi zvogëlon gjasat e mbipërshtatjes dhe rrit stabilitetin e parashikimeve.

Formula për llogaritjen e saktësisë në kontekstin e një pylli të rastësishëm (ose çdo modeli klasifikimi) është:

Ku:

* TP (True Positives) - Numri i rasteve pozitive të parashikuara saktë nga modeli.
* FP (False Positives) - Numri i rasteve negative të parashikuara gabimisht si pozitive.

Hiperparametrat e modelit të pyjeve të rastësishme mund të kenë gjithashtu një ndikim të rëndësishëm në saktësinë e tij. Parametrat si thellësia maksimale e pemëve vendimmarrëse, numri minimal i mostrave të kërkuara për të ndarë një nyje dhe numri maksimal i veçorive për t'u marrë parasysh në çdo ndarje, të gjitha mund të akordohen për të optimizuar saktësinë e modelit. Teknika të tilla si trajtimi i vlerave që mungojnë, shkallëzimi dhe normalizimi i veçorive dhe zgjedhja e veçorive më informuese mund të kontribuojnë të gjitha në përmirësimin e saktësisë së modelit.

A graph with a line and a line

Description automatically generated with medium confidence

*Fig 21 Precision*

Në përgjithësi, saktësia e një modeli të rastësishëm pylli është një metrikë komplekse dhe e shumëanshme që varet nga një sërë faktorësh, duke përfshirë cilësinë dhe diversitetin e të dhënave të trajnimit, numrin e pemëve në pyll, hiperparametrat e modelit dhe parapërpunimin. dhe tipar hapat inxhinierik.

## Rikujtesë (ang. Recall)

Recall, i njohur gjithashtu si ndjeshmëria ose norma e vërtetë pozitive, mat përqindjen e rasteve aktuale pozitive që identifikohen saktë nga modeli. Ai përcakton aftësinë e modelit për të identifikuar saktë të gjitha rastet pozitive në grupin e të dhënave.

A graph showing a red line

Description automatically generated

*Fig 22 Recall*

Në kontekstin e Pyjet e rastësishme, metrika e rikujtimit mund të përdoret për të vlerësuar performancën e modelit në identifikimin e rasteve pozitive të një klase të veçantë.

Formula për llogaritjen e rikujtimit (e njohur edhe si ndjeshmëria ose norma e vërtetë pozitive) është:

Ku:

* TP (True Positives) - Numri i rasteve pozitive të parashikuara saktë nga modeli.
* FN (False Negatives) - Numri i rasteve pozitive të parashikuara gabimisht si negative.

## Rezultati F1 (ang. F1 Score)

Rezultati F1 është një metrikë e përdorur gjerësisht në të mësuarit të makinës për të vlerësuar performancën e një modeli klasifikimi. Është veçanërisht i dobishëm për grupet e të dhënave të çekuilibruara, ku numri i rasteve në një klasë është dukshëm i ndryshëm nga numri i rasteve në një klasë tjetër.

|  |
| --- |
| A graph of different colored bars  Description automatically generated  *Fig 23 F1 Score* |

Rezultati F1 është mesatarja harmonike e saktësisë dhe rikujtimit, dhe siguron një masë të balancuar të performancës së një modeli.

Formula për llogaritjen e rezultatit F1

Ku:

* Precision =
* Recall =

Rezultati i F1 varion nga 0 në 1, ku 1 tregon performancën perfekte dhe 0 tregon performancën më të keqe të mundshme.

## Sipërfaqja nën lakore (ang. AUC-ROC)

AUC-ROC është një grafik që tregon shkëmbimin midis normës së vërtetë pozitive (TPR) dhe normës false pozitive (FPR) të një klasifikuesi binar.

Metrika AUC-ROC llogaritet duke integruar sipërfaqen nën lakoren ROC, e cila varion nga 0 në 1. Një vlerë AUC-ROC prej 0.5 tregon se performanca e modelit nuk është më e mirë se hamendja e rastësishme, ndërsa vlera 1 tregon një perfekt klasifikues që mund të dallojë midis klasave pozitive dhe negative pa asnjë gabim.

A graph of a curve

Description automatically generated

*Fig 24 AUC-ROC*

AUC-ROC (ang. Area Under the Curve - Receiver Operating Characteristic) është një matje e performancës për problemet e klasifikimit në cilësime të ndryshme të pragut. Ndërsa nuk ka asnjë formulë të vetme për AUC, ajo përfshin disa koncepte kryesore:

1. Lakorja ROC: Lakorja ROC paraqet normën e vërtetë pozitive (TPR, ose Recall) kundrejt normës false pozitive (FPR) në nivele të ndryshme. TPR dhe FPR përcaktohen si:

* **True Positive Rate (TPR)**:
* **False Positive Rate (FPR)**:

1. Lakorja AUC: AUC është zona nën lakorën ROC. Ai përcakton aftësinë e përgjithshme të modelit për të bërë dallimin midis klasave pozitive dhe negative. Vlerat AUC variojnë nga 0 në 1, ku:

* AUC = 0.5: Modeli nuk ka aftësi diskriminuese (ekuivalente me supozimin e rastësishëm).
* AUC = 1: Modeli dallon në mënyrë të përkryer midis klasave pozitive dhe negative.

Për të llogaritur AUC, zakonisht përdorni metoda numerike si rregulli trapezoid për të vlerësuar zonën nën lakore, ose mund të përdorni funksione të integruara nga bibliotekat si Scikit-learn në Python.

# Ndërtimi i një aplikacioni web të bazuar në Random Forest

Sëmundjet e zemrës janë një çështje e përhapur shëndetësore që prek miliona njerëz në mbarë botën. Sipas Organizatës Botërore të Shëndetësisë, sëmundjet e zemrës janë shkaku kryesor i vdekjeve në nivel global, duke llogaritur se çdo vit janë 17.9 milionë vdekje. Duke pasur parasysh pasojat e rënda të sëmundjeve të zemrës, zbulimi i hershëm dhe parandalimi janë vendimmarrëse në uljen e barrës së kësaj gjendjeje[26].

Një mënyrë për të adresuar këtë çështje është përmes zbatimit të projekteve të modelimit parashikues që synojnë të parashikojnë nëse një person është në rrezik për të zhvilluar sëmundje të zemrës. Duke analizuar një sërë faktorësh si mosha, gjinia, historia familjare, presioni i gjakut, nivelet e kolesterolit dhe zakonet e jetesës, këto projekte mund të ndihmojnë profesionistët e kujdesit shëndetësor të identifikojnë individët që kanë më shumë gjasa të zhvillojnë sëmundje të zemrës në të ardhmen. Ky informacion më pas mund të përdoret për të përshtatur ndërhyrjet dhe trajtimet parandaluese për të zvogëluar rrezikun e sëmundjeve të zemrës dhe për të përmirësuar rezultatet për pacientët.

## Qëllimi i Web Aplikacionit

Duke zbatuar një model parashikimi duke përdorur algoritmin Random Forest, profesionistët e kujdesit shëndetësor mund të identifikojnë individë që janë në rrezik të lartë të sëmundjeve të zemrës dhe të ndërhyjnë herët për të parandaluar shfaqjen e gjendjes.  
Një projekt i tillë është zbatimi i një modeli parashikimi duke përdorur algoritmin Random Forest për të parashikuar nëse një person ka një sëmundje të zemrës apo jo. Qëllimi i këtij projekti është të ndihmojë në zbulimin e hershëm dhe parandalimin e sëmundjeve të zemrës, të cilat janë shkaku kryesor i vdekjeve në mbarë botën.

|  |  |
| --- | --- |
| Të dhënat (dataset) që kam përdorur në këtë aplikacion janë marrë nga uebfaqja ***Kaggle*** [27].  Dataset-i përbëhet nga disa atribute të predikimit mjekësor dhe një atribut të synuar (angl. target). Ai përfshin gjithsej 1026 rekorde, duke ofruar një bazë të mirë për analizën dhe modelimin e të dhënave.  Atributet përfshijnë:   * Age - Mosha * Sex - Gjinia * Cp (Chest Pain) - Dhimbja e Kraharorit * Trestbps (Resting Blood Pressure) - Presioni i Gjakut * Chol (Cholesterol) - Kolesteroli * Fbs (Fasting Blood Sugar) - Sheqeri në Gjak * Thalach (Maximum Heart Rate Achieved) - Ritmi Maksimal i Zemrës | A screenshot of a computer  Description automatically generated  *Fig 25 Dataseti me 1026 të dhëna* |

## Zhvillimi i Aplikacionit RF: Struktura dhe Teknikat

Aplikacioni është zhvilluar në gjuhën e programimit Python më saktësisht në kornizën Flask. Duke përdorur editorin Visual Studio Code, ka filluar puna e trajnimit të të dhënave me ndërfaqen e aplikacionit.

Kombinimi i Python, Flask dhe Visual Studio Code përfaqëson një treshe të fuqishme mjetesh për ndërtimin e aplikacioneve moderne në web. Duke shfrytëzuar pikat e forta të secilës teknologji, zhvilluesit mund të krijojnë aplikacione të fuqishme dhe të besueshme që plotësojnë nevojat e përdoruesve të tyre.

### Python

Python është një gjuhë programimi e interpretuar, e orientuar nga objekti, e nivelit të lartë me semantikë dinamike e zhvilluar nga Guido van Rossum. Fillimisht u publikua në vitin 1991. I projektuar për të qenë i lehtë dhe argëtues, emri “*Python*” është një shenjë e grupit britanik të komedisë Monty Python [28].

* Python mund të përdoret në një server për të krijuar aplikacione në internet.
* Python mund të lidhet me sistemet e bazës së të dhënave. Ai gjithashtu mund të lexojë dhe modifikojë skedarët.
* Python mund të përdoret për të trajtuar të dhëna të mëdha dhe për të kryer matematikë komplekse.
* Python mund të përdoret për prototipimin e shpejtë, ose për zhvillimin e softuerit të gatshëm për prodhim.

Përdoret për:

* Zhvillimi i webit (nga ana e serverit),
* Zhvillimin e softuerit,
* Matematikë,
* Skriptimi i sistemit.

A chart of a program

Description automatically generated with medium confidence

*Figura 26 Karakteristikat e Python*

*Pse Python?*

* Python punon në platforma të ndryshme (Windows, Mac, Linux, Raspberry Pi, etj).
* Python ka një sintaksë të thjeshtë të ngjashme me gjuhën angleze.
* Python ka sintaksë që lejon zhvilluesit të shkruajnë programe me më pak rreshta se disa gjuhë të tjera programimi.
* Python funksionon në një sistem interpretues, që do të thotë se kodi mund të ekzekutohet sapo të shkruhet. Kjo do të thotë që prototipi mund të jetë shumë i shpejtë.
* Python mund të trajtohet në një mënyrë procedurale, një mënyrë të orientuar nga objekti ose një mënyrë funksionale.

*Sintaksa e Python*

* Python është dizajnuar për lexueshmëri më të lehtë dhe ka disa ngjashmëri me gjuhën angleze me një ndikim nga matematika.
* Python përdor rresht të ri (angl. new lines) për të përfunduar një komandë, në krahasim me gjuhët e tjera të programimit të cilat shpesh përdorin pikëpresje ose kllapa.
* Python përdor hapësirat për me i definuar loop-at, funksionet dhe klasat. Në dallim me gjuhët tjera programuese të cilat përdorin kllapat {}, (), për këto qëllime.

Python mbështet modulet dhe paketat, të cilat inkurajojnë modularitetin e programit dhe ripërdorimin e kodit. Interpretuesi Python dhe biblioteka e gjerë standarde janë të disponueshme në formë burimore ose binare pa pagesë për të gjitha platformat kryesore dhe mund të shpërndahen lirisht.

### Visual Studio Code

Visual Studio Code (VSC) është një redaktues i njohur dhe i gjithanshëm i kodit burimor i zhvilluar nga Microsoft. Është një mjet falas, me burim të hapur që u ofron zhvilluesve një ndërfaqe shumë të personalizueshme dhe miqësore për përdoruesit. VSC mbështet gjuhë të ndryshme programimi dhe ofron një gamë të gjerë funksionesh, duke e bërë atë një mjet thelbësor për zhvilluesit e të gjitha niveleve të aftësive.

|  |  |
| --- | --- |
| A screenshot of a computer program  Description automatically generated | A screen shot of a computer screen  Description automatically generated |
| *Figura* ***27*** *Editori* ***Visual Studio Code*** | |

Nga pamja e ekranit e treguar më sipër, mund të shihni Demon e projektit të sapokrijuar dhe follderin e paketave për menaxhimin e tyre.

Kështu duket projekti në VS Code  
  
/random\_forest

│

├── app.py # Flask app code

├── heart\_disease\_model.pkl # Trained model file

├── /templates

│ ├── index.html # Your main page

│ ├── predict.html # Form for prediction

│ └── result.html # Page to show the prediction result

└── /static # (Optional) for CSS, JS, or images

Kodin hap pas hapi do të shqyrtohet më vonë tek një nën-kapitull tjetër për ekzekutim.

**Karakteristikat – VS Code**

Visual Studio Code ofron karakteristika të shumta, përfshirë:

* redaktimin inteligjent të kodit, ngjyrimin sintaksor,
* integrimin me Git,
* mbështetje për shtesa që zgjasin funksionalitetin e tij.

Gjithashtu, ka një terminal të integruar që lejon ekzekutimin e komandave direkt brenda mjedisit të zhvillimit.

**Personalizimi dhe Shtesat**

Një nga përparësitë e Visual Studio Code është aftësia për t'u personalizuar. Përdoruesit mund të instalojnë shtesa për funksionalitete të reja, si përkrahja për framework-e të ndryshme ose mjete për debugging, që ndihmojnë në optimizimin e procesit të zhvillimit.

**Përdorimi i VS Code**

Visual Studio Code është një mjet i fuqishëm për zhvillimin e softuerit që ofron një ambient të integruar pune për programuesit. Ai mbështet shumë gjuhë programimi dhe ka një ndërfaqe përdoruesi të thjeshtë dhe të intuitshme, duke e bërë të lehtë për çdo nivel zhvilluesi.

### Flask

Flask është një kornizë (angl. Framework) e njohur në internet për zhvillimin e aplikacioneve në ueb duke përdorur Python. Është i njohur për thjeshtësinë, fleksibilitetin dhe lehtësinë e përdorimit, duke e bërë atë një zgjedhje të shkëlqyer si për fillestarët ashtu edhe për zhvilluesit me përvojë. Flask është një mikrokornizë, që do të thotë se ka një bërthamë të vogël dhe të lehtë për t'u zgjeruar që u lejon zhvilluesve të shtojnë vetëm veçoritë që u nevojiten për projektin e tyre specifik.

Një kornizë(framework) është një mjedis ose platformë e strukturuar që ofron mbështetje themelore për zhvillimin e aplikacioneve softuerike. Zakonisht përfshin kode të parashkruara, biblioteka, mjete dhe udhëzime për të ndihmuar zhvilluesit të ndërtojnë dhe organizojnë aplikacionet e tyre në mënyrë më efikase.

*Ose*

Një kornizë “është një bibliotekë kodesh që e bën jetën e një zhvilluesi më të lehtë kur ndërton aplikacione ueb të besueshme, të thjeshta dhe të mirëmbajtura”, duke ofruar kode të ripërdorshme për operacionet e caktuara.

