မာတိကာ

အမှာ

ယခုစာအုပ်သည် ချင်းမိုင် တက္ကသိုလ်တွင် လက်ရှိ သင်ကြားလျက် ရှိသော **‘Introduction to Supervised Machine Learning’** ဘာသာရပ်ကို အခြေခံ၍ ပြုစုထားသော စာအုပ်တစ်အုပ် ဖြစ်ပါသည်။

စာရေးသူသည် စင်ကာပူနိုင်ငံ နန်ယန်း တက္ကသိုလ်တွင် အင်ဂျင်နီယာ ဘွဲ့ကြိုသင်တန်းအား ၂၀၀၁ မှ ၂၀၀၅ ခုနှစ်ထိ တတ်ရောက်ခဲ့ပါသည်။ ထိုဘွဲ့ကြိုသင်တန်းကာလအတွင်း ပြုလုပ်ခဲ့သည့် **"Literature Review on Face Recognition Methods"** ဟုခေါ်ဆိုသည့် စာတမ်းငယ် မှာ စာရေးသူ၏ ပထမဦးဆုံး မှတ်ဥာဏ်အတု (**Artificial Intelligence or AI)** နှင့် ပတ်သက်သည့် သုတေသန လုပ်ငန်းစဉ် တစ်ခု ဖြစ်ခဲ့ပါသည်။ ထိုသုတေသနမှ အစပြု၍ **Computer Vision and Machine Learning** ကို ပိုမို စိတ်ဝင်စား လာကာ မာစတာနှင့် ဒေါက်တာ ဘွဲ့များကို ဆက်လက် ရယူခဲ့ပါသည်။ ၂၀၀၃ ခုနှစ် မှ ၂၀၁၃ ခုနှစ် ကာလ များတွင်လည်း သုတေသန စာတမ်းငယ်များစွာကို ထုတ်ဝေခဲ့ပြီး ဒေါက်တာ ဘွဲ့အတွက် တင်သွင်းခဲ့သည့် သုတေသန စာတမ်းကို ပြင်ဆင်၍ **"Contextual Analysis of Videos".** ခေါင်းစဉ်ဖြင့် စာအုပ်တအုပ်ကို လည်း ရေးသား ထုတ်ဝေခဲ့ပြီး ဖြစ်သည်။ အဆိုပါ စာအုပ်မှာ ယခုတိုင် ရောင်းချနေရဆဲ ဖြစ်ပြီး Amazon Website တွင် ဝယ်ယူဖတ်ရှုနိုင်ပါသည်။

အင်္ဂလိပ်ဘာသာဖြင့် နည်းပညာစာအုပ်များ စာစောင်များစွာ ရေးသား ထုတ်ဝေခဲ့သော်လည်း နည်းပညာ ဆိုင်ရာ စာအုပ်များကို မြန်မာ ဘာသာ အသုံးအနှုန်းဖြင့်ရေးသားရန် ယုံကြည်မှုအားနည်းခဲ့ပါသည်။ သို့သော် မြန်မာနိုင်ငံသို့ ပြန်ရောက်ပြီးနောက် လူငယ်များကို Computer Vision And Machine Learning ဆိုင်ရာ ဘာသာရပ်များကို သင်ကြားဖြစ်ရင်းမှ လိုအပ်ချက်များကို တွေ့မြင်လာရသည့်အတွက် ၂၀၁၇ /၂၀၁၈ ကာလတွင် "Introduction to MATLAB: Learning by Doing” ခေါင်းစဉ်ဖြင့် ပထမ ဦးဆုံး စာအုပ်ကို ထုတ်ဝေဖြစ်ခဲ့ပါသည်။ မြန်မာစာလုံးပေါင်း သတ်ပုံများ အမှား အယွင်းများစွာ ပါရှိခဲ့ သော်လည်း နည်းပညာစာအုပ်ဖြစ်သည့်အတွက် ဖတ်ရှုသူ များပြားခဲ့ပါသည်။ စာအုပ် စောင်ရေ ၅၀၀ နီးပါး ရောင်းချခဲ့ရပြီး စာဖတ်သူ

