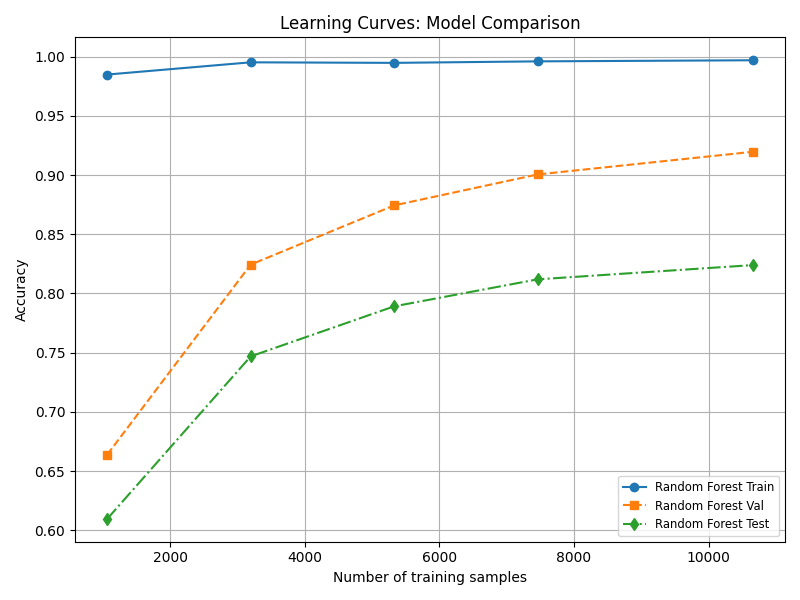
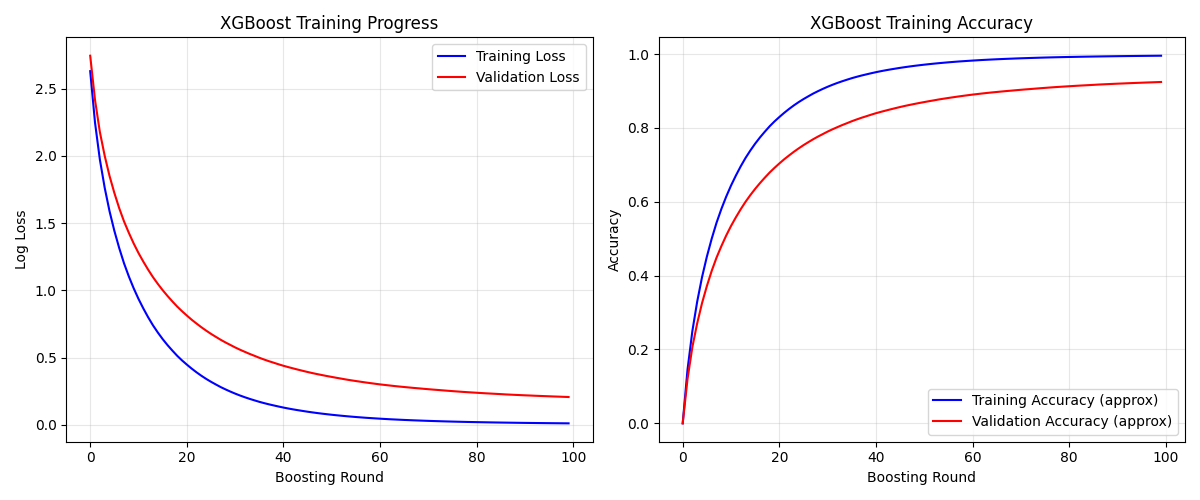
Baigiamojo Projekto Aprašymas

Eismo Ženklu Atpažinimo Modelio Treniravimas

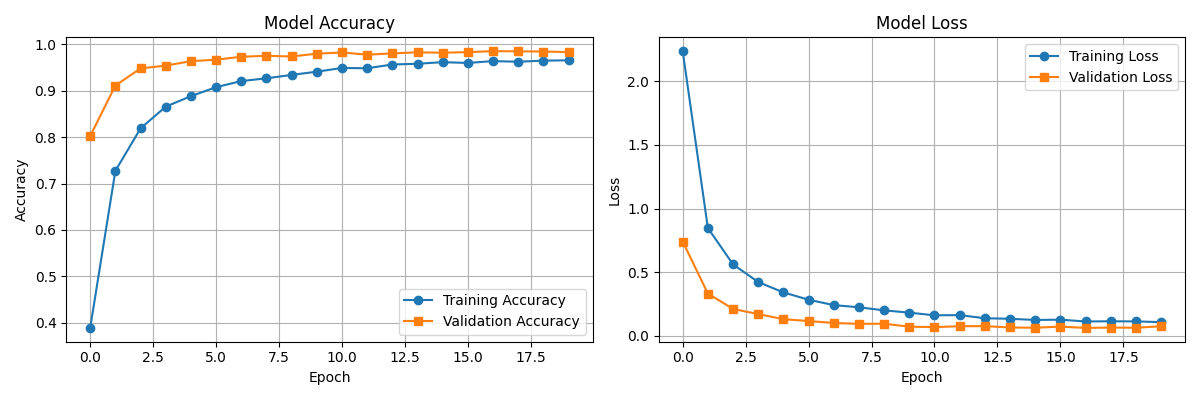
Mano baigiamojo projekto tema yra eismo ženklu atpažinimo modelio treniravimas. Treniravimui parsisiunčiau duomenų rinkinį iš duotos interneto svetaines. Pirmiausia užsikroviau duomenys ir juos susitvarkiau naudodamas tokias bibliotekas kaip opencv, pandas, numpy, pillow ir taip toliau. Pasiruošęs ir išsiskaidęs duomenys i treniravimo ir testavimo rinkinius tai pat juos normalizavęs. Perėjau i modeliu pasirinkimo etapus išbandžiau įvairius modelius tokius kaip randomforest, kneighborsclassifier , svm, xgboost. Stebėjau jų rezultatus keičiau parametrus tiek rankiniu būdu tiek naudodamas gridboost apsunkino kompiuterio našumą ir su juo nepavyko gauti geru rezultatu. Todėl pasirinkau keletą modeliu Pabandžius paleisti KNC modeli tai pat susidūriau su našumo problemomis todėl likau prie pasirinkimo su keliais modeliais randomforest ir xgboost.