Përveq Flask, Python ka edhe korniza tjera si Tornado, Piramida dhe Django. Shpesh Flask referohet si mikrokornizë për thejshtësinë që e ka.

|  |
| --- |
| A screenshot of a computer  Description automatically generated  *Figura* ***28*** *Framework(kornizat) e Python* |

Shpeshherë hasim në korniza të ndryshme në të cilat rrallë e gjejmë vetën, por mund të shihni përhapjen e kornizës Flask dhe të vërehet lehtësisht pse është aq e përdorur.

|  |
| --- |
| A screenshot of a computer  Description automatically generated  *Figura* ***29*** *Përdorimi i Flask në krahasim me kornizat tjera* |

**Karakteristikat e Flask**

* I lehte dhe modular
* Lehtë për t'u mësuar dhe përdorur
* Ideale për aplikime të vogla dhe të mesme
* Ekosistem i gjerë i zgjerimeve

Flask ndryshe nga korniza Django dhe kornizat tjera është më qartë, është më e lehtë për t’u mësuar dhe ka më pak kod nëse dëshirojmë të zbatojmë në krijimin e një ueb aplikacioni.

P.sh për të krijuar një aplikacion “Hello World” duhen vetëm disa rreshta kod, si më poshtë.

Fillojmë duke importuar Flask nga paketa flask që është në çdo IDE python:

![A computer code with text

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RCORXhpZgAATU0AKgAAAAgABAE7AAIAAAAJAAAISodpAAQAAAABAAAIVJydAAEAAAASAAAQdOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEZpdG9yZSBIAAAAAeocAAcAAAgMAAAIZgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAARgBpAHQAbwByAGUAIABIAAAA/+ELG2h0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8APD94cGFja2V0IGJlZ2luPSfvu78nIGlkPSdXNU0wTXBDZWhpSHpyZVN6TlRjemtjOWQnPz4NCjx4OnhtcG1ldGEgeG1sbnM6eD0iYWRvYmU6bnM6bWV0YS8iPjxyZGY6UkRGIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iLz48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOnhtcD0iaHR0cDovL25zLmFkb2JlLmNvbS94YXAvMS4wLyI+PHhtcDpDcmVhdGVEYXRlPjIwMjItMDItMThUMTU6MTg6MzQuOTk1PC94bXA6Q3JlYXRlRGF0ZT48L3JkZjpEZXNjcmlwdGlvbj48cmRmOkRlc2NyaXB0aW9uIHJkZjphYm91dD0idXVpZDpmYWY1YmRkNS1iYTNkLTExZGEtYWQzMS1kMzNkNzUxODJmMWIiIHhtbG5zOmRjPSJodHRwOi8vcHVybC5vcmcvZGMvZWxlbWVudHMvMS4xLyI+PGRjOmNyZWF0b3I+PHJkZjpTZXEgeG1sbnM6cmRmPSJodHRwOi8vd3d3LnczLm9yZy8xOTk5LzAyLzIyLXJkZi1zeW50YXgtbnMjIj48cmRmOmxpPkZpdG9yZSBIPC9yZGY6bGk+PC9yZGY6U2VxPg0KCQkJPC9kYzpjcmVhdG9yPjwvcmRmOkRlc2NyaXB0aW9uPjwvcmRmOlJERj48L3g6eG1wbWV0YT4NCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgPD94cGFja2V0IGVuZD0ndyc/Pv/bAEMABwUFBgUEBwYFBggHBwgKEQsKCQkKFQ8QDBEYFRoZGBUYFxseJyEbHSUdFxgiLiIlKCkrLCsaIC8zLyoyJyorKv/bAEMBBwgICgkKFAsLFCocGBwqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKv/AABEIAT4B8wMBIgACEQEDEQH/xAAfAAABBQEBAQEBAQAAAAAAAAAAAQIDBAUGBwgJCgv/xAC1EAACAQMDAgQDBQUEBAAAAX0BAgMABBEFEiExQQYTUWEHInEUMoGRoQgjQrHBFVLR8CQzYnKCCQoWFxgZGiUmJygpKjQ1Njc4OTpDREVGR0hJSlNUVVZXWFlaY2RlZmdoaWpzdHV2d3h5eoOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4eLj5OXm5+jp6vHy8/T19vf4+fr/xAAfAQADAQEBAQEBAQEBAAAAAAAAAQIDBAUGBwgJCgv/xAC1EQACAQIEBAMEBwUEBAABAncAAQIDEQQFITEGEkFRB2FxEyIygQgUQpGhscEJIzNS8BVictEKFiQ04SXxFxgZGiYnKCkqNTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqCg4SFhoeIiYqSk5SVlpeYmZqio6Slpqeoqaqys7S1tre4ubrCw8TFxsfIycrS09TV1tfY2dri4+Tl5ufo6ery8/T19vf4+fr/2gAMAwEAAhEDEQA/APb/AA4fmuB/u/1rcrC8Of6y4+i/1rdoGwooooEFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAYXhz/AFlx9F/rW7WF4c/1lx9F/rW7QNhRRRQIKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAwvDn+suPov8AWt2sLw5/rLj6L/Wt2gbCiiigQVQ1bWrHRLZZtRn8lXbah2M2WxnHANX6bKCYXA5O01nV5uR8js/v/wAhrc4jwrqNpqt/bzT+INQm1KTc72QZlgHXjbtxgD3rYn8aafBJOi219O1tI6TiGDf5QU4LMc4A4Pv7Vb8LQy2/hawiuI3ikSLDI6lSpyeoNR+HrRo4NUS6tyom1CdtsiY3oTweeoIripwrQjGEWrtN3s99O7fz1JW1yS/8SWVjDbMqz3cl0u+GG1j3yOuM5x6VGnivT30m5vylwgtWCTwyR7ZIySAMgn39aq6ks+i+IYNSttNmu7P7J9lMdnGGeLDZGF9O3HpWbe2d9qWma3qDafPAb0wJDbMmZGVGGWZR0Pt6CnOtVTly766WenZ363/XyYzRXx3pTSbfJvQfMCtugxtU42ueeFORg9eeldLXFapp108niIxWczea9p5W2IneF2528c4xzjpXbVth51JXVT8rd/8AIRkeIPEeneHLMS6lcCFpQwhBjZt7AZx8oOK5PwRqVhq2oW08nijVLzVmiaWaxZmW3Ukcjbt28Z7HrXd36s+m3KoCzNC4AAyScGqHhWGW28I6XDcRvFLHaorxupVlOOhB6Gupbt/11G9kv66GPaWU+lfEGKBdV1K7gurOaZoru5LojB1xtHAAGa2ta16HRBCHs769lm3bIbKAyuQMZOOgHI/OuVuPEc7eMbbUl8NeIDBBayW7AWByWZlII5xjj1q340fVTqVksS60NLaJjIdFUef5ueAx7Lil9lf13H9p/wBdjd07xLp+paRcaiplt4rUsLhLiMo8JUZIZfpUGk+LLbV75bZNO1O18xS0Mt3amOOYDn5Wz6c84rE8J2V5aaBrwvtNvrgzTsy2984aW4QxrwW6Enp+nameHJL9detodIstds9LUMLmDVkHlxrt+UREkt17ZxVdfkT0NK31tNN1XxNcandSfZrWaERozFsZiB2ovqT2HU10dncNd2cU7W81sZF3eVOAHX6gEgH8a4m60HUJfE+q6xZwM1xZ3Uc1vBOp8q6HlAMBnjd6MOhrtNPvBf2EVyIJ7fzFyYp4yjoe4INJbfd+Q3uWK5nTruy1DxlJP9k1S1vhaldt0gSNkDdQM5PJ+ldNXGnXJX8VR340HWhCLUwEGzOdxfOeuMVx4iajUp3fV9PJ/wBfMH8L/rqauo+MNO0zUJbS5S5LwlPMZI8qisB8xOeByB9T3oTxfp76Xd3/AJdykdo6LIrxhWwxG1gCehzn6Vka5p91O3iNorOaQzG0ERWIneAV3Y45x3xUfiGCWCx8TSvBIsckloYyUIDgbAcevpXNKvXhzN7K/T1/y/EOpvaf4qsdQ1BLRIbuEzAmCWeEok4HUoe/6U2+8W2VleTW6W17dm3/ANfJawb0hPoxzVOO4ufEGsaa6aXeWNtYuZXkvI/LLHaQFUd+vWo7e9vvD01/Zvo15eme5kngmtk3I4c5Ac/w46VbrVOXfTWz5XrtbTzu/u8wRr3fiTTrLT7S+llZrW6bakqLkDgnkde2PXNRWnii2uxcqllqEc9vF5v2eS32ySL6oueazI9HurPSvDttLC0kkF6JJgi7hHkOecdACQM11XkRfaPP8pPO27PM2jdtznGeuM9q3g60223ZbWt/dT38mxdjldK8YTz3F0lxpOrSj7UUj2WgxCvGFfngjqetdbXLWl9c6HrGo29xpN/cR3V2Zop7WHzEwwHU54xiupowkpOnacrtb6WsPqzH8XX9zpnhPULyxl8q4hj3I+0NtOQOh4rM8OzXc2qL5vjez1pBGS1pDBCrfXKMTxV/xrbzXfgzUoLWGSeV4gFjjUszfMOgHWrtjoml6e4msdMs7WYrtLw26o2PTIGa61uD2X9djAvrKfTvHekTxarqUkd/PN5lrLckwqBESAqdhn1zW7rOtQ6LbRyy291dPI+yOG0hMkjnGeB9BXLa9r07eKdNli8Pa5LHplxN5jx2RZZAUKAoc8jPPbirXi+51We10uWyh1iKwmy14mnIBdplQUXHbnOaXTQf2vl/mbWjeI7PWobl447i0ktTieC7i8uSLjIJHoRVTTvGVnqd/FbwWGpxxTkiC8mtCsEvGflb3A4yBWP4Ks721utdmubPVGWZIjCNUcNLMArcFsY9sdsjNV9Ha+i1mzh8P6Zr2nWwlH2q21BB9lSPnd5ZYkg56Y6+lV9qxPS5unUxZeNNUa+u2isrfToZSHc7EO58nHqcD3PFbOl6gNU0+O8W2uLZJOUS4UKxXscAnAPvzXJ674du9X8XXc9urI0Frby27SoTBLIjudjdiOfqMg11OkahJqenJPcWc9lPkrLBOhUow64JHzD0I4NKPw/13f8AX/Djfxf12ReooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAwvDn+suPov8AWt2sLw5/rLj6L/Wt2gbCiiigQUUUUAFFFFABRRRQAUUUUAFFFFABRWa3iXQllMTa1pwkDbShu0yD6Yz1pmteJ9I8OmEaxd/ZjPny/wB077sYz90H1FAGrRTYZUngSaI7o5FDKcYyCMinUAFFFV7vULaxkt0updjXMohiG0nc5BOOBxwD1oAsUUVTu9UtrO+tLOQs1xdsVjjRcnAGSx9FHr7igC5VTVNNh1bTpLO5Z1jkKkmMgHhgR1B9Kt0UpRUlZgA4FFFFMAooooAKKKKACiiigAooqnqmqW+kWf2q93iAOqu6oWCAnG5vQDuaALlFAIYAqcg8gjvRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGF4c/wBZcfRf61u1heHP9ZcfRf61u0DYUUUUCCs3xDeNYaFcXEd7b2DqBi4uF3KnIycdzjOB3OK0qyvEehL4g0sWv2hraSOVZopVQPtdTwSp4I9qTGjmfCHime/8Stpv9vR67A1s0vnfYjbNEwYDbjAyCDUFtf8Ai+/0W+1aDV7aGKxln2QG1VjcBGPDHjbwMDHPr61v6X4ZvrTXo9W1TXJdSnWB4cNbrGoBII2hTx0565/Cub8OeHdR1rSbqN9elt9LmvZ1lsordQzDzDkeZ1Gae+i7P8xbb9/0NS/1jXNR1bRrXQ7qGzXUbA3EhliD+V907gO5GcAZxzzWl4ZvdTN9qek6zcpeXFg8ZW6SMR+ajrkZUcAjBrJ17T7uTxxo1pol/wD2W8dhKElECy7UBUbdrceldDoOhDRo7h5rqS9vbt/MubqUAGQgYAAHAAHQU1bf1/MWu3p+RrVz+o6x4kttQlisPCv223UjZcf2jHHvGP7pGRzxXQVz2o+BPDeq6hLe3+nebcTEF38+RcnGOgYDoKnqV0NbS7m9u9PSXU7D+z7gk7rfzhLt54+YcHI5q3VTS9KstF09LLTIfIt4ySqb2bGTk8kk9at1TEeZ+HLeeVGI8EWWpRG9lB1CWaENjzTk7WUt8v17V2viz/kTtX/685f/AEE1Noek/wBi6abTzvOzNJLv2bfvuWxjJ6ZxVPxLoep65Abax1v+zraSJo54vsiy+aD7kgjj0qGvc5f62KT97m8/1LMM5tvCMUyTQwNHZKyyTnEaHZwW9s1x2heLriXxTY2SeJodcju2dJYhYG38nCkgq2Pm5GK6KPwvdz+F7rRda1hr5JkCRypbLCYlAGBgE55HeoLPwjqKalp95qniGS++wOTFF9lSJNpUrjCnryOfwrRu82+hmlaFupT8Raxq1v4ikt31ldAslVPs08lj50dyxHzBpDwmDxjirHiy9W1tPDt9eOsix3qSyvbgsrYickqO49Kuat4e1e/vJ2s/EktpZ3AxJataJMMYwQrN0Bqd/DMIs9Gtbedo4tKmSRd67jIFUrgnjGc5z+lQtl6op7/Jh4cuNR1OJtWvZ1S1ugGtbOMKwjTszOBksfTOBVfQAL3xPr2oy8yQzixiz/AiKGIH1Zs1d0rQzo1/cmyudunTnetkY+IXPUo2eFP93HXpiqUayaH4ymLRs1jrJUq6qSIrhVxg+gZRnPqKrqHQ6OiudvvH3hnTb6azvdS8ueFtkieRIdp+oXFQj4k+EzGXGq/KpAJ+zS9TnH8PsaQHUV5/9v8AFd/Fr13aavBaW2l3VwsSm1V2mCc7CewAwM9eTWzF8R/Ck0qRR6rud2CqPs8vJP8AwGrlp4c+y6brFp9q3f2nPPNv8vHleYMYxnnH4ZpO+r8vx0GrbP8ApFHVtfkHgyw1L+1bbRnu0jd55YTKVDJkhE/iOcfhmsbSPFt7c6P4iEOsR6obC0863vltfJO4q3BQjHBX0rcvfCEk+k6PBaam1reaQqrDdCBXB+UKSUJxzj14pkHg64EWr/bdalvLjVLUW7zSQAbCAwyADjHzfd9uvNOX2reYo7K/kc1Pr3jCJbzdqVuhi09NSOLZT5anP7oeue7H04q74h8aPHrC2A12LQUit45XlNkblpXcZ2gYIAAxz15rbufB/wBo+2f6dt+1aYmn/wCpzt25+f73PXp+tPvfC94btLzRNak0u6MKQzsIFlSYKMKSjdD70Pf+vP8A4H9XEv6/D/gkng3Xn8QaCbmZ45ZIpngaaNSqy7ejgHkZBBxW/WBLrMHhPS4B4s1r7RLK7Bbj7KU398bUBxgVVX4leE3YKurZJOAPs0v/AMTQ2gVzqaZPDHc28kE6B4pFKOrDhgRgiuY/4WZ4R/6C3/ktL/8AEVfk8U2U/h06ppBe+WRvKt0SNgZJScBcEAjnuaT2H1IvBMsh8O/ZZWLmxuJbQOTyyo5C/pgfhXQVmeHNKfR9CgtZ3Elx80k7j+KRiWY/ma06piQUUUUhhRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBheHP8AWXH0X+tbtYXhz/WXH0X+tbtA2FFFFAgooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigBaKSigBaSiigAooooAKKKKAClpKKAFpKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMLw5/rLj6L/AFrdrC8Of6y4+i/1rdoGwooooEFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAYXhz/WXH0X+tbtYXhz/WXH0X+tbtA2FFFFAgoorJ1/UJrWG3tbORYrm8l8tZmAIhQKWeTnjhQcZ4yRQBrUVxdhr2ltqNq+gatdX0ZuVtrxJ5ZXU7w21gX4B3L/AA4BBPHStC/8cWOn3k0U2n6o1vA+yW9S0JgjIODls9j6CgDpKKw9Z8Xafol3b21zHczS3MRkgW3j8wycgbQM5JOeO3vRZ+LbC80W+1FYbqL+zwxubaaLZNHgZwVJxyOnNHS4G5RXM2Xj3Sr69toY7e+jhunEcN3LblYJH/uBs9c8dMV01ABRVDWdZtNCsVvL8ssBlSMsoB27jjJ56DvVDSfF9jq2pfYY7a+tZGjMsLXduY1nQdWTPUcjrijcNjeorln+IWkpMx+zag9ksnltqKWxNsDnH38+vtW02sW665DpW2QzTW5uFcAbNoIHXOc8+lAF+isSbxRBHHqBgsL+7ewnEEkVtCHdmIByoB5HPtWR4S8Z3Oq2tpDe6VqrzTO4a8FmBbj5jj5geMAAdOtC1B6HZUVz+qeMbLTNQktFs9QvngANw1lbGRYMjPznPHHNXL/XrW18Ly63bsJ4Fg82Ij+PP3R7ZJApdLjtrY1KKy/D2nzWOmB76Vpr66bz7pyeN5A4A7AAAAegqEWmpjx012Xl/ss2GwJ53yedv/uZ67e+PxqutielzaoqvqN9Fpmm3F7cnEVvGZG98DpVTw7DeRaHC+pyySXc+Z5Q7E7C5zsGegUEDHtSGadFYvhy01O1k1Y6q8riW/ke18ybzMQnG3HJ2jrxxVbWmk0TXrPV4nb7NdSJZ3sRY7fmOI5AOxB4PsaO3mHc6OiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDC8Of6y4+i/wBa3awvDn+suPov9a3aBsKKKKBBXI+NLO4v7y0tbQgS3FleRRZOAXKpgZ7ZAb9a66qepaZBqkCJM0kbxOJIpom2vE47g/iRg5BzzSY07O5x2laLr02vRS6jpUFhZ+bDOdlwr+WIkKJEAO+Tuz0rC1q31/UYtThvYPE0t8zSBIrZgliY/wCEf7XHYcnp1r0+ysZLRpGmv7q8ZwBm4KDaBnoEVQOvpVum9RLQ4TV9QOj+L9BnfT7y8ZdNkQw2kPmSA/Lzt46UskF5P4f8Wa1e2cliNQtiIreXiQIkZAZh2Jz07V1kukwTa7b6qzyCe3heFVBG0hiCcjGc8etTahYx6lptzYzsyxXETROUIDAEYOM96UruL+f4u4R0a+X4I4e0vLvxJpeh6VaaLf2sNs9vNPd3cPlxhYwD8nPzZxx9a6zWdWu9MmsUtNKn1AXM4jkaE8QL/ePB4/Icdav2lslnZQWsRYpBGsalupAGBn8qlq5PX5iS0+RgeMLWW80q1jggeci/t2ZUQt8okGSQOwHWte7sobpSzRx+eI3SOYoC0YYYOD1FWKKi2lir63PPEutWTwgfCf8Awjd6b7yDaCfyx9lI6eZ5mfTnGOtaOpxXfh7X9I1CPT7zU7eCwayk+xx+ZIrZUhiPQ4612VFO7vf+v61Fpt/X9aHLeDI9QN3rlzqdhJYvdXYlSN+flKDGD0Pvjvmq3hPUbvSYoPD9/omprLHNIou0g3W5UszBt+eBzXZUUB0PM9T0q703xBqTXA8VmG7uDPC+hyZjIYDIZezDp7jFbl5o4/4VW9jpkV4uyASxRXigTfK/mbWA78YxXYUUbRsv6sP7Vyvp99Dqem297bMGinjDqR7irFU9N0q20pZ0sg6RTSmXyy2VQnqFHYE849SauU2Sjn/EH/Ez1fTtDXmN3+13Y/6ZRkbVP+8+PyNdBUYt4VuGuFhjEzqEaQKNzKOgJ645NSUug+oVznjMi6s7HSY/mnv7yIKB1VEYO7fQBf1ro6pppdsusSamQz3LxiIF2yI1HOFHbJ5Prijqg6MuUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAYXhz/WXH0X+tbtYXhz/AFlx9F/rW7QNhRRRQIKKKyvEGo3FjaQR2AT7VdzrbxNJ91Cc/MR3wAaic1CN2Bea9t0v0smkxcOhkVNp5UHBOenep65SytNRtPG1uNU1IX7NZyFWECxbBuHGB1rFn8bzSNPeRa5BbGNm8rTWs2feAeA0mOCfbiuOWOhSinV0vfTTo/XX5XfZB1f9dD0KeeK1t3nuJFjijG5nY4AFUNK8Q6XrUkkemXXntEAXHlsuAfqBWB4un1W88OCe1WyGnTRwu6y7/N3F1IAxxjkfrXQ6SNYVJF1oWAxgRCy34A753fh0rRVpyrunayXlv87/AKO/kHTQ0KrwahbXN5c2kMm6e12iZdpG3cMjkjB49KsVxKW+sXfjLX4tGvotPUGBpJmhErE7PlUKeMdcn6V6NKmp813ayv8Aiv8AMyqTcLWV7u34M7aiuMn8YXEHgew1CeSKC6u5fs7TshZIyGYF9o5PC5x6mk8K+KHvddOmtrMesxvCZVnFobdo2BHykYwRg9fatPqtTllLtfv038jL61T5ox727ddvP7jfsPE+j6nqcmn2N55t1Fu3x+U4xtODyRjrT9Y8RaXoPlf2tdfZ/Ozs/ds2cdfug+tU7v8A5KBp3/XjP/6ElXPEf/Ir6n/16yf+gmp5afNHR2fn5tdv0LTm1JXV0+3kn38+5fhlSeBJojujkUMpxjIIyKfWFcam+keCIbyGMSSpbRCNCeCxAAz7ZNUHl8RaFd2M+qapBqFtdXCW8sK2wj8kvwCrDkgH17UlRcr2fWy8wdblSbXS78jrKKKxdV1SbTdesxJJiyltp2dMD7yAMDnr03VlGLk7I1lLlV/T8XY2qK5i38ST317p8iI1vD9luJ7u3OGYGMhducdjnpiqDXvig6D/AMJIuo2yw+X9oGm/ZwV8rrjzPvZxW6w8urS/4dr9PQxeIj9lN/8ADJ/qvM7aoby7gsLOW6u38uCFS7vgnAHsOa57UNR1e812xs9EuIreK7sTOzzRhvLG5fmA7nBxjOOa0tCXWIY7mDXJVuGjlxDdKip5yEA5KjoQcj/Oah0uWPNJr0672KVW8uVJ+vTa5NpGuadrtu8+lXHnxxtsZtjLg4zj5gPWrzuscbO5wqgkn0FYXh3/AJDHiD/r+H/otKz7mbW9efVH03U49NsrKR7cILdZWmKr8xJPQc8YqnSTm0nZWT187dl+go1Go+8ru7WnWzfd+Xc6i0u4L6ziurV/MgmUOj4IyD3weah03VLfVY5ZbIs8UcrRCQrhZCOpU9xnIz6g1y011LafCG3a3cpLLaxQK4/h3kIT+RNdfZWkNhYw2lsgSGBAiKOwAxWU4qM5JdHYunJzpxk+quQ2Gq22oyXUUBdZbSUxTRyLtZT2OPQjkHuKuVzeor9h8faRcw8HUIpbWcD+IIu9D9Rgj8a09bh1aeziXQrqG1nEyl3mTcDH/EOh5/zkdajov6/ruadTRooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMLw5/rLj6L/Wt2sLw5/rLj6L/AFrdoGwooooEFZ+s6UNXsVhE7280ciywzIMmNx0OO/U8VoUVM4qceWQGHp3h2e11hdTvdUkvrnymjcvGFBBIxtAOFAweO+ah/wCEc1O0kdNF1+SytWcutu9qkwQk5IBPQZ7V0VFYvD03FR1VuzaevmncChq2mNqujtYvPsZihMuzOSrBumR1xV+iit+VXcu4BWdY6T9j1nUr/wA7f9uMZ8vZjZsUjrnnOfatGiqUmk0uonFNpvoc/F4UWPw1b6X9tkWa1lM0N3Gu1kcsWBxk/wB4jHerOk6drNndM+qa9/aMJTCxfY0i2tkfNlTzxkY9616K0dabTT6+S/pfIz9jBNNdPN/0/mcpN4W1+XVFvx4q2zRq0cZ/s6M7UYgkdcHoOa0tf0fUtXgEFlrH2CB42jmT7Ksvmg+5II79K2aKp15tp6aeS/yEqEFffXfV/wCf5GFbeHJ38Pz6TrepHUYZFCRssCwmJQOMYzkggHJqG18MXhvbaXWtcm1KG0cSQQtCsYDDozEcsR7966Oil7epr5+S/Dt8g9hTsl283+Pf5nNt8QvC6MVbVMEHB/0eX/4mm3MeneOrW2uNM1A+TaXJEjCFvnBXDR/NjGVbrzXT0lP2kI6000/VP9BclSWk2mvRr8bmNaeHEtvEF9qRnMkd1GUFuU4jzjdznnJUcYrNHgu58j+zjr1ydGz/AMeXlLu25zs8zrj29K6uikq9RbP8um3/AA+45UYS3X59d/8AhtjPOlD+3odSWQKsNq1sIQnYspznPbbjGKs30NxcWMsVldfZJ2GEn8sSbD67TwanorPmbtfoacqV7df+GOb0bw5q+marJdT+IftUU8hkuIfsSJ5rbdoO4E4xgdPSi68Jzy312bXWbi0sb1zJc2kcY+diMHD9VBxyBXSUVq69Ry5vlsv8jJUIJW13vu/8zDn8OCXwT/YBn3FbcRJNtx8y8q2Oe4Bqz4f1KXUtLBvIXgvYG8m6jZSNsgAzj1ByCCOxFadFYuTbbfU1jFRioroc7bI+teL/AO0CjLZaWslvAWGPNmbiRh7KBtz3Oa0db0251Szihs9Sm050mWQywjJZR1XqOD/nNaNFLsUFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBheHP8AWXH0X+tbtYXhz/WXH0X+tbtA2FFFFAgooooAKKKKACiiigAoork30W21rxhq0d+0r2yJATAshVXJU4LY5OKxq1JQsoq7btvbo359g6HWUtcXrtrdaL4csdMhlkvIpbvyiPNEJaM7iI95Jx0xnvS+H9HvYNRmhk0V9N0q4t2SWA3wmVmPcYOQSMisFiZOp7NR/PR2vuk15b/INjqLbUIbu8u7aIPvtGVJCRwSRkY/CrVcPpXhLQZ9e1e2lsgwtZY/KTznBVSgP97nn1rXnKp8QrTJCg6e4GT1+enSrVHGLqJau2jv38l2B9f66nQ0VxMc0dxpPjGWB1kjaSTDKcg/u6v6f4UgC6dqUN1Ot+uySWd3LGVdvKYzgDnjjjFOGIlUa5Y6aPfu2vntf+tR6fj+B1FJXDxaDaazqniCTUDLIsNwfKjWQqqNsB34HU9K6LwtPJc+FdPlncvI0Iyx6nHFVRrOo9VbqtQej+/8DWrNGrk+KDpH2fj7L9o87f8A7W3btx79c1pUz7PD9p+0eVH5+zZ5u0btuc4z1xntXSiZKTWjK0epwy6zNpqrJ50MSysxA2kMSBjnOePSiz1SG9vr20iWQSWTqkhYDBJGRjn/AArmruz1G88fXq6Xqn9nMtnEXb7Osu8ZPGG6VltLqOlWHikm7a5vFngR7lVEJIYAE8cLwcZ/GtFFNfL9bHJLESi9U7JvXTZRb73vp2PRqwr3Xb6x1y3tJdI/0O4mWFLz7SvJIz9zGfWsDQtK1Ky1yzuLHQJNNtmJF1J/aSzrMpBwSM9c4ORW94k/4+tE/wCwin8jRypSXUaqyqU3LWLX9dUvy+Zf1a7vrK0Eum6d/aEm75o/PWLAx1yf5UaNqJ1fR7e+MPkGZSTHu3beSOuBnpVyX/Uv/un+Vc9oE0lv8PYpoBmSO2kdOM8gsRUacrf9dTd3VVa6NPT0t8+p0lJXnlxpOn23g6LxFb3cp1Zo0lF355LSSnGUxnHXIxivQkLNGpcbWIBI9DVSjYmlVdTdW0TWvR/1/wAEq32ow6e1qs6uftM6wIVAwGPTOT04pb/UItPWBpldvPnSBQgB+ZjgfhWX4tdYrPT55DtWHUYHZj2G7BP61hQXt1dXllbX0jyNPfw30G7nETKxCj2G39aIxv8Af/l/wSalVwk4+Wn4neUlcLa6Rpuu6bqGraxcSNfRzSgyeeyfZNpIUAA4HAB59adP9p1fSPCouriaGa4kIeaM4cjy25z2JHf3o5f0/EXt3vb017NL5bnbuWWNii72AJC5xk+lY2ja5e6hqVzY6lpf9nzW8aSbftCy7gxOOgx2pdI8N2+h6hPLp0jx2s6ANbElgHB+/kknJHFR2f8AyPep/wDXpB/NqFYpufut6a7bp/h/kbtUdH1P+1rFrjyvJ2zSRbd277rFc5wOuKwLuxste8UanBrzs1rYRRGGFpjGgDAlnOCM9hmrfgURL4ZAtjmEXEwjIOfl3nH6UWVriVWUqiXTX108v+CaWi6r/bGnm68nyf3rx7d277rEZzgelMudYNv4jstKFvvF1G8hl342bR6Y5/OofCsPkaIUzGT9omJ8uRXHMjEcgkdMcVqtbwvcJO8MbTRghJCoLKD1APUZpaJ/12LjzyprXXT89fw0JKKKKk2CiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAwvDn+suPov9a3awvDn+suPov8AWt2gbCiiigQUUUUAFFFFABRRRQAVBHZW8N5PdRx4nuAokbcfm2jA46Cp6KVk9wILyyttQtXtr2FZoX+8jCqWl+GtJ0W4efTLTyJHXYzeY7ZGc9yfStSiodODmpuKuuvUDK1Hwxo2rXYub+yWWYADeHZSceu0jP41LqehabrKRLqVqs4iPyEsVK/iCDWhRS9jSaa5Vrvpv6gZ8Ohabb2lzawWixwXQxNGhIDDbt9eOPSr0cawxJHGMIihVGegFOoq1GMdkBWh061t3uXhi2tdNvmO4nccYz1449KfZ2cFhZx2tonlwxDai5JwPqeamooUYrZAFFFFUBXSwto9QkvkjxcyoI3fceVHQY6d6YulWSveN9nUm9INwGJYSYGOh46elW6KLi5V2MWw8IaHpmoJe2NiIbhM7XErnGRg8E46Gm3vgzQNRvZbu8sPMnlOXfzpBk/QNityiq5pXvcy9hR5eXlVt9luZV14Y0i80230+5tN9rbf6qPzXG38Qcn8aXSfDek6HJJJpdp5DSqFc+Y7ZH/Aia1KKXM+4/Y0+ZS5VdeRjw+EtCg1EX0WmxLcBtwbJIB9QucD8q2KKKLtlRhGF+VWuVdS0y01eya01GHzoGIJTcV5ByOQQaZJo1hLfWt49spuLNSsD7iNgxjGM4P41doou1sNwjJ3a/rf8zHvPCeh3+ofbbvTo5LgnJbLAMfcA4P4itCawtZ5LaSWIFrVt0OCQEOMdB7GrFFF3sJU4JtpLXcZNEk8DwyrujkUqwzjIIwaw7fwN4dtbmK4g07ZLE4dG8+Q4IOQeWrfooTa2FOlTqW54p27mZqXhzSdXuo7jUbKOeWMYViSOPQ4PI+tXLOxttPtzDZxCKMuz7QSeWOT19zU9FF3aw1CKlzJalaw0600u2MFhCIYi5cqCTyep5qzRRSKSSVkFFFFAwooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAMLw5/rLj6L/Wt2sLw5/rLj6L/Wt2gbCiiigQUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBheHP9ZcfRf61u1heHP8AWXH0X+tbtA2FFFFAgooooAKKo6Rq8GtWbXNqkqIsjRkSptORTdV17TNFCHU7tYPM+6NpYn8ACazdWmoe0bSXcDQoqG0u4b60iurV/MhlXcjYIyPoeamrTcAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACimyypDC8shwiKWY4zgDrUVle2+o2Ud3ZyeZBKMo+0jI+h5oFdXsT0UUUDCiiigAooqG8uksbKa6mDMkKF2CLkkAZ4FAE1FVtOvotT02C9t1dY50DqJFwwHuKs02mnZijJSV0FFFFIYUUVVl1K0g1KDT5Zdt1cKzRR7T8wHJ5xigTaSuy1RRRQMKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDC8Of6y4+i/wBa3awvDn+suPov9a3aBsKKKKBBVbUILq4snjsLz7HOcbZvKEm3nn5Twas1V1G3u7mzMen3v2GbIIm8kSYHcbTxWdTWD0b9NH991+Yyh4Zur260uf7fcC5nhuZYfNKBN204HA6VgBfFB8ZMcaMb1bEYJEvlhN56d92fwxW1oWg6jpFw5n1r7XbyM8jQ/ZVTLscltwJP4dK0BpuPEDan5v3rYW/lbfRi2c59+mK41RnVhS57prfVfytX69fzF0f9dS3B5v2eP7Ts87aPM8vO3djnGecU+iqWrazYaFY/bNVn8iDcE37Gbk9OFBNegCLtFcr/AMLM8I/9Bb/yWl/+IpW+JXhNGKtq2CDgj7NL/wDE0AT+M9S1PTbLTjorxrc3N/Hb4lUFWDBuD3AyB05qDRrrWrLxa+ja1qMepLLZfao5Vt1iMZD7SuB1HPU81IbjTPHEFpNo+o7o9Ov47hz5DDcVB+X5sYznrzWk+jB/FK6wZ+BZm18nZ1y4bduz7YxihXX9eWn4g7P+vP8AyOF1fxpc6dqM0sHiy0uZI58DS4rAlGXdjb52OuPfrWl4r1zX7TVtRTSbyO3trGwju23wq5J3MNoz/e7ntt461Ifh9e/2W+kp4lnXTNxaK2FqgK/Nu+ZgcuM/StfVvCv9qTapJ9s8r+0LJLTHlbvL2knd1GevTj60vsq39af5j0v/AF3/AMjCj8U6xpNprU+syw3LwWsF1CiR7ViMpICZHLAHHPWqmieNZW8QWFs3iSHWkvZPKkgWwaAwEjgqxA3DPHPNdRJ4SguZL/7XcM8V7aQ2xVV2lPLzhwcnnJB6cY70mm6L4gsr6J7vxS97aR53W72Mas4xwC4Oc9Dmq0UvInXl/rsv1Ohorlj8S/CQJB1bkf8ATtL/APEUf8LL8JYz/a3/AJLS/wDxFIZ1NFYGm+OfDusahFY6dqPnXMudieRIucAk8lQOgNb9AGJq+qa9Z3oi0nw5/aUGwEzfbo4ee42sM+nPvRDquvPo09zL4b8u+RwIrP7dGfNXjLb8YHU8H0o1fwboOvXou9WsPtE4QIH86ReB0GFYDvSQ+DdBg0WfSYrDbY3DiSWLzpDuYYwc7sjoOhpdB9SKw1nxJcX8UV94U+x27HDz/wBoxybB67QMmuhrntP8B+G9Lv4r2x03yriE7kfz5GwfoWIroaroT1GTRLPBJE+dsilTj0IxUOm2EWl6bBY27O0UCbFLkFiPfAFWar319baZYyXd7J5UEQBd9pOOcdBz3peQnb4n0LFFc5/wsDwx/wBBP/yXl/8AiaD4/wDDKnB1Ptn/AFEv/wATVckuxj9aofzr70WfFN9fWOn2zaU6rcTXccI3gFW3HGDntWBZ6x4gjvovtd/FNbw6oNPkAgVTPn+I4+7jIwB171srfaX4wWJdKv8Af9huYrh/3LDOCSB82OuDzzTh4YwzH7X11Qaj/qvp8nX26/pVR93SX9bf8EwqKVV89KV15PTr+tv6uYfiLxJPYajdCLxPbW7wk+XYrZGXdgfdZ8cE/pXQXjalqXh62vNN1AabK0InfECy7gUzt+bp9aoSeD7rdexWmuTW1jeO8klukCltzdfn649vStC70bUH0a0sNO1f7F5EIikf7MsnmgKF6E/L0/Wk7ctlv/X9dRxjV9pJzT5X0T8+nvdv8I2z1G6l8CrqMkubo2JlMm0fe2k5xjHWsW31HxHb2ukapf6hBJBeywxPaLABhX4Db+u7vjpWtpnh69stFuNMu9X+1wvB5EH+jKnkjBHY5bqOp7VPNoHm6Pplj9px9gkhff5f+s8vtjPGce+Kq8VJ22uvu6i5Ksqcb3TSfXrpbZ6/1c2KKwLjxx4dtLmS3uNR2SxMUdfIkOCOvRaj/wCE/wDDO3d/afGcf6iX/wCJqOWT1sdDxNBOzmr+qOjqlNpUE+s22pO0gntkdEAI2kN1yMVm23jjw7d3UVtb6jvlmcIi+RIMsTgDJWt+iziVGdOsvdafp95V1SO9l0u4TSpo4LxkIhkkXKq3YkYP8j9KksluUsIFvpEluVjUSugwrNjkgemaj1S0lv8AS7i1truSzllQqlxH96M+o6fzFSWUElrYQQTTvcyRRqjTP96QgY3H61JqTUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGF4c/1lx9F/rW7WF4c/1lx9F/rW7QNhRRRQIKKKKACiiigAooooAWikpaAEooooAKKWkoAKKKKAFopKKAFpKKyrnV5D4hg0mwiWWQL5127H5YY+QP+BE9B6AmgDVorK13VptFigvDCslism27fJ3RKeA4HcA9fatUEMoKkEEZBHegAooooAKKKKAFopKKAFpKKKACiiigAooooAWikooAWkoooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAwvDn+suPov9a3awvDn+suPov8AWt2gbCiiigQUUUUAFFFFABRRRQAV514pjtNZ8Q3qR+H7jWXsEVZpZNSFrHbkjI2gnB9Sa9FrF1HwfoGq6kNQ1DTY5rnjLlmAbHTIBw34g0mrjucdM02p/DzwmLi6mSSbUYozOr/vFGZFyG9QO9a+maXY+HvHsdhobOsM9i8l3AZWcBgy7XOScE5Iqz4j8KRXei6TpGn2W6xgv43mhEuNsXzbuSc/xdjn0rY0fw7pPh+KRNHskthJ98glmb0yWJP4U5a8z9fyJtol5L8zze00LT4vhmNfnuZ11KFWa0mM7DyWDnaiLnHJHoepr1L7XHbaclzqEsdsoRTI8rBFUn1J6c1ynhHwDpthp9pd6tpSDVo2ZnZ5S4B3HacBivTFdVqemWesafJZalAJ7eTG5CSM4ORyMEU5dbD6lHxS6yeC9VeNgytZSFWU5BGw81jaX4KtwmlavBe3K6mnlyzXUjs5mQrzHtyAF5wMDgDvXUSadaS6WdNeEfZGi8kxAkfJjGMjnpU8MSQQJDENscahVGc4AGBRs215fqG6Sf8AWxyOk3MFpe+MJLqVIkjutzF2xgGIYNc/HZ/2h4V8EWhmlgWaZlZ4W2sF2tkA9sjI/Gu3vvCOg6lqo1K+02Ka6GPnYnDY6ZXOD+INWI9A0yGGwijttqaexa1HmN+7JBHrz1PXNJbK/l+APfTz/EwPDunW+g+ONR0rSw8Vk1nFceS0jOFfcQSMknkVd8M/PrniSSX/AI+Pt4Q56+WI12foTW0unWqapJqKxYu5IhC0m48oDkDGcdT6VnS6ZdWnilNU04K8N2ghvombH3fuSD1I6EdwaF0v/Wt/+AD6/wBdP6Zf1WOKXRr2O5x5LQOHz/d2nNUvCLyyeDdJa4z5htI857jbx+mKXxDY3mq2sem2xEdtcttvJ93zLEOqqPVumewzWrFEkMKRRKEjRQqqOgA4AoXX+u/+YPp/X9bDqKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigDC8Of6y4+i/wBa3awvDn+suPov9a3aBsKKKKBBRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFAGF4c/1lx9F/rW7WH4b/5ef+A/1rcoGwooooEFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAf/9k=)