အများစုမှ အကျိုး ရှိသည်ဟု ပြန်လည် ပြောကြားမှုများ လက်ခံရရှိခဲ့ ရာမှ Introduction to Machine Learning စာအုပ်ကို ရေးသားရန် ခွန်အားများ ရရှိခဲ့ပါသည်။ ၂၀၁၉ ကာလတွင် စတင်ရေးသားခဲ့သော်လည်း အခြားလုပ်ငန်းများဖြင့် အချိန်ပေးနိုင်ရန် ခက်ခဲခဲ့ ရသောကြောင့် အဆုံးသတ် နိုင်ခဲ့ခြင်း မရှိပါ။

ယခုအခါ AI, Machine Learning ဆိုင်ရာ နည်းပညာ စာအုပ်များ၊ သင်တန်းများ အလွယ်တကူ ရရှိနိုင်ပြီ ဖြစ်သော်လည်း အင်တာနက် ခက်ခဲသော ဒေသမှ လူငယ်များ၊ အင်္ဂလိပ်စာ ဘာသာစကား အားနည်းသော လူငယ်များအတွက် လေ့လာရန် ခက်ခဲနေဆဲ ဖြစ်သည်ကို တွေ့မြင်နေရသည့်အတွက် အဆိုပါ လူငယ်များ အတွက် ရည်ရွယ်၍ 'Introduction to Supervised Machine Learning' (Supervised Machine Learning မိတ်ဆက်) စာအုပ်ကို အင်္ဂလိပ်-မြန်မာ နှစ်ဘာသာဖြင့် ရေးသားပြုစုထားပါသည်။

Supervised Machine Learning မိတ်ဆက် စာအုပ်တွင် အခန်းပေါင်း ၄ ခန်း ပါဝင်ပြီး ပထမ အခန်းတွင် Machine Learning နှင့် ပတ်သက် သည့် ယေဘုယျ သိသင့်သည်များကို မိတ်ဆက်ထားပါသည်။ ဒုတိယ အခန်းတွင် Regression Method နှင့် ပတ်သက်၍ ရှင်းလင်းထားပြီး တတိယ အခန်းတွင် Classification Method အချို့ကို ရှင်းလင်း တင်ပြထားပါသည်။ စတုတ္ထ ခန်းတွင်မူ ဆက်လက် လေ့လာသင့်သည့် သင်တန်းများ၊ စာအုပ်များကို အကြံပြုထားပါသည်။

မြန်မာစာပေ ဘာသာစကား အားနည်းချက်ကြောင့် စာလုံးပေါင်းသတ်ပုံ အမှားအယွင်းများ ရှိပါက သည်းခံ နားလည် ပေးပါရန် ကြိုတင် မေတ္တာရပ်ခံအပ်ပါသည်။

အခန်း (၁): နိဒါန်း

၁။ Machine Learning ဆိုသည့် စကားလုံးနှင့် ဘာသာရပ်သည် စာရေးသူတို့၏ နေ့စဉ် လူနေမှုဘဝအတွင်း စိမ့်ဝင်နေသည်မှာ နှစ်ပေါင်း ရာစုနှစ်တစ်ခုမက ကြာမြင့်ခဲ့ပြီ ဖြစ်သည်။ ယခု ခေတ်ကာလတွင် ဂူဂဲလ်ကို အသုံးပြု၍ လိုချင်သည့် အချက်အလက်များကို ရှာဖွေခြင်း ၊ မြေပုံထောက်၍ လိုရာခရီးကို သွားနှင်ခြင်းဆိုသည်မှာ လူငယ်များ အတွက် သာမက လူကြီးများအတွက်ပါ မဆန်းကြယ်တော့သည့် ဖြစ်ရိုးဖြစ်စဉ် လုပ်ငန်းများ ဖြစ်ပါသည်။ သို့သော် Machine Learning၊ Deep Learning ၊ Neural network ၊ AI စသည့် စကားလုံးများကို အလွယ်တကူ ဖလှယ်သုံးစွဲလျက် ရှိရာ စတင်လေ့လာသူများအတွက် နားလည်ရန် ခက်ခဲစေပါသည်။ ယခုအခန်းတွင် Machine Learning နှင့် သက်ဆိုင်သည့် စကားလုံးနှင့် အခေါ်အဝေါ်များကို ရင်းနှီး နားလည်စေရန် ဦးစွာ ပထမ ဆွေးနွေးသွားမည် ဖြစ်ပါသည်။