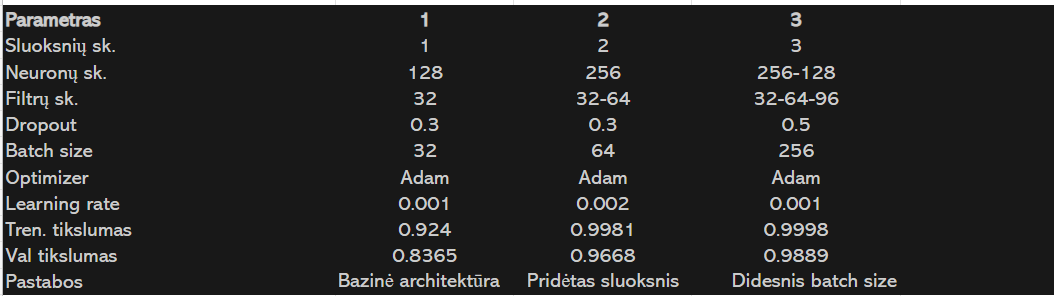
Palyginti randomforest treniravimo, validacijos ir testavimo našumui palyginti pasirinkau mokymosi kreive. Treniravimo kreive stabili ir nutolusi nuo validacijos ir testavimo kreiviu. Dėl to manau gali būti overfitas. Tuo tarpu testavimo ir validacijos kreives gauna geresnius rezultatus didėjant duomenų kiekiui. Tai pat išbandžiau pakeisti modelio hyperparametrus bet rezultatai nedaug kuo skiriasi treniravimo linija išliko tokia pati o validavimo ir testavimo kreives siek tiek pakilo didėjant duomenų kiekiui . Kitas modelis kuri pabandžiau naudoti buvo xgboost.



XgBoost pirmame grafike pavaizduotas treniruotes progresas jame lyginama loginis praradimas ir boosting round. Galima matyti tai kad mažėjant loginiam praradimui didėja atstumas tarp linijų. Bet iš linijos formos ir tolygiu linijų galima spręsti kad tai yra tinkamai balansuota. Antrame grafike yra pavaizduotas treniruotes tikslumo rezultatai juose galima matyti priešingai nuo treniruotes progreso didėjant tikslumui matosi skirtumas tarp validacijos ir treniravimo. Treniravimo tikslumas yra siek tiek didesnis nuo validacijos tai galima pamatyti kylant boostingui. Iš abiejų galima spręsti kad linijos yra gerai balansuotos nes nėra dideliu šuoliu nes skirtumo didėjimas tarp linijų yra tolygus. Taip pat modelio treniravimui panaudojau neuroninius tinklus.



Iš neuroniniu tinklu modeliu pasirinkau konvoliucini neuroniniu tinklu modeli. Taip pat modeliu keičiau įvairius hyperparametrus rezultatai buvo įvairus. Iš grafiko viršuje galima matyti kad yra skirtumas tarp validacijos ir treniravimo linijų abiejuose grafikuose. Treniravimo linija yra ilgesne už validacijos . Pirmame grafike galima matyti kad validacijos tikslumas prasideda nuo 0.8 tikslumo o treniravimo linijos tikslumas prasideda nuo apačios Nuo validacijos pradžios yra atstumas tarp linijų didėjant epochoms jis mažėja. Dideliu šuoliu nematyti. Antrame grafike validacijos linija prasideda esant mažiems praradimams ir vienu metu padidėja didėjant epochoms ir paskui mažėja keičiant hyperparametrus kartais gali rezultatai pagerėti o kartais tai gali pablogėti nes vieni parametrai pagerina rezultatą o kiti priešiningai pablogina ir naudoja daugiau kompiuterio resursu taip tai priklauso nuo tinkamai pasirinkto modelio užduočiai ir duomenų kiekiui. Palyginus šiuos modelius galime matyti rezultatu skirtumą. Bent jau šiuo atveju matosi kad didesniems duomenų rinkiniams geriau naudoti neuroninius tinklus. Nes galima pridėti įvarius hyperametrus ir kurie padeda rasti našesnį modelio treniravimą.



[Aivaras6889/BaigiamasisProjektas: Baigiamojo projekto užduotis su eismo ženklu duomenų rinkiniu.](https://github.com/Aivaras6889/BaigiamasisProjektas)

[Aivaras6889/BaigiamasisProjektas2](https://github.com/Aivaras6889/BaigiamasisProjektas2) -pagrindine repozitorija nes pirmoje neleido įkelti dėl per didelio failu didžio pirma palikau todėl kad parodyčiau kad dariau commits.