*Figura 30 Struktura kryesore e një fajlli në Flask*

URL-ja ‘/’ lidhet me funksionin hello world(), përmes **route()**. Ky funksion thirret dhe del si output kur faqja kryesore e uebserverit hapet ne shfletues (browser).

Funksioni kryesor që aplikacioni në Flask të niset dhe të ekzekutohet është funksioni **run().**

Dhe nëse dëshirojmë të zhvillohet në kompjuterin tonë kjo pjesë e kodit, bëhet shumë lehtë duke e ruajtur këtë program si server.py dhe pastaj ekzekutojmë python server.py:

Më pas fillon një server në internet që është i disponueshëm vetëm në kompjuterin tonë. Në një shfletues uebi hapet localhosti në portin 5000 (url) dhe do të shohim që shfaqet “Hello World”:

![A close up of text

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RDmRXhpZgAATU0AKgAAAAgABAE7AAIAAAAJAAAISodpAAQAAAABAAAIVJydAAEAAAASAAAQzOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEZpdG9yZSBIAAAABZADAAIAAAAUAAAQopAEAAIAAAAUAAAQtpKRAAIAAAADMDAAAJKSAAIAAAADMDAAAOocAAcAAAgMAAAIlgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADIwMjI6MDI6MTggMTU6MTk6MjMAMjAyMjowMjoxOCAxNToxOToyMwAAAEYAaQB0AG8AcgBlACAASAAAAP/hCxtodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIyLTAyLTE4VDE1OjE5OjIzLjAwMTwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5GaXRvcmUgSDwvcmRmOmxpPjwvcmRmOlNlcT4NCgkJCTwvZGM6Y3JlYXRvcj48L3JkZjpEZXNjcmlwdGlvbj48L3JkZjpSREY+PC94OnhtcG1ldGE+DQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgIDw/eHBhY2tldCBlbmQ9J3cnPz7/2wBDAAcFBQYFBAcGBQYIBwcIChELCgkJChUPEAwRGBUaGRgVGBcbHichGx0lHRcYIi4iJSgpKywrGiAvMy8qMicqKyr/2wBDAQcICAoJChQLCxQqHBgcKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKir/wAARCACRAcgDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwD6C0y5luNOilmbc7ZycAdzVvefWs7RP+QVH9W/nV+gB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tG8+tNooAdvPrRvPrTaKAHbz60bz602igB28+tFNooAoaJ/yCo/q386v1Q0T/kFR/Vv51foAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAKGif8gqP6t/Or9UNE/5BUf1b+dX6ACiiigArM1zxFpnh61E2qXSQ7gfLQ5LSEdgACfx7ZrTri/H2k3KwTa5p08azJZtZyQzJkOjnGVP8LZb8aTv0GrdToPDuv2viTR47+yPXCypz+7fAJXJAzjPXGK1KxfDerjULaWzmtHsrzTysM9uzBgvy5UhhwQRW1VStfQlXtqFFcJBpuo+INe16N/EGo2VraXW2GO2l2kMUB5P93/ZGO9OXVtRl8D6N4hkuZDJZzBrsRsQs8W4xuSBweMN7YNJaq/p+I+v3nc1Bb31tdXFzBbyh5LVwkygH5GIBA/IjpWGvna34m1BYry4hsbS2+yg28pXdNINzOO2VUrg9ia5/Q/Cfn65rcX/AAkGux/ZrpF3R3uGl/dqcuccnt9BQtX/AF3B7f12O1/taD/hIP7H2SfaPs32ndgbNu7bjOc5z7VeriNZ0651T4lpbW2oz2CHSQZpbcgSsvmn5Vb+HnHPtVeHWNTsPCmp2yX7y3Ftqv8AZ8F5c/MyKzKAzHuRuPWhar+u9gej/rtc7+iuLksb7wtrWkPHrmoajFfXItp4L2XzM5UncnHy4Iqmtjqms3viOR/EGo2lvY3Ugt4raXbhggPJ67enyjA60r9f66f5jt0/rr/kegUVl+GL6bUvC2nXl026aa3VpGxjcccmtSqas7Ep3Vworz6wN9Y65BJ4l1PW7O8e425Yh7CfLYCLtGFyPXBFHifVbu58VXGnf8VCtrZxRkDQosszsM5duwxwB35pdEV1Z6DRXCNqmr/8Ku1K4vPt1td2zMkM1xGYZmQMNrEDocHBI9K1LPw/rVrqlnqB1+4uyxP26Cd8QlSv/LNAPlIOO/4+r6i6X/rQ6eiuLGr3ejaTr9hLNLcXtpNizaRyzus3+q5PJwxI/CoPEt3faVb6Loaz6vN5sLNdT6cvm3UmwDoSeMk5J+gqf6/r0H/X9ep3dFcT4YvtTNrq9vOmsraQweZazatDsmBwdw3fxYIBB61Do2i6/qGg6XrCeI7pr0pE4gkkIt2j4yrjBLMRn5j3/Oq6/cI7yiiub8dXF3beHo2065e1na7hRZEJGMuBz6j1HekB0lFcdbWl94f8XaZavrV7qUWpRzCdLuTdtZFDbkA+6O2KhTVL7T9A1bR2uJZtUguvslrLI5LuJjmJ8nngE8/7FHp/WtgO3oqGztza2UNu0skxijVDJIxZnIGMknqTU1ABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBQ0T/kFR/Vv51fqhon/IKj+rfzq/QAUUUUAFc14p8I/8JA8UsF5c2su+NZvKuGRXjVtx+XBBYcke9dLRQBnaLoVnoVvJHZ+bI8z75p55C8krerMa0aKKAOFtvCLat4g16a/u9WsoJLsBYoJjFFcJsGSRj5h2yDW1rkyaPosWladod1fpPC1vFDbR5jQYAw7E/KDnr7GugopW05R31uZPhfRjoPh22sZG8yZV3TPnO5zyef0+gFVdQ8HWt7q0uowajqenTzbfO+w3PlrLtGAWGD24roKKpu7uSlZWKP8AZEH/AAkH9sb5PtH2b7LtyNm3duzjGc596qjwvpzWOpWc6yTwalO1xMsjDhjj7uAMY2gjvWxRSGc/pfg6y0zUI72S81DUJ4QVga+uDL5IPB2jAxxV+10O2s/7R8p5T/aMrSy7iPlLKFO3jgYHfNaNFG4bFTS9Oi0jSrfT7ZnaK3QIjSEFiB64Aq2QGUg9CMGiih67gtNjmrfwLYQ3UTy6hqt1BC4kjs7i8LwoQcr8uOx6ZNWtY8K2mr3q3gu77T7sJ5bT2E/lM69g3ByK26KAMc+G4ZPDU2i3V/f3UUwIae4mDzYJz94j+lbAG1QB2FFFAGVe+HbK/wBctdUmaUTWwGEVgEkwSV3DGTtJJHIp2t6Baa7FELl54JoGLQ3FtJ5csRPXDe9adFHSwdbmRpfh5dNtrqB9V1O/W5XaTfXHmlBgj5eOOv6Cr2m2EWl6XbWNuztFbxiNC5BYgDHOMVZooAy9e8P2niK1ht76SeNIZlmUwPtJYevB45rO8d2M+o+HY7a1jndmu4cmAEuq7xlhjpjrntXS0UAYWk+FrPRbqS/Nzf6jd+WUE97N5sir12rwMVmaZay+IPGK+IJ9LutOt7WDyo0vF2STSc/MU7BQzAH3rsKKOtw6WCiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAoaJ/yCo/q386v1Q0T/kFR/Vv51foAKKKKACiiigCG8leGxnlieFHSNmVp2xGpx1Y9h61yfh7xfdavrsto6wRRmXI82QDcojX5YcAeZ82Wyf4SPXjsZI0ljaOVFdGGGVhkEehFQRafawzyTJAvmSSeazHnD7QmRnp8oA4oW4PYsVi+Jtcn0W1tVsoY5bq9uFtofOfbGrEE5Y+nHQVtVieKb7RrXTVi8SQNLYzkhnMLSIhHQnbyp9CKTGiXRZPEHmSx+IYdPAABjlsXfafVSG5z71EPGXh46x/ZY1SH7Xv2bMNjd6bsbc9sZ61z/hi7+1axeW3hm8vLrQ/sh2SXQYrBNnAWNnG7GO3auThkRdFTQdY8WtphUhJNPk0XcUfdnIdeTzzuzk1S1aQuh6tqXiDS9Hk2aneJbN5Rm+cHlQQDjjk5I46n0qCDxZolzpQ1KC/V7TzRCZAjfK5OACMZHUckY5rPvrWGf4j6T9pjSYw2EroXXOGDqN315NaWoeHLG80/VYI4RG+pr++cEnLgYVsdARgHj0qel/63Drb+ti7d6ja2M1tFdS7JLqTyoVCkl2xnHA44HU8Vz2r6zp1z4t07TIvEE9leWtwDJZxwuRcZAIVmxtxj69fWq3hm4m8Q6xb3t4pDaRa/Z3BH/L0xxIfwCj/AL6qv4v8W6JDrWl2sl7tmsNRWS5Xyn/dqEbnO3B6jpmq0Ul6/r/TDWz9P0/pHW6nrenaN5X9qXaWwm3bC4ODtGTz0HHr1qPRvEOleIIZJNHvFuViO18KylT2yGAP41ga9LY65rHhKePbc2k9xJIm9DhgI8g4PuO9SxAJ4+1/CMwOnQllj4Zj83T37VLdr38/wVx77eX4uxPcfETwra3LwS6um+NtrbIZHGfqqkH8K6RWDoGU5DDINeOaFrlvcWuj6Xqms6Za2VnOkyKttMJ9ynIVnKhBycFga9kByMjpVW0JvqZfibU59G8NXuoWqxvNbx7kWQEqTkDnBB71W0g+LDeqdc/sX7IVOfsfm+Znt97jFR+PP+RE1X/riP8A0IVLovg/QtCuheaVY+RO0ewv5ztwcEjDMR2pLcb2X9dildya1p/jbTUk1nz9P1CaVRZ/ZUXywsZYDf1PP0rb1bWtO0KzF1q10ttCW2hmBJJ9AACTXH+IvGWgx+L9ID3+P7OuJ1uv3Mn7s+WV/u888cZq34y8QPbW+kT2V2ljZXZaQ6o9mZ/IG0FcJjgtnvS6aD+1/XmdJpGu6Zrto1zpN2lxEh2sQCpU+4IBFU7Hxl4f1LVf7OsdTimuskBFVsNjrhiMH8DXN+CJ2ub7xJcXdxNqSyJCxmNp5DTrsbkRj1HA9ar6HrcEOqadpnh/UJNWtFlCGyurFhLZJg5cS4A46c59M1X2rE9LnWRanOPGV/ZzzqtnBYxTBWAAQlm3MT16AdTitHTtStdWslu7CRpIHJCuY2TdjuNwGR79K4nxHpl3qfjDUEsv3vlWdtLLZs21btA75jJHIz27Z612Oi6pZ6vpcVzp/wAsQ+QxEbWhYcFGXsR6Uo6x/ruxv4v67Iv1hXXjbw7ZaodPutUijuVbYylWIU+hYDaPxNbtePa9r97fWWsWs2qfYpd8obRodK3kqD99pSOM9S340r2Y0ro9Q1TxBpeitGuqXiWxkRnTeDhguM8gdeRx1Pao7LxPo2o6TcalZXyy2lsCZnCMCgAycqRu6e1ct4jvtN03X/Cl3rZVbWK3lJZ4y4DbE2nABPXFMeZNUsPF+tWMMken3ViI4pHQp57IjbnAPOOQM05aJ+V/wFHW3nY6a08ZeH77VE0+01SKS6cfKgDYbjOA2MZ9s5rbrzW21TSdc0nw/ovh2MzXVtPbzTGOBkW22YLsWIAyeRx1Jr0qqasJO5l+JJb+Dw/dS6R5hvFAMYjjDseRn5SDnjNM02XU28QajFemQ2caQ/Zy0QVSSp34OOeR68Vr0tJOxEoNyUr/ANanJ6P4zsI7NI9c1ONbuSeRVBXoochc7RgcDqcVvaprOn6LbrNqd0sCMcLkElj7AZJrj7a2hT4Z6xKsSCSR5y7Y5YhyBn6Uuvtc2eo6VqJ1B9OtxYiP7WLQXAR+CQQfu5Hf2xV8qbX9dDj9tUhTbeun621u/n0/U6+2v7fWdLafSLxWWRSqTKudjY7qe464NZvh6bU11bVbDVdQ+3m18nZJ5CxfeUk8L+FUfCtzZW1jquqNrY1GFnEk8wszCEKrydo68Y6DtUOh+KNHuPF+p+TebvtzQLb/ALpxvKqQe3HPrinyu7S7f5B7eMvZznKzbfXfR9m1272NDW31ax1ixuINVxZXF5FA1n9mTgHr8557frW5e31rpto91fTLBCn3nY/5zXKeLPEukxX1jaSXeJ7PUIZZ08t/kUck5xg8EdKsa/e213baPrMcbXmlw3Bkl2xk4UqQHKkZwDz0qUm4q/f/ACL9rGM5qLu10ve29/S3U2dK8QaXre/+y7xJzHyy4KsB64IBx71hi78R5XJuP+Qz5R/0df8Aj245+793r83606wvrbX/ABlBqOjq721vbOk9z5RRZCSNqcgEkda6uh+607f1f+vvKjetH4tuq2enr5990FFFFQdYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQBQ0T/AJBUf1b+dX6oaJ/yCo/q386v0AFFFFABRRRQAUUUxJopXkWORHaNtrhWBKHAOD6HBB/GgB9FFFABS0lFABRRRQAUtJVO91ew066tLa9uFimvH8uBSCd7enA46jr60AXKhvLVL2xntZSypPG0bFDggEYOD681NRQ9dA2ONXwPqUtgml33ieafSFCp9kWzjRii9F8zr2FdiiLGiogwqjAA7CloouAUUUUALRSUUAFLTJZY4IXlmdY40Us7ucBQOpJ7CiOVJoklhdZI3UMrochgehB7igB1FFFABS0lFAGZeaP9q8Radqnn7PsKSr5WzO/eAOueMY9DVjVrD+1NHu7DzPK+0wtFv27tu4YzjjNW6KVrqw72dyCwtfsWm21rv3+REse7GN20AZx26VPUctxDAYxPNHGZXCRh2A3sewz1PHSpKpu+pKVtAoqOC4huoVmtZo5om+68bBlPbgipKQwopaKACikooAWkoooAKKKWgBKKKKACiiigAoopaAEooooAKKKWgBKKbJIkMTSSuqRoCzOxwFA6kntRHIksayROro4DKynIYHoQaAHUUUUAUNE/5BUf1b+dX6oaJ/yCo/q386v0AFFFFABRRRQBX1CPztNuYhB9o3xMvk+Zs8zI+7u7Z6ZrkPDuiavaeIZZ9SD3MKXHy7p3UQnyVG8ZH74Y/d7jzxnua7eihaO4bqwVx/xI+wf2Tpf9sf8AHj/aUX2j733Nr5+7z+VdhWVrulz6m+mG3aNfsd/HcybyRlVDAgYHXmjqvVfmPv8AM5nwUlomsalP4YFwPD/kKIxKX2NODyUD84x19/wq83iu+Hw1TxD5Vv8Aa2Ckptby+ZdnTOenvWla6HNYeKLu8s2iXT7+IG5gJIImHAdRjHI69Oea5mbwZ4mk8PnQE1LT102Ft0JCP5ko37grnGAB14z0FNdPl+ev/A8hdTY1jXtcj8VR6LodnaTPLZi4825ZlWI7yCWx1HAGBzk1WfxfqMHgzU9Qntbcajpl19mljTcYnYOoJXnOMN61tDSJx40/tffH9n/s8Wu3J37vM3ZxjGMe9ZF34Tvp/D+vWKS24l1LUPtUJLNtVdyHDccH5T0zSX9ff/kHX7vy/wAyWy17X4vEFhaa9Y2UEGprIbdbeRmkhKru2uTwePSjWbrV7XxRpn2m10mbTZrxYbd2jdriIlSSwJ+UHg8j2rS1PSJ7zxDot/E8YisGlMoYnc29No28evrisnX9L8W6lqkElk2ira2dyLi284yhyQpGHwMdz0o6r+uodPl+Jd8Ua7qOkXel2+k2cV3LfSvF5chK8hcg57AHk+wpuh65qj6jqOm+IYLVbqziScPZFijowPZucjFSXOk6lf3ugXl49os1hI8l0Ii20lkK/JkZ6+uKbc6fNp+s6zrkkqrbyaeqL5YLSKUDEnGMd+OaT0T+f5f5j3t8vz/yOfl8XeK5NPstUtLfRY7LULhYbdJXkeVdxwN2CAffGSPSvQF3bBvxuxzjpmvFNJeLQra11uFfDN5Oio3ki5la7YnA4QsVD++ODmva0YvGrFSpIBKnqPartZE9TN8QaudF0o3McBuJ3kWGCEHHmSMcKM9qyLXWvEOn6xZWnia108QX7GOGaxZ/3cmMhWDdc47VreIdHbW9JNtDOba4jkSaCYDOyRTlTjuKy7TRdfv9Xs7vxPdWBisGMkMFij4dyCNzFvQHoKlb6/0hvb+tzMbxV4ouLfVbuw0/TvsmlzzJJJOzhplQnhADwQByTxzxXaWF2t/ptteIpVbiJZQp7BgDj9awrTw7d2/hzXNPeSEy6jNcyRMGO1RIMLu4/PGa2dItHsNFsrOYq0lvAkTFDwSqgHHtxQtvu/4IPf7/APgFfxN/yKerf9eU3/oBrldH1/X9P03QZL2ws49JujBaRgSMbgblwrn+HBxnHXmux1i0k1DQ76zhKrJcW8kSFzgAspAz7c1kXnh67uNA0KxSSES6dPbSyksdrCMfNt46+mcULf7v1uD2+/8AQx/EbeKB410kWZ0naXm+xCXzeR5fzeZj26YrV1U+KR4TlJOkC9CyfaCvm+X5W0/c77vrxVnxJpGo30+n3+iT28V9YSMyLcgmN1ZdrA7eauWkGo3WiyW+vG1FzKro5st2wKeBjdznFS1eLRSfvJmH4KPiU6Vpn2/+yv7L+yJ5fk+Z5+No25z8ufWqMXirxVd6bd6lZ6bpxs7GSUS+Y7h51RjnYATggDv1PT0rW8N6f4m0n7PYajNpc+mW0flpJEJBOQBhcg/LXN6Bp/iTVNEvbGyutPtdKuLu4SSUq7XABchsD7vPNXJ3lp/Wq3Ijotf60O4mvrq58Nm+0WBJrmW3EtvFKcBiRkA8j19R9at2T3MlhA99GkVy0amVEOVVscgH61VubK5g8PNZaHKlvcRQCK2klG4IQMAng+nofpTxLdWOhCW9Bu7uG33SC3QnzXC87VHqaUmk20NXsjMX/ib+Ni3W20aPA9DcSDn/AL5T/wBDqfxTeS22im3szi8vnW1t/Zn4LfgMn8Kf4Z0+XT9Ej+1/8flwzXFyf+mjnJH4cD8Kh+zzah40+0TwyJa6ZDtgLqQJJZPvMPUBQBn1Y0W2i/67/wCQX+0v6/rcluvD6y6Ha6VZ3tzYW8G1WNs213RRjbu7Z4yRWb8PYlh8MPFGCES8nVQTngSHvW3q/wDa32If2D9i+1bxn7bv2be/3ec9Kw/CWk+JNFZrbU30p7JnklJtzIZd7Nn+IAY5PvQnq3/XQTWi/ruZnizw1p9kq3unSXR8RXVyv2SU3LtIzbskYzjYq57cAda6DXvC2gau39oa/aLK1vDhpTM6BUGSfusB3NYkOi+NbbWrrUlk0G5nmJVJLkzFoo+yKAAFHr6nqTW9rml6hrWjW9g0sESzOn29kZhlBy6px3Ixzjil9n+tCvtf1/XoZvw+0eDT9JuL62tzax6lL50MBYny4hwgOSTkjn8a6ykRFjjVEUKqjCgDgCszSbfWob/UX1e8guLaSXNmkSYMSc8NwPb1+tU9WStjz/UbjSNb1XU21dL3UtVF1Jb6fY2bsGgROA4xwuTyS35V2vgi/utR8IWcuoMzXKb4ZGY5LFGK5J7njrWfN4d17TtX1CfwxdadFBqT+ZN9riYyQuepQr19cNxmt3TNIOjeHY9NsJcyRRMFmkGcyHJLEf7xziktI/1/X+Y3rItahcmz025ugNxhheQD1wCf6V5DNPYto66ppRv7/wASxxrd3WoRSN5dvxuKOSduMZG0Dtj2r1PSLTUhoQtvEdxDeXbhllkhXCspJwOg7ewrmYfB+vR6b/wjxv7BdC3/ADSRxMLl4852H+Hnpnr/ACos7jurHaWk/wBpsoJ8Y82NXx6ZGai1Ozlv9OltoLyWyeTA8+HG9RnnGehIyM9s1W1u21WTRjB4cuYbO7BUJJMu5Qo6joe3sanuxqQ0lhp7WragEG1rgMIi3GSdvOOtDs7ijdWOe8EWcen6h4htYWkdIr4ANK5Zm/drkknqTW5qE+rR6pp6adawzWcjsLySR8NEuOCozz+R/DrWDoGk+LNO1q5uL59Fa3vp/OuRCZS4+XHyZGOw65re1CHVpNU099OuoIbON2N5HImWlXHAU4479x+PSn2+Qu/zMbx/aLd6VYLPp11qNul8jz29pGzuybHzjaQe47iqGhXvh/StE1ubw5p9xp15bQGWe0vA4kBVSVJDMePpXUaz/bX2aP8A4R77B5+/5/t2/btx22c5zis3TPDl082oXviO4huLzUIPszpbKViii5+Vc8nqTk1OtmkVpdXPPriewttNi1XQft9/rtuEuL7UlkbykJwWRyTtPXGAPx7VveJdQ0jUfFE9t4leeW0tYIvsunQFt1xLICcgKQSQMDrjn3q9B4R146fDoF5faf8A2FE4y0ETLcTIDkK38Iz3I5+taOr6BqieIl1zw3JYrdPB5E0d8jFGUdGBXkHt71Tt8v8Agf16biV/69Tk7XWb2y8I32mTy3GnKmoxWqSXEgMlrBJhsFuxC5+mfan6Pcadpvi3S5fCdpfRaXdSG1uLiZm8q6YgkFQ3JIIPPHpjnnpH8FPdeF7yyv7xZNRvZ/tUtyI8qJRjACn+EAYx6ZqTT9D1261ezvfE1zYmPTwTbQWCsFZyNu9t3cDoBxQt9fL8v+HE9tPP8yz4pY3q2ehxEhtSlxNj+GBfmkP4jC/8CreVQihVACqMADsKxNLt5rrxJqWq3cMkQTFnaLIpU+WvLOAezMevoorcpLYfUKKKKAKGif8AIKj+rfzq/VDRP+QVH9W/nV+gAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKAKMOh6Tb3QuYNLsorgHIlS3QPn1yBmr1FFABRRRQAUUUUAFFFFABRRRQAVHb20FpGY7WCOFCxYrGgUEnknA7mpKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAoaJ/yCo/q386v1R0UY0mL3Lf8AoRq9QAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAUdG/5BEP/AAL/ANCNXqKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigAooooAKKKKACiiigD/2Q==)![A screenshot of a computer