၂။ ဤစာအုပ်တွင် နည်းပညာဆိုင်ရာစကားလုံးများကို မြန်မာဘာသာသို့ ခက်ဆစ် အဓိပ္ပာယ် ပြန်ဆိုခြင်း ပြုလုပ်သွားမည် မဟုတ်ပဲ အင်္ဂလိပ်ဘာသာဖြင့်သာ ဖော်ပြသွားမည် ဖြစ်ပါ သည်။ သို့သော် နားလည်မှု လွယ်ကူစေရန် နေ့စဉ် လူနေမှု ဘဝတွင် တွေ့ကြုံရသည့် ဥပမာများကို အသုံးပြု၍ ရှင်းလင်းတင်ပြသွားမည် ဖြစ်သည်။

Machine Learning ၏ အဓိပ္ပာယ်

၃။ အင်္ဂလိပ်ဘာသာစကားတွင်ပင် Machine Learning ၏ အဓိပ္ပာယ်ကို ဖွင့်ဆိုချက် အမျိုးမျိုး ရှိပြီး အချို့ဖွင့်ဆိုချက်များမှာ နားလည်ရန် ခက်ခဲပြီး နည်းပညာသမားများသာ နားလည်နိုင်သည့် စကားလုံးများကို အသုံးပြုထားကြသည်။

၄။ Machine Learning ၏ အဓိပ္ပာယ်ကို လူအများစု အလွယ်တကူ နားလည်နိုင်ရန် အဓိပ္ပာယ် ဖွင့်ဆိုရမည်ဆိုပါက ကွန်ပြူတာ (သို့မဟုတ်) စက် တစ်ခုခုကို လူ့ကိုယ်စား (သို့မဟုတ်) လူကဲ့သို့ ပြုမူဆောင်ရွက်နိုင်စေရန် သင်ကြားပေးခြင်းဟု ယေဘုယျ ဖွင့်ဆို နိုင်ပါသည်။ ဥပမာ ပေးရမည် ဆိုပါက ၂၀၂၂ ခုနှစ် နိုဝင်ဘာလတွင် OpenAI မှ စတင် ထုတ်လိုက်သည့် ChatGPT [၁] သည် Machine Learning ကို အသုံးပြုထားသည့် အလိုအလျောက် ဖြေကြားပေးနိုင်သည့် ChatBot တစ်ခု ဖြစ်သည်။ အဆိုပါ ChatGPT သည် လူအများအပြား၏ စိတ်ဝင်တစား အသုံးပြုခြင်း ၊ ဖေဖန်ခြင်းများ အများဆုံး ရရှိလျက် ရှိသည်။ ChatGPT ကို ဒေတာ အများအပြား အသုံးပြု၍ လေ့ကျင့်သင်ကြားပေးထားရာ အသုံးပြုသူ မေးလိုသည့် မည်သည့် မေးခွန်းကိုမဆို သက်ရှိ လူတစ်ဦးမှ ဖြေကြားသကဲ့သို့ စကားပြောများဖြင့် ဖြေကြားပေးနေခြင်း ဖြစ်သည်။

၅။ Program များကို ChatGPT အား ရေးခိုင်း၍ ရသည့်အတွက် Software ရေးသူ Developer များအတွက် အနာဂတ် အလုပ်အကိုင် အခွင့်အလမ်းများ ခက်ခဲသွားနိုင်မလား ဟူသော စိုးရိမ်ပူပန်မှုများလည်း ပေါ်ထွက်လျက် ရှိသည်။ ထို့ပြင် ဆောင်းပါးရေးခြင်း၊ စာအုပ်၊ ကဗျာ ရေးခြင်းများကိုလည်း ခိုင်းစေနိုင်သည့်အတွက် ကိုယ်ပိုင်ဉာဏ်ဖြင့် ရေးသား ခြင်းများ ပျောက်ကွယ်သွားမလားဟု စိုးရိမ်ကာ အချို့တက္ကသိုလ်များတွင် ChatGPT အသုံးပြုခြင်းကို တားမြစ်ရန် ကြိုးစားခြင်းများပင် ရှိသည်။ AI ၊ Machine Learning ၏ တိုးတက်မှုများကို ကြားသိဖတ်ရှုရခြင်းသည် စာဖတ်သူများအတွက် အလှမ်းမမှီတော့သည့်

ဘာသာရပ်တစ်ခု အဖြစ် အားလျော့စေခြင်း ဖြစ်စေသည် ဆိုပါက အောက်ပါ ဥပမာလေး ကို စဉ်းစားစေချင်ပါသည်။

၆။ စာရေးသူတို့ ငယ်စဉ်ကလေးဘဝတွင် ပထမဦးဆုံး ပြောတတ်ရန် သင်ပေးခံရသည့် စကားလုံးမှာ 'မေမေ' ဟူသည့် စကားလုံးပင် ဖြစ်ပါလိမ့်မည်။ အဆိုပါ 'မေမေ' ဟူသည့် စကားကို စာရေးသူတို့ ရွတ်ဆိုနိုင်ရန် မိခင်ဖြစ်သူက သူမ၏ မျက်နှာ သို့မဟုတ် ခန္ဓာကိုယ် ကို ညွှန်ပြပြီး ပါးစပ်မှလည်း 'မေမေ' ဟု အကြိမ်ကြိမ် သင်ပေး တတ်ကြသည် မဟုတ်ပါ လား။ ထိုသင်ကြားခြင်းဖြစ်စဉ်မှာ Machine Learning ပင် ဖြစ်ပါသည်။ကွာခြားသွားသည့် အချက်မှာ သင်ပေးခံရသူ စာရေးသူတို့သည် လူသားများဖြစ်ပြီး ယခုအခါ Machine Learning ပညာရှင်တစ်ဦးအနေဖြင့် ကွန်ပြူတာကို သင်ကြားပေးမည့် Program ကို ရေးသား ရမည်ဖြစ်သည်။ ယခု ဥပမာလေးကို ဖတ်ရှုခြင်းဖြင့် Machine Learning ဆိုသည်မှာ စာရေးသူတို့ နေ့စဉ် လေ့လာသင်ယူခဲ့သည့် ဖြစ်ရပ်များကိုသာ အခြေခံထား ခြင်းဖြစ်ပြီး အလှမ်းမဝေးလှသည့် ဘာသာရပ်တစ်ခု ဖြစ်သည်ဟု ခံစားရမည်ဟု ယုံကြည် ပါသည်။

Machine Learning အမျိုးအစားများ

၇။ Machine Learning ကို ယေဘုယျအားဖြင့် (၃)မျိုး ခွဲခြားထားပါသည်။ ၎င်းအမျိုးအစားများမှာ အောက်ပါအတိုင်း ဖြစ်သည်။

(က) Supervised Machine Learning

(ခ) Unsupervised Machine Learning

(ဂ) Reinforcement Learning

၈။ ဤစာအုပ်တွင် Supervised Machine Learning နှင့် ပတ်သက်သည့် ဘာသာရပ်များကို အဓိက တင်ပြသွားမည်ဖြစ်ပြီး နည်းပညာဆိုင်ရာ စကားလုံးများကို မြန်မာဘာသာသို့ ပြန်ဆိုခြင်းမပြုပဲ စာဖတ်သူများ အလွယ်တကူ နားလည်နိုင်ရန် ဥပမာများဖြင့် ရှင်းပြသွားမည် ဖြစ်သည်။