Description automatically generated](data:image/jpeg;base64,/9j/4AAQSkZJRgABAQEAeAB4AAD/4RDmRXhpZgAATU0AKgAAAAgABAE7AAIAAAAJAAAISodpAAQAAAABAAAIVJydAAEAAAASAAAQzOocAAcAAAgMAAAAPgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAEZpdG9yZSBIAAAABZADAAIAAAAUAAAQopAEAAIAAAAUAAAQtpKRAAIAAAADMTcAAJKSAAIAAAADMTcAAOocAAcAAAgMAAAIlgAAAAAc6gAAAAgAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAADIwMjI6MDI6MjIgMTY6MTU6NDQAMjAyMjowMjoyMiAxNjoxNTo0NAAAAEYAaQB0AG8AcgBlACAASAAAAP/hCxtodHRwOi8vbnMuYWRvYmUuY29tL3hhcC8xLjAvADw/eHBhY2tldCBiZWdpbj0n77u/JyBpZD0nVzVNME1wQ2VoaUh6cmVTek5UY3prYzlkJz8+DQo8eDp4bXBtZXRhIHhtbG5zOng9ImFkb2JlOm5zOm1ldGEvIj48cmRmOlJERiB4bWxuczpyZGY9Imh0dHA6Ly93d3cudzMub3JnLzE5OTkvMDIvMjItcmRmLXN5bnRheC1ucyMiPjxyZGY6RGVzY3JpcHRpb24gcmRmOmFib3V0PSJ1dWlkOmZhZjViZGQ1LWJhM2QtMTFkYS1hZDMxLWQzM2Q3NTE4MmYxYiIgeG1sbnM6ZGM9Imh0dHA6Ly9wdXJsLm9yZy9kYy9lbGVtZW50cy8xLjEvIi8+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczp4bXA9Imh0dHA6Ly9ucy5hZG9iZS5jb20veGFwLzEuMC8iPjx4bXA6Q3JlYXRlRGF0ZT4yMDIyLTAyLTIyVDE2OjE1OjQ0LjE2NjwveG1wOkNyZWF0ZURhdGU+PC9yZGY6RGVzY3JpcHRpb24+PHJkZjpEZXNjcmlwdGlvbiByZGY6YWJvdXQ9InV1aWQ6ZmFmNWJkZDUtYmEzZC0xMWRhLWFkMzEtZDMzZDc1MTgyZjFiIiB4bWxuczpkYz0iaHR0cDovL3B1cmwub3JnL2RjL2VsZW1lbnRzLzEuMS8iPjxkYzpjcmVhdG9yPjxyZGY6U2VxIHhtbG5zOnJkZj0iaHR0cDovL3d3dy53My5vcmcvMTk5OS8wMi8yMi1yZGYtc3ludGF4LW5zIyI+PHJkZjpsaT5GaXRvcmUgSDwvcmRmOmxpPjwvcmRmOlNlcT4NCgkJCTwvZGM6Y3JlYXRvcj48L3JkZjpEZXNjcmlwdGlvbj48L3JkZjpSREY+PC94OnhtcG1ldGE+DQogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgIAogICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgCiAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAgICAKICAgICAgICAgICAgICAgICAgICAgICAgICAgIDw/eHBhY2tldCBlbmQ9J3cnPz7/2wBDAAcFBQYFBAcGBQYIBwcIChELCgkJChUPEAwRGBUaGRgVGBcbHichGx0lHRcYIi4iJSgpKywrGiAvMy8qMicqKyr/2wBDAQcICAoJChQLCxQqHBgcKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKioqKir/wAARCADcAbcDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtRAAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2JyggkKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIWGh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBAQDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVVldYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiImKkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09fb3+Pn6/9oADAMBAAIRAxEAPwDyv7NB/wA8Y/8AvgUfZoP+eMf/AHwKlooAi+zQf88Y/wDvgUfZoP8AnjH/AN8CpaKAIvs0H/PGP/vgUfZoP+eMf/fAqw8UkSoZY2QSLuQspG5ckZHqMgj8KZQBF9mg/wCeMf8A3wKPs0H/ADxj/wC+BUtFAEX2aD/njH/3wKPs0H/PGP8A74FS0UARfZoP+eMf/fAo+zQf88Y/++BUtKVIAJBAPQ460AQ/ZoP+eMf/AHwKPs0H/PGP/vgVLRQBF9mg/wCeMf8A3wKPs0H/ADxj/wC+BUtFAEX2aD/njH/3wKPs0H/PGP8A74FS1asNNvtUuPs+mWdxeTYz5dvE0jY9cAUAUPs0H/PGP/vgUfZoP+eMf/fArX1Tw5rWiKravpV5ZIxwrzwMik+gJGKpXNrPZ3Bgu4XhlUAlJFKkAjI4PsQaAKv2aD/njH/3wKPs0H/PGP8A74FS0UAQtBbqMtFGB/uCmYs/+ecf/fv/AOtRdnhB9arUAWcWf/POP/v3/wDWoxZ/884/+/f/ANaq1aGl6Bq+tsV0fS7y+KnDfZoGkCn3IHFAEGLP/nnH/wB+/wD61GLP/nnH/wB+/wD61O1HSdR0icQ6rYXVjKeiXMLRk/gwFVKALOLP/nnH/wB+/wD61GLP/nnH/wB+/wD61VqKALOLP/nnH/37/wDrUYs/+ecf/fv/AOtVaigCziz/AOecf/fv/wCtRiz/AOecf/fv/wCtVaigCziz/wCecf8A37/+tRiz/wCecf8A37/+tVaigCziz/55x/8Afv8A+tRiz/55x/8Afv8A+tVaigCziz/55x/9+/8A61GLP/nnH/37/wDrVWooAs4s/wDnnH/37/8ArUYs/wDnnH/37/8ArVWooAs4s/8AnnH/AN+//rUYs/8AnnH/AN+//rVWooAuLNAi4QhR6BcU77RF/e/Q1RooAvfaIv736Gj7RF/e/Q1RooAvfaIv736Gj7RF/e/Q1RooAvfaIv736Gj7RF/e/Q1RooAvfaIv736Gj7RF/e/Q1RooAvfaIv736GiqNFAGlRRRQAV1+jeFjZ6THr2u6fcXMMoJsdPjjfddn++5H3Ivfq3QetchXaaJ4wuL3SIdA1XWLvTxApWw1CKd1EHpHKFPzR9geq9sjigDH1ePxBrmoveX+n3ZcgKiJbMqRIOFRFxhVA4ArMuNPvbSMPd2c8CE4DSxMoJ9ORWxquqeK9F1GSy1DWNSjmTB4vXKup5DKQcFSOQR1rLvNZ1TUYRFqGpXd1Grbgk87OAemcE9eTQBSooooAKKKKAJvstx/wA8Jf8Avg1ctTL5P2W9tpntycqQh3RH1X+o71V/tC8/5+5/+/p/xq5aSXTR/aby9uI7ZTjIkbdIf7q/49BQBUvbGWxlCycq43RuBw4/z2qtVu/1CbUJVaUkIg2xpuJ2j6nqfeqlABRRRQBPY2kuoahb2duMzXEqxRj1ZiAP1Nb/AIh1uSymuNA0GeS20m1kMRETbTdspwZZCPvEkZAPAGAKxdH1BtJ1yx1FF3NZ3Mc4X1KMGx+laPizR20vWpJoMyadesbixuQPlliY5HPqM4I6gg0AWfCOtWOjvK+oahqEUbkB7SC1SeC4TusivIoI/D3yDVbxnqlrrXiy71LTrea3srjYbdJ1wwRVCep7qe5rKsbG61K+is7CCS4uJmCxxRjJY1teNJoBqlpplrMk8ekWcdkZo/uu6lmkKnuN7sAe+M0Ac7RRRQBWvP4PxqtVm8/g/Gq1AGv4W0mHWvEENteSPHZxpJcXToPmWGJGkfHuVUge5FaM2vXfirXbTTGkm0/SZJxDb6fYRl1hRmwAsYx5j89Tyx71V8GX1tZ+IhFqMwgs763msp5SOIxLGyBz7KxVj7CpNLgl8H/EbTB4gje1/s3UoZbjKk4RJASwx94YGQR1HSgCXRPExguDo+uyy6hoE8myWKcZaEH5RLHuzsdevBwcYPFY2t6XLoevX+l3DBpLK4eBmAwGKsRkexxmrmjaNJ4l8RtHEfKtN7TXVy4+S3hByzse2B+ZwByai8Uasmu+K9T1OFSkV1cvJEp6qmflH4LgUAZVa3h200W71Fv+Ek1G4sbONN+LaDzZZyGH7tckBSQT8zcDFZNOjOJFz6itKaTmkxPY67xXp+h+GvijqNk9hLNpNnLhLRJiC/7sFVLnJ2lsZI5xnGDip7+z0eTwDp3iq30K3tJ11N7KazE07W90oTeH+Zy4x904f8q1NVtdB8TfG7V5dR1K1Okgtcb1vI41utsa4jWViFBZuM56Z6VheNJNW1ARXF3/AGPbabaZis9P07UraVbZSckBI5CxJPLORknr2Fc9Nv2cb72X9er2/pXp6t/1/X9fK/DFos3w71XXb/wxp9o7SrZaabee6DNMRl3+eZgQi84xgkgVl+GNHsf+Eb1rxJq9sL2DTRHDBaNIyJNNISAXKkNtUAtgEEnHPWp/HN3axab4c0LTLm3ubfT9PEkr28odTcSnfJyD1HyjHbFamm2VraeGfEHhGfW9Iku7tLa+s7mK8H2d3TJaIythVbaehwM8ZrR/at6f5/hd/wBIlfZ+/wC/b9E/mY3izSNOTQNB8Q6NbfY4dVikSa0EjOsM0TBW2liW2tkEAkketa0Wl+HPD58MWGt6SuozazCl1eXLXEiNbxyttjEQRguVA3HcGyeOBSatbWmoaH4e8I22s6ZHcaZb3F3eXU10BbiSQhxEsi5DMAAOMgk4B61amj0zxS3hLVv7XsLOHTrOKz1OK5ukjkhEBzuVCd0m5TxsDc8VS5b69/w1/C4Ntr3d7fj0/A47XtCTQvGd7olxcbIba7MJuGXdhN3DkDr8uDgVveM/D/hzS/B+gX3hma5vPtc1zHNeXCmMzmMqoKx5O1epAPzc81Q8YTp4i1LU/FkN3aLFeak8Udk0v+khNuVkKf3cADOevFTa5e2s3wu8K2kVzC9zBPeGWFZAXjDOu0svUZxxnrWcebkSlvp+hpp7R9tfyZnado0sc2oxXenG6uLaBWSAFm+ZmXH3DzwSetQ21pHdapKLuyFnHaQtLNAhdSdozj5ySCSQKrNpT/aJYYruykMW35hcqqtkZ+UsQDjoa2hf2v2mCynukd2sHtJbrdlVcklRu7gcDPT8Krp/XmR/X5FSDTrfWbWCS0gS0m+1JbyIjMVKtkhvmJORg96Rbew1Fb+CztRA9rGZYJRIxMiqcENk4yQc8AdKmsLuLQbeBJpI5ZnvI5pFhkWQJGmepUkZJJ49qZEsOjnUbgXMEyzRNDbCKUMz7j1IByuAOc45o0/r5fqH9fiyRdPsDrDaH9nHmhCgu97bvNAz0zt25+XGM471znSuo821j8RtrouYTa4M6xiUeYXIxs2Z3Z3d8YxzmuXJyc0teoBRRRTAKKKKACiiigAooooAKKKKACiiigDSooooAKKKKAJZrme4WITzSSiFPLjDsTsXJO0Z6DJJx7moqKKACiiigAooooAKcXZlVWYlV+6Cen0ptFABRRRQAUUUUAFbGk+KNS0i2NpGYLqyZt7Wd7As8O71CsDtPuuDWPRQB0Nx401FrWS2063sdIimUpL/AGdbLE8inqpk5fHtnHtXPUUUAFFFFAFa8/g/Gq1X5IhKuDx6Gofsf+3+lAFat+y8ZalbWMVjeRWWqWkIxDFqNss3lD+6jn51HsGArJ+x/wC3+lH2P/b/AEoA0dU8Wanqlj9gP2azsNwc2ljbpBG7DoWCgFyOxYnFYtWfsf8At/pR9j/2/wBKAK1FWfsf+3+lH2P/AG/0oArUVZ+x/wC3+lH2P/b/AEoArUVZ+x/7f6UfY/8Ab/SgCtRVn7H/ALf6UfY/9v8ASgCtRVn7H/t/pR9j/wBv9KAK1FWfsf8At/pR9j/2/wBKAK1FWfsf+3+lH2P/AG/0oArUVZ+x/wC3+lH2P/b/AEoArUVZ+x/7f6UfY/8Ab/SgCtRVn7H/ALf6UfY/9v8ASgCtRVn7H/t/pR9j/wBv9KAK1FWfsf8At/pR9j/2/wBKAK1FWfsf+3+lH2P/AG/0oArUVZ+x/wC3+lFAH1x/wqXwP/0AIf8Av9J/8VR/wqXwP/0AIf8Av9J/8VXY0UAcd/wqXwP/ANACH/v9J/8AFUf8Kl8D/wDQAh/7/Sf/ABVdjRQBx3/CpfA//QAh/wC/0n/xVH/CpfA//QAh/wC/0n/xVdjRQBx3/CpfA/8A0AIf+/0n/wAVR/wqXwP/ANACH/v9J/8AFV2NFAHHf8Kl8D/9ACH/AL/Sf/FUf8Kl8D/9ACH/AL/Sf/FV2NFAHHf8Kl8D/wDQAh/7/Sf/ABVH/CpfA/8A0AIf+/0n/wAVXY0UAcd/wqXwP/0AIf8Av9J/8VR/wqXwP/0AIf8Av9J/8VXY0UAcd/wqXwP/ANACH/v9J/8AFUf8Kl8D/wDQAh/7/Sf/ABVdjRQBx3/CpfA//QAh/wC/0n/xVH/CpfA//QAh/wC/0n/xVdjRQBx3/CpfA/8A0AIf+/0n/wAVR/wqXwP/ANACH/v9J/8AFV2NFAHHf8Kl8D/9ACH/AL/Sf/FUf8Kl8D/9ACH/AL/Sf/FV2NFAHHf8Kl8D/wDQAh/7/Sf/ABVH/CpfA/8A0AIf+/0n/wAVXY0UAcd/wqXwP/0AIf8Av9J/8VR/wqXwP/0AIf8Av9J/8VXY0UAcd/wqXwP/ANACH/v9J/8AFUf8Kl8D/wDQAh/7/Sf/ABVdjRQBx3/CpfA//QAh/wC/0n/xVH/CpfA//QAh/wC/0n/xVdjRQBx3/CpfA/8A0AIf+/0n/wAVR/wqXwP/ANACH/v9J/8AFV2NFAHHf8Kl8D/9ACH/AL/Sf/FUf8Kl8D/9ACH/AL/Sf/FV2NFAHHf8Kl8D/wDQAh/7/Sf/ABVH/CpfA/8A0AIf+/0n/wAVXY0UAcd/wqXwP/0AIf8Av9J/8VR/wqXwP/0AIf8Av9J/8VXY0UAcd/wqXwP/ANACH/v9J/8AFUf8Kl8D/wDQAh/7/Sf/ABVdjRQBx3/CpfA//QAh/wC/0n/xVH/CpfA//QAh/wC/0n/xVdjRQBx3/CpfA/8A0AIf+/0n/wAVR/wqXwP/ANACH/v9J/8AFV2NFAHHf8Kl8D/9ACH/AL/Sf/FUf8Kl8D/9ACH/AL/Sf/FV2NFAHHf8Kl8D/wDQAh/7/Sf/ABVH/CpfA/8A0AIf+/0n/wAVXY0UAcd/wqXwP/0AIf8Av9J/8VR/wqXwP/0AIf8Av9J/8VXY0UAcd/wqXwP/ANACH/v9J/8AFUf8Kl8D/wDQAh/7/Sf/ABVdjRQBx3/CpfA//QAh/wC/0n/xVH/CpfA//QAh/wC/0n/xVdjRQBx3/CpfA/8A0AIf+/0n/wAVRXY0UAFFFcL4w8QeKNK8beHdJ0W50hLTW5ZId13YyyyQGOMuWysyhgcYxgY9TR1SA7qiuH8X654p0rxh4e03RrrR0tNameAm7sJZJIWSMuWysyhgduMYGPU11V5rmlabd29pqWqWVrc3JxBDPcJG8p/2VJyfwoWobF6iuWm1jUI/iza6KJ1/s+bRprsxeWM+as0ag7uvRjxSeB9cvdWs9ck1e4ST7DrN3aRvsCBYo2AXOPbuaFqr+Tf3O35g9PvS+9XOqoqjpmt6VrcckmjanZ6gkTbZGtLhJQh9CVJwa5j4keIfEPhe00/UdFl0xNPe8itr9720klNusjhRMNsqDAJAIPrnNHVLv+oHa0Vw2peIvEemfFPRtDnvNFXSNWjleFjaSfaC0QUtHnzsZIJIbbjjGD32Yr3WT4yvYpbvTRoVpbo7/wCjOsyyMD8pkMm3AADE7BwwHvR0uB0FFUNO13SdYt5LjSdUsr6GIlZJLW4SRUPoSpIFZPiPxRaxeEtYutB1aylv7XTZryERypKcIrYbaDyu5cZ6Z4pSfKrsqMXKSijpaKxdP123t/Bunavr+oW9oklpDJNc3MixJuZASSTgDJNc1p3jDU5PDXjfVIp7XVG0e5m/s/ylAjeNbaOVRlT8wyx5zk+tVJcrkn9nX8Uv1Jh76i11/U7+iqGhXs+peHdOvryJYbi6tY5pY0zhGZQSBn3NX6JJxdmKLUldBRRRSGFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABXknxV1Xw+fiN4IsNY1mGzWG5uJLrZqBtpIEaEhWZ0ZWQE8A5Gelet0UdUPozx3xc/hbwp8TvBQfXpIpYruZ7ldS1ye4EMbW7hWImlYICSAG4zTNaWwuvEfiXw94v8Qa1pf8Aa1xutrW2tIJI7+AxqF8pzbu5YYI2hsgjIA617LRSS7+f6f5Cfl/W/wDmeZx6np2n/G7QtOu9SiW7j8NyW5juLhDKZDLEwViMZcqrHoM4JArl7XVYtU8E+JToVxHqcdr4va81C1s3Ery2X2hWYhVyWVlUn0YBhzXulFUm1byv/wClc3/ADS1vT8IuP6nn0Utj4h+Kmia54Sure7tobC5i1O6tGDRspKeVEzDjeG3HaeQAema6/wAQ6Lb+I/DeoaPejMF9bvC3tkYBHuDz+FaVFS0nHlGm07njjab4g1j4VHXNStmTxJoDRtaqesj2TsHP/bX94PcFa2/FUOpL4AsNTks5ZPO1S21HWLRFLubfeCyYH3tihARjkRmvSKKq7vfzT/z++yuKytb5fLp92p5vLf6PqPxIs/E+i6hZvpFtpNwmsahHKptyu5PKR3zt3KQ5weVGc4yK5rwvceHF/ZovNSik01bqHSLu0muQUEiO+7ETN1BYlMKevy+1e20VLV4uP9df8yoy5ZqXn+Vv8jyTTtfsIdW8D6zqF7bt4cXQ2t4b0uDBBe/ux879EbarqCcYO4dTWl4Wk0m2sfiGJMW2m/2nPNI6Qts8traLc4AHzZO48da9JoqpPm5vNNfe7/8AA9CILkUV2t+CsVdMe3k0m0eybdbNAhhbaVym0bTg8jjHBq1RRQ3d3BKysFFFFIYUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQAUUUUAFFFFABRRRQB//9k=)

*Figura 31 Fajlli kryesor i ruajtur si server.py dhe Ndërfaqja e aplikacionit “Hello World”*

## Aplikacioni në Jupyter Notebook

Datasetin dhe algoritmin e përdorur në këtë aplikacion e kemi vendosur edhe në Jupyter Notebook për të qenë më të saktë dhe për të sqaruar në detaje për to.

Jupyter Notebook është një ueb aplikacion tek i cili shohim të gjithë procesin e llogaritjes: zhvillimin, dokumentimin dhe ekzekutimin e kodit, si dhe gjithashtu komunikimin e rezultateve. I zhvilluar fillimisht si pjesë e projektit IPython në 2014, Jupyter Notebook ka fituar një popullaritet të jashtëzakonshëm midis shkencëtarëve të të dhënave, studiuesve, edukatorëve dhe zhvilluesve të programeve kompjuterike për shkathtësinë dhe lehtësinë e përdorimit.

Për të përdorur këtë, vetëm duhet të shënoni komandën *jupyter notebook* në shfletues (angl. browser), dhe më pas hapet një shfletues uebi në URL-në e aplikacioni në internet *https://jupyter.org/try-jupyter/lab/*

**Eksplorimi i Ndërfaqes së Jupyter Notebook**

Kur ne krijojmë një dokument të ri në notebook duhet të caktojmë emrin e atij dokumenti dhe kemi pjesën (angl. code cell) ku mund të shkruajmë kodin për ta ekzekutuar më pas.

A screenshot of a computer

Description automatically generated

*Figura 32 Ndërfaqja e Jupyter Notebook*

**Emri i fletores (angl. Notebook name)**: Shihet se është krejt në fillim të faqes, dhe tregon emrin e fajllit të ruajtur në formatin ***.ipynb*** .

**Shiriti Menu (angl. Menu bar)**: Menu bar na paraqet opsione të ndryshme të cilat i përdorim për të manipuluar në dokument.

**Shiriti i veglave (angl. Tool bar)**: Tool bar na jep një qasje më të shpejtë në operacionet më të përdorura në dokument.

**Hapësira e kodit (angl. Code cell)**: Lejon të ndryshojë dhe të shkruaj kodin. Gjuha programuese që ju përdorni varet nga Kernel si në rastin tonë kernel ekzekuton kodin në Python.Në vazhdim do të shtjellojmë dokumentin tonë të krijuar në Jupyter Notebook.

Si fillim bëjmë importimin e paketave të nevojshme për ekzekutim si dhe leximi i të dhënave (datasetit) dhe shfaqja e pesë rreshtave të parë:

A screenshot of a computer program

Description automatically generated A screenshot of a computer

Description automatically generated

*Figura 33 Importimi i librarive dhe rreshtimi i 5 rreshtave të parë*

Informata më të detajuara rreth atributeve, shohim se asnjë kolonë e përdorur nuk e ka vlerën null:

A screenshot of a computer

Description automatically generated

*Figura 34 Tipi i të dhënave të atributeve*

Po ashtu është krijuar një histogram të të dhënave në DataFrame-n df, me 40 bin-e dhe një madhësi figure prej 20x15 përmes kodit:

*df.hist(bins=40, figsize=(20,15))*

*plt.show()*

A group of blue and white graphs

Description automatically generated

*Figura* ***35*** *Histogrami i të dhënave*

Para se të aplikohet algoritmi së pari ndahen atributet në ato predikuese dhe në atributin e synuar:

A screenshot of a computer code

Description automatically generated

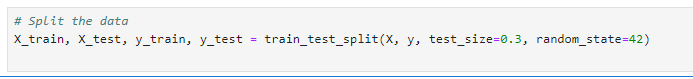
*Figura 36 Ndarja ne Train dhe Test*

Këtu tek parametrat e funksionit **train\_test\_split**, **train\_size** e kemi caktuar **0.7** që do të thotë se **70%** të të dhënave do të jenë të dhëna për trajnim e ndërsa pjesa tjetër (respektivisht **30%**) të të dhënave janë testuese që përdoren për trajnimin e modelit.

Rezultati i kodit ***((717, 7), (308, 7))*** tregon dimensionet e grupeve të ndara:

* **(717, 7)**: Kjo tregon se grupi i trajnimit (X\_train) ka 717 rreshta dhe 7 kolona. Kjo do të thotë se ke 717 mostra të dhënash për të trajnuar modelin dhe secila mostra ka 7 karakteristika (ang. features).
* **(308, 7)**: Kjo tregon se grupi i testimit (X\_test) ka 308 rreshta dhe gjithashtu 7 kolona. Këto 308 mostra do të përdoren për të testuar modelin pas trajnimitt.

Kjo ndarje tregon se modeli do të trajnohet me 70% të të dhënave dhe do të testojë performancën e tij me 30% të të dhënave, duke ndihmuar në vlerësimin e saktësisë së tij.

Ose ndarja mund të bëhet edhe kështu:  


Tani bëjmë trajnimin e modelit

A screenshot of a computer program

Description automatically generated

*Figura 37 Tajnimi i modelit RF*

Me anë të vlerësimit Out-of-Bag (OOB), i cili përdoret për të vlerësuar performancën e modelit në mënyrë të brendshme pa nevojën për një grup të veçantë testimi shohim:   
A white rectangular object with blue text

Description automatically generated  
Një rezultat OOB prej 86,33% sugjeron që modeli juaj parashikon saktë klasën për rreth 86,33% të mostrave në grupin tuaj të të dhënave. Kjo përgjithësisht konsiderohet një performancë e mirë, në varësi të kontekstit të problemit tuaj.

Pjesa e kodit është konfiguruar duke përdorur hiperparametrat për një model RF me Grid Search me verifikim të kryqëzuar

A screenshot of a computer program

Description automatically generated

*Figura 38 Hiperparametrat e një modeli RF*

**"Fitting 4 folds for each of 180 candidates, totaling 720 fits"**:

* Kjo nënkupton që po kryhen 720 përshtatje për të gjetur parametrat më të mirë.
* **4 folds**: Kjo do të thotë se ke përdorur validimi me 4. Pra, të dhënat janë ndarë në 4 pjesë, dhe për çdo kombinim të parametrave, modeli trajnohet 4 herë (për çdo pjesë si grup testimi një herë).
* **180 candidates**: Kjo tregon se ke 180 kombinime të mundshme parametrash që do të shqyrtohen nga GridSearchCV. Ky numër vjen nga shumëzimi i mundësive për secilin parametrar në grupin e parametrave që ke caktuar.

Kështu duket një vizualizim i pemës vendimmarrëse nga modeli Random Forest.  
A diagram of a company

Description automatically generated

*Figura 39 Vizualizimi i një peme nga modeli RF*

A white box with black text

Description automatically generatedMënyra më efikase për të ruajtur dhe ngarkuar objekte të ndërlikuara si modelet e të mësuarit të makinës në këtë rast modelin e trajnuar të Random Forest në një skedar të formatit .*pkl*, bëhet duke përdorur bibliotekën joblib.

Modeli *pkl* me vonë do e përdorim për të testuar nëse personat kanë apo jo sëmundje.   
Po ashtu kemi bërë edhe ruajtjen e pemës në pdf file.

Po ashtu, është bërë edhe testimi i saktësisë së metrikave të modelit RF

Confusion Matrix:

* Negativë të vërtetë (TN): 145 (negativë të parashikuar saktë)

[[145 0]

[ 0 163]]

* Pozitive të rreme (FP): 0 (pozitive të parashikuara gabimisht)
* Negativë të rremë (FN): 0 (negativë të parashikuar gabimisht)
* Pozitive të vërteta (TP): 163 (pozitive të parashikuara saktë)

Kjo do të thotë se të gjitha rastet janë klasifikuar saktë.

Precision:

[1. 1.] tregon se modeli ka saktësi të përsosur për të dyja klasat (0 dhe 1). Kjo do të thotë se çdo rast i parashikuar si pozitiv është në të vërtetë pozitiv.

Recall

[1. 1.] tregon se modeli ka rikujtim perfekt për të dyja klasat. Kjo do të thotë se të gjitha rastet aktuale pozitive janë identifikuar saktë.

A screenshot of a computer program

Description automatically generated

*Figura 40 Metrikat e modelit RF*

F-Score:

[1. 1.] tregon rezultate të përsosura F, që do të thotë se modeli balancon saktësinë dhe rikujtimin në mënyrë të përsosur.

Support:

[145 163] tregon numrin e dukurive aktuale të secilës klasë në grupin e testimit (145 raste të klasës 0 dhe 163 raste të klasës 1).