Supervised Machine Learning

၉။ Supervised Machine Learning အမျိုးအစားဆိုသည်မှာ ကြိုတင် စုဆောင်းထား သည့် အချက်အလက်များကို အခြေခံ၍ ရလဒ်နှင့် အချက်အလက်များ အကြား ယေဘုယျ ဆက်သွယ်နိုင်သည့် ဆက်သွယ်ချက်တစ်ခုကို ရှာဖွေခြင်း ဖြစ်သည်။ ထို့နောက် အချက် အလက် အသစ်တစ်ခုရှိလာပါက အဆိုပါ ဆက်သွယ်ချက်ကို အသုံးပြု၍ ဖြစ်ပေါ်နိုင်မည့် ရလဒ်ကို ခန့်မှန်းခြင်း ဖြစ်သည်။

၁၀။ ဥပမာ 'မေမေ' ဟု ခေါ်ဝေါ်တတ်စေရန် သင်ကြားခြင်းသည် Supervised Machine Learning အမျိုးအစားတွင် အကျုံးဝင်သည်။ မိခင်ဖြစ်သူ၏ မျက်နှာသည် ကလေးငယ်ကို ပေးမည့် အချက်အလက်ဖြစ်ပြီး 'မေမေ' ဟူသော စကားလုံးသည် ကလေးငယ်ထံမှ ရရှိ လိုသည့် ရလဒ် ဖြစ်သည်။ သို့ဖြစ်ရာ မိခင်ဖြစ်သူ၏ မျက်နှာနှင့် 'မေမေ'ဟူသော စကားလုံး ကို ကလေးမှ တွဲ၍ မှတ်မိစေရန် အကြိမ်များစွာ လေ့ကျင့် သင်ကြားပေးရမည် ဖြစ်သည်။ လေ့ကျင့်သင်ကြားပေးသည့် အကြိမ်အရေအတွက် လုံလောက်မှသာ ကလေးငယ်သည် မိခင်၏ မျက်နှာအမျိုးမျိုးနှင့် 'မေမေ'ကို တွဲမိသွားပြီး မှန်ကန်သည့် ခေါ်ဝေါ်ခြင်းကို ပြုလုပ် နိုင်လာလိမ့်မည်။

၁၁။ Supervised Machine Learning အတွက် အခြားပေးလိုသည့် ဥပမာ တစ်ခုမှာ မျက်နှာ (သို့မဟုတ်) လက်ဗွေရာကို အသုံးပြု၍ ကွန်ပြူတာနှင့် မိုဘိုင်းဖုန်းများကို ဖွင့်ခြင်း ဖြစ်သည်။ စာရေးသူတို့အနေဖြင့် မျက်နှာ (သို့မဟုတ်) လက်ဗွေရာကို အသုံးပြု၍ ကွန်ပြူတာကို ဖွင့်လိုသည် ဆိုပါက ဦးစွာ စာရင်းပေးသွင်းရသည်ကို အများစု သတိပြု မိမည်ထင်ပါသည်။ ထိုသို့ စာရင်းသွင်းရာတွင် မျက်နှာကို အထက်သို့ မော့စေခြင်း၊ အောက်သို့ ငုတ်စေခြင်း၊ ဘေး ဘယ်ညာကို ကြည့်စေခြင်းများ ပြုလုပ်ခိုင်းခြင်းသည် မျက်နှာကို ရှုထောင့် အမျိုးမျိုး မှ ရိုက်ယူ၍ အချက်အလက်များ ရယူခြင်းပင်ဖြစ်သည်။ အဆိုပါ အချက်အလက်များကို စာရင်း သွင်းထားပြီးနောက် အသုံးပြုပါက ဆင်တူသည့်

အနေအထားတွင် ကွန်ပြူတာမှ လက်ခံ၍ ဖွင့်ပေးပြီး ကွဲပြားနေပါက ဖွင့်ပေးခြင်း မရှိပဲ ငြင်းဆိုမည် ဖြစ်သည်။