Së pari është bërë serializimi i fajllit në Jupyter Notebook në mënyrë që të mund ta

përdorim atë në Flask dhe pastaj të bëjmë lidhjen me ndërfaqjen grafike.

## Zhvillimi i Aplikacionit

Aplikacioni “Aplikacioni i Parashikimit të Sëmundjeve të Zemrës” përcakton duke pas parasysh disa atribute se personi a vuan nga sëmundja e zemrës apo jo. Ky aplikacion realizohet duke përdorur datasetin të marr nga Kaggle, i cili ka disa atribute të predikimit mjekësor dhe një atribut i synuar.

Në editorin VS Code është filluar puna e lidhjes së modelit dhe csv file me të dhëna ndërfaqen e aplikacionit me anë të Flask.

Fajlli kryesor i këtij aplikacionit është app.py, përmban importimin e Flask nga paketa flask, thirrjen e modelit nga metoda **joblib,** ku me *joblib* lexohen të dhënat e ruajtura nga një fajll i formatit ***.pkl*** dhe kjo ruhet, ngarkohet, në një variabël të tipit dataframe. Përveq këtyre përmban edhe funksionin **home()** i cili del si output kur faqja kryesore e webserverit hapet ne shfletues (angl. browser). Ky funksion dhe url-ja lidhet përmes route(‘/’), dhe return i bie që localhosti do të paraqet pamjen vizuale të index.html fajllit.

Të treguara edhe në figurën më poshtë:

A screen shot of a computer program

Description automatically generated

*Figura 41 Kodi i aplikacionit në Flask*

Parashikimi i këtyre rezultateve me anë të këtij algoritmi bëhet duke thirrur route predict

i cili edhe është route kryesor i këtij aplikacioni.

Funksioni **route predict** është i dizajnuar për të interpretuar informacionin në mënyrë dinamike, duke u përshtatur me ndryshimet e mundshme në të dhëna dhe duke ofruar parashikime të mençura.

A computer screen with text

Description automatically generatedRezultati varet nga informacionet që i jep përdoruesi në bazë të atributeve të përcaktuara në dataset.

*Figura 42 Thirrja e route për predikim*

### Ekzekutimi i Aplikacionit

Aplikacioni është i hostuar lokalisht në kompjuter me anë të VS Code dhe mund të hapet përmes terminalit me http://127.0.0.1:5000.

Pamja vizuale e aplikacionit duket si më poshtë:  
  
A screenshot of a screen

Description automatically generated

*Figura 43 Faqja kryesore Index.html*

Momentin që klikojmë në butonin “ Predict Heart Disease”, na hapet një faqe e re në mënyre që të kemi mundësinë me dhënë informatat e nevojshme për predikim/detektim.

A screenshot of a computer

Description automatically generated

*Figura 44 Ndërfaqja e Web Aplikacionit*

Gjithashtu janë bërë validimet e field-ave kur përdoruesi dëshiron të përcaktojë rrezikshmërin e semundjes, ai nuk mund të vazhdojë të kryejë këtë proces pa i plotësuar të gjitha informacionet:

A screenshot of a computer screen

Description automatically generated

*Figura 45 Validimet në Aplikacion*

Përdoruesi jep të dhënat kryesore që ndikojnë në predikimin e sëmundjes që ju nevojiten, siç janë:

* Age - Mosha
* Sex - Gjinia
* Cp (ang. Chest Pain) - Dhimbja e Kraharorit
* Trestbps (ang. Resting Blood Pressure) - Presioni i Gjakut
* Chol (ang. Cholesterol) - Kolesteroli
* Fbs (ang. Fasting Blood Sugar) - Sheqeri në Gjak
* Thalach (ang. Maximum Heart Rate Achieved) - Ritmi Maksimal i Zemrës

Vlerat e tyre:

* Mosha: tregon moshën e individit.
* Gjinia : shfaq gjininë e individit duke përdorur formatin e mëposhtëm:
* 1 = mashkull
* 0 = femër.
* Lloji i dhimbjes së gjoksit: shfaq llojin e dhimbjes së gjoksit të përjetuar nga individi duke përdorur formatin e mëposhtëm:
* 0 = anginë tipike
* 1 = anginë atipike
* 2 = dhimbje jo-anginale
* 3 = asimptotike
* Presioni i gjakut: tregon vlerën e presionit të gjakut të një individi në mmHg (njësi)
* Kolestroli: shfaq kolesterolin në mg/dl (njësi)
* Sheqeri i gjakut : krahason vlerën e sheqerit në gjak të një individi me 120 mg/dl. Nëse sheqeri në gjak > 120 mg/dl atëherë:
* 1 (e vërtetë)
* 0 (e gabuar)
* Frekuenca maksimale e arritur e zemrës: tregon rrahjet maksimale të zemrës të arritura nga një individ.

Në vijim është shfaqur rezultati që jep aplikacioni pasi që i kemi dhënë të dhënat:

A screenshot of a medical form

Description automatically generated A screenshot of a test results

Description automatically generated

*Figura 46 Rezultati pas dhënies së informacioneve*

# Përfundimet

Në këtë punim të diplomës është bërë një studim rreth algoritmit Random Forest ose pyjeve të rastësishme, duke treguar për vetitë, llojet, avantazhet dhe disavantazhet e tyre. Gjithashtu kemi shpjeguar algoritmet në përgjithësi duke analizuar dhe historikun e tyre.

Me anë të këtij projekti unë kam sqaruar konceptet kryesore rreth Data Mining, si dhe për shumicën e algoritmeve që bëjnë pjesë në të, po ashtu është bërë edhe shtjellimi i metodave të asamblimit në mënyrë që ta kuptojmë më mirë algoritmin Random Forest. Gjatë këtij punimi qartë shohim dallimin në mes këtyre algoritmeve si dhe përdorimin e tyre.

Me pak fjalë unë kam përforcuar konceptet themelore të funksionimit të algoritmit Random Forest, por edhe njohja dhe përdorimi i gjuhës Python duke përdorur kornizën Flask, si dhe editorin Visual Code.

Krejt në fund të këtij punimi shohim implementimin e algoritmit në një web aplikacion, jam munduar që ta bëjë implementimin e kodit duke i’u përshtatur programimit modern për zgjidhjet më adekuate në mënyrë që kodi të jetë sa më i kuptueshëm megjithëse mendoj se ka vend për përmisime.

Është bërë strukturimi i projektit në atë mënyrë që nëse një fillestar dëshiron të dijë më shumë rreth algoritmit Random Forest, atëherë nuk do e kanë problem.

# Conclusions

In this diploma thesis, a study is made about the Random Forest algorithm or random forests, showing their properties, types, advantages and disadvantages. We have also explained the algorithms in general by analyzing their history.

Through this project I have clarified the main concepts about Data Mining, as well as for most of the algorithms that are part of it, as well as the elaboration of assembly methods in order to better understand the Random Forest algorithm. During this paper we clearly see the difference between these algorithms as well as their use.

In short I have reinforced the basic concepts of the operation of the Random Forest algorithm, but also the knowledge and use of the Python language using the Flask framework, as well as the Visual Code editor.

At the very end of this paper we see the implementation of the algorithm in a web application, I have tried to make the implementation of the code adapting to modern programming for the most adequate solutions so that the code is as understandable as possible although I think there is room for adjectives.

The project has been structured in such a way that if a beginner wants to know more about the Random Forest algorithm, then they will have no problem.

# Shtesat

**Për krijimin e dokumentit janë përdorur veglat:**

Microsoft Word

Google Docs

Python

Jupyter Notebook

Visual Studio Code

**Lista e Shkurtesave**

RF – Random Forest

ML - Machine Learning

DAA - Dizajni dhe Analiza e Algoritmeve

MDI - Mean Decrease in Impurity

MDA - Mean Decrease Accuracy

AUC-ROC - Area Under the Curve - Receiver Operating Characteristic

Web - website

Cp - (Chest Pain)

Trestbps - (Resting Blood Pressure)

Chol - (Cholesterol

Fbs - (Fasting Blood Sugar)

Thalach - (Maximum Heart Rate Achieved

VS Code/VSC - Visual Studio Code

OOB - out of bag

**Lista e Figurave**

*Figura 1 Figurë e thjeshtë e algoritmit*

*Figura 2 Përmbledhje e algoritmeve në përgjithësi*

*Figura 3 Modeli bazë*

*Figura 4 Metodat e Ansamblimit*

*Figura 5 Modeli i Bagging (Bootstrap Aggregating)*

*Figura 6 Predikimi në bazë te Boosting*

*Figura 7 Parashikimet e modeleve të stacking*

*Figura 8 Figurë e thjeshtë e Random Forest Algoritmit*

*Figura 9 Funksioni i pemëve vendimmarrëse në pyjet e rastësishme*

*Figura10 Klasifikimi i i pyjeve të rastësishme*

*Figura 11 Regresioni i pyjeve të rastësishme*

*Figura 12 Pseudokodi i RF*

*Figura 13 Hapat e RF*

*Figura 14 Struktura e Decision Tree*

*Figura 15 Struktura e Random Forest*

*Figura 16 Decision Tree vs Random Forest*

*Figura 17 Analogjia e RF në jetën reale*

*Figura 18 Mean Decrease in impurity(MDI)*

*Figura 19 Mean Decrease Accuracy (MDA)*

*Figura 20 Accuracy*

*Figura 21 Precision*

*Figura 22 Recall*

*Figura 23 F1 Score*

*Figura 24 AUC-ROC*

*Figura 25 Dataseti me 1026 të dhëna*

*Figura 26 Karakteristikat e Python*

*Figura 27 Editori Visual Studio Code*

*Figura 28 Framework(kornizat) e Python*

*Figura 29 Përdorimi i Flask në krahasim me kornizat tjera*

*Figura 30 Struktura kryesore e një fajlli në Flask*

*Figura 31 Fajlli kryesor i ruajtur si server.py dhe Ndërfaqja e aplikacionit “Hello World”*

*Figura 32 Ndërfaqja e Jupyter Notebook*

*Figura 33 Importimi i librarive dhe rreshtimi i 5 rreshtave të pare*

*Figura 34 Tipi i të dhënave të atributeve*

*Figura 35 Histogrami i të dhënave*

*Figura 36 Ndarja ne Train dhe Test*

*Figura 37 Tajnimi i modelit RF*

*Figura 38 Hiperparametrat e një modeli RF*

*Figura 39 Vizualizimi i një peme nga modeli RF*

*Figura 40 Metrikat e modelit RF*

*Figura 41 Kodi i aplikacionit në Flask*

*Figura 42 Thirrja e route për predikim*

*Figura 43 Faqja kryesore Index.html*

*Figura 44 Ndërfaqja e Web Aplikacionit*

*Figura 45 Validimet në Aplikacion*

*Figura 46 Rezultati pas dhënies së informacioneve*

**Lista e Tabelave**

*Tabelë 1. Decision Tree vs Random Forest*

*Tabelë 2 Tabela e metrikave*

# Referencat

[0] – Algoritmet dhe strukturat e të dhënave, Avni Rexhepi

Link: avnirexhepi - Algoritmet dhe Strukturat e të Dhënave (google.com)  
ASDh-Libri-2019-2020.pdf - Google Drive [Qasja: 25.09.2023]

[1] - L. Meinecke, "What is an Algorithm in Programming?"   
 Link: https://study.com/academy/lesson/what-is-an-algorithm-in-programming-definition-examples-analysis.html [Online]. [Qasja: 25.09.2023]

[2] - B. MEHRI, From Al-Khwarizmi to Algorithm, 2017. [Online]  
 Link: https://jurnal.institutsunandoe.ac.id/index.php/prevenire/article/view/186 [Journal] [Qasja: 25.09.2023]

[3] - Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. [Qasja: 26.09.2023]

[4] - Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley. [Qasja: 26.09.2023]

[5] - Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques (3rd ed.). Burlington, MA: Morgan Kaufmann Publishers [Qasja: 9.10.2023]

[6] - Elmasri, R., & Navathe, S. B. (2015). Fundamentals of Database Systems. [Qasja: 9.10.2023]

[7] - Kimball, R., & Ross, M. (2013). The Data Warehouse [Qasja: 25.09.2023]

[8] - Mitchell, T. M. (1997). Machine learning. [Qasja: 10.10.2023]

[9] - Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques. [Qasja: 10.10.2023]

[10] - Quinlan, J. R. (1986). Induction of decision trees. Machine learning, 1(1), 81-106. [Qasja: 29.10.2023]

[11] - Zhou, Z.-H. (2012). Ensemble Methods: Foundations and Algorithms. Qasja: 29.10.2023]

[12] - Hastie, T., Tibshirani, R., Friedman, J. (2009). The Elements of Statistical Learning: Data Mining, Inference, and Prediction. [Qasja: 02.12.2023]

[13] - Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32. [Qasja: 02.12.2023]

[14] - Fisher, R.A. (1959). An algorithm for constructing decision trees for the classification of biological organisms. Journal of the Royal Statistical Society. Series C (Applied Statistics), 8(1), 1-16. [Qasja: 05.12.2023]

[15] - Ho, Tin Kam. "Random Decision Forests". Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, Quebec, Canada, August 14–16, 1995. [Qasja: 24.12.2023]

[16] - Breiman, Leo. "Random Forests". Machine Learning, 45(1), 5-32, 2001. . [Qasja: 24.12.2023]

[17] - Breiman, L., Friedman, J., Olshen, R., & Stone, C. (1984). Classification and regression trees. . [Qasja: 15.01.2024]

[18] "Introduction to Machine Learning" by Ethem Alpaydin [Qasja: 15.01.2024]

[19] - https://towardsdatascience.com/random-forests-algorithm-explained-with-a-real-life-example-and-some-python-code-affbfa5a942c [Qasja: 16.01.2024]

[20] - Liaw, Andy, and Matthew Wiener. "Classification and regression by randomForest." R news 2.3 (2002): 18-22 [Qasja: 16.02.2024]

[21] - Amit, Y., & Geman, D. (1997). Shape quantization and recognition with randomized trees. Neural Computation, 9(7), 1545-1588. [Qasja: 15.08.2024]

[22] - https://dataheadhunters.com/academy/decision-trees-vs-random-forests-comparing-predictive-power/ [Qasja: 15.08.2024]

[23] - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10443071/ [Qasja: 16.08.2024]

[24] - Louppe, Gilles, et al. "Understanding variable importances in forests of randomized trees.2013 [Qasja: 16.08.2024]

[25] - Brownlee, J. (2021). How to Calculate Precision, Recall, F1, and More for Deep Learning Models. Retrieved from https://machinelearningmastery.com/how-to-calculate-precision-recall-f1-and-more-for-deep-learning-models/ - Raschka, S. (2020). [Qasja: 25.08.2024]

[26] – World Health Organization. Cardiovascular diseases (CVDs). Retrieved from https://www.who.int/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)2 Krittanawong, C., et al. (2017). [Qasja: 25.08.2024]

[27] - https://www.kaggle.com/datasets/johnsmith88/heart-disease-dataset/data [Qasja: 26.08.2024]

[28] – Python Mini Reference: A Quick Guide to the Modern Python Programming Language for Busy Coders -By Harry Yoon [Qasja: 14.09.2024]

https://www.google.al/books/edition/Python\_Mini\_Reference/3sWoEAAAQBAJ?hl=en&gbpv=1&dq=python+language&printsec=frontcover [https://www.w3schools.com/python/python\_intro.asp] [https://www.python.org/doc/essays/blurb/?external\_link=true]

[29] – Flask - https://flask.palletsprojects.com/en/3.0.x/ [Qasja: 14.09.2024]

[30] - Jupyter Notebook Documentation Release [Qasja: 14.09.2024]