၁၂။ အထက်ပါ ဥပမာ ၂ ခုကို စဉ်းစားကြည့်ပါက Supervised Machine Learning ဟု ဆိုလျင် ကြိုတင် စုဆောင်းထားသည့် အချက်အလက်များ ရှိရန် လိုအပ်သည်။ အဆိုပါ အချက်အလက်များကို အခြေခံ၍ အချက်အလက်နှင့် ရလဒ်တွဲတတ်စေရန် ကွန်ပြူတာမှ သင်ကြားရမည်။ ထို့နောက် အချက်အလက် အသစ်ရရှိလာပါက ယခင် သင်ကြားခဲ့သည်များကို အခြေခံ၍ ရလဒ်ကို ခန့်မှန်းရမည် ဖြစ်သည်။ ခန့်မှန်းသည့် ရလဒ် အမျိုးအစားကို မူတည်၍ Supervised Machine Learning ကို အမျိုးအစား ၂ မျိုး ထပ်မံ ခွဲခြား နိုင်သည်။

(က) Regression Method များ၏ ရလဒ်မှာ ကိန်းဂဏန်း အရေအတွက် ဖြစ်သည်။ ဥပမာ အသက် ခန့်မှန်းခြင်း၊ ဈေးနှုန်း ခန့်မှန်းခြင်း တို့ဖြစ်သည်။

(ခ) Classification Method များ၏ ရလဒ်မှာ အမျိုးအစား ဖြစ်သည်။ ဥပမာ - မျက်နှာ သို့ လက်ဗွေရာကို အသုံးပြု၍ ကွန်ပြူတာနှင့် မိုဘိုင်းဖုန်းများကို ဖွင့်ရာတွင် ရလဒ်မှာ လက်ခံခြင်း (သို့မဟုတ်) ငြင်းပယ်ခြင်းဖြစ်သည်။ ထို့အတူ ကင်ဆာရောဂါ ရှိ/မရှိ ခန့်မှန်းခြင်း၊ စာမေးပွဲ အောင်/မအောင် ခန့်မှန်းခြင်းတို့မှာ Classification Method များ၏ ဥပမာများပင် ဖြစ်သည်။

၁၃။ Supervised Machine Learning Method များသည် ယနေ့ နည်းပညာလောကတွင် အသုံးအများဆုံး Machine Learning Method များဖြစ်ပြီး ပညာရေး၊ ကျန်းမာရေး၊ စီးပွားရေး နယ်ပယ် အသီးသီးတွင် အသုံးချလျက် ရှိသည်။

Unsupervised Machine Learning

၁၄။ Unsupervised Machine Learning အမျိုးအစားကိုမူ အသစ်ရရှိလာသည့် အချက်အလက်များကို အသုံးပြု၍ ရလဒ်ကို ခန့်မှန်းခြင်းထက် ရှိနေသည့် ဒေတာ အချက်အလက်များကို နားလည်အောင် ကြိုးစားရာတွင် အဓိက အသုံးပြုသည်။

၁၅။ ဥပမာ - စာရေးသူသည် ဆိုရှယ်မီဒီယာတွင် နေ့စဉ် ပုံတင်လေ့ ရှိသူ ဆိုကြပါစို့။ နှစ်ပေါင်း ၁၀စုနှစ် တစ်ခု အကြာတွင် စာရေးသူ တင်ခဲ့သည့် ဓါတ်ပုံအရေအတွက်မှာ သိန်းဂဏန်းခန့် ရှိသည်ကို တွေ့ရသည်။ စာရေးသူအနေဖြင့် မည်သည့်ပုံများကို တင်ခဲ့သည် (ဥပမာ - တစ်ဦးချင်း (သို့မဟုတ်) အသင်းအဖွဲ့ပုံများ) ၊ မည်သည့်နေရာများကို သွားခဲ့သည်ကို သိလိုပါက Unsupervised Machine Learning အမျိုးအစားတစ်ခု ဖြစ်သည့် Clustering Method ကို အသုံးပြု၍ အဖြေရှာနိုင်သည်။

၁၆။ အထက်ပါ ဥပမာတွင် Unsupervised Machine Learning Model ကို ပေးမည့် အချက်အလက်မှာ ဓါတ်ပုံ များစွာ ပါဝင်သည့် Computer Folder များ ဖြစ်ပြီး Clustering Method မှ ပြန်ပေးမည့် ရလဒ်မှာ တစ်ဦးချင်းပါဝင်သည့် ဓါတ်ပုံများ အုပ်စု နှင့် အသင်းအဖွဲ့ ပုံများပါဝင်သည့် ဓါတ် ပုံများ အုပ်စု ဟူ၍ အုပ်စု ၂ စု ကို ပြန်ပေးမည်ဖြစ်သည်။

၁၇။ Unsupervised Machine Learning Method တွင် ကြိုတင်လေ့လာထားသည့် အချက်အလက်များ မရှိသည့်အတွက် ရရှိလာသည့် အဖြေမှာ တိကျမှု အားနည်းမည် ဖြစ်သည်။ သို့သော် အုပ်စုတစ်ခုချင်းအတွင်း ရှိသည့် ဓါတ်ပုံများသည် အခြားအုပ်စုရှိ ဓါတ်ပုံများထက် တစ်ခုနှင့် တစ်ခု ပို၍ ဆင်တူမည် ဖြစ်သည်။

၁၈။ နှိုင်းယှဉ်ရမည် ဆိုပါက Supervised Machine Learning ကို အချက်အလက်များ ခန့်မှန်းရာတွင် အသုံးပြု၍ Unsupervised Machine Learning ကို လက်ဝယ်တွင် ရှိနေသည့် အချက်အလက်များမှ အဓိပ္ပာယ်ကို ရှာဖွေရာတွင် အသုံးပြုသည်။ အခြား

Unsupervised Machine Learning အမျိုးအစား တစ်ခုမှာ Dimensional Reduction Method ပင် ဖြစ်သည်။

၁၉။ အချုပ်ဆိုရသော် Unsupervised Machine Learning Method များသည် များပြား လှသည့် ဒေတာများ၏ အချက်အလက် အနှစ်ချုပ်ကို ရှာဖွေရန် အသုံးဝင်သည်။ ယနေ့ နည်းပညာလောကတွင် ဒေတာများသည် အရှိန်အဟုန်နှင့် တိုးပွားလျက် ရှိသည်။ ထိုသို့ များပြားလှသည့် ဒေတာများကို အလိုအလျောက် စုစည်းပေးခြင်း၊ အနှစ်ချုပ်ပေးခြင်း၊ အုပ်စုတူသည့် အချက်များကို စုစည်း ပေးခြင်း၊ ပုံမှန် မဟုတ်နေသည့် အချက်အလက် များကို ခွဲထုတ်ပေးခြင်းတို့သည် Unsupervised Machine Learning Method များ၏ စွမ်း ဆောင်ရည်များပင် ဖြစ်သည်။

Reinforcement Learning

၂၀။ Reinforcement Learning ဆိုသည်မှာ ပတ်ဝန်းကျင်မှ ပေးသည့် တုန့်ပြန်မှုကို မူတည်၍ ထပ်မံသင်ယူပြီး ပိုမို ကောင်းမွန်သော ရလဒ်များကို ခန့်မှန်းပေးခြင်း ဖြစ်သည်။

၂၁။ ဥပမာ - ChatGPT တွင် အသုံးပြုသူ မေးကြားလာသည့် မေးခွန်းများကို ဖြေကြား နိုင်ရန်အတွက် ကြိုတင်စုဆောင်းထားသည့် အချက်အလက်များကို အသုံးပြု၍ လေ့ကျင့် သင်ကြားထားခြင်းသည် Supervised Learning ဖြစ်သည်။ ထို့နောက် မေးခွန်းမေးသူ၏ တုန့်ပြန်ချက်များကို အသုံးပြု၍ ChatGPT ၏ မှတ်ဉာဏ်ကို ပိုမိုကောင်းမွန်လာစေရန် လေ့ကျင့်သင်ကြားပေးခြင်းသည် Reinforcement Learning ပင် ဖြစ်သည်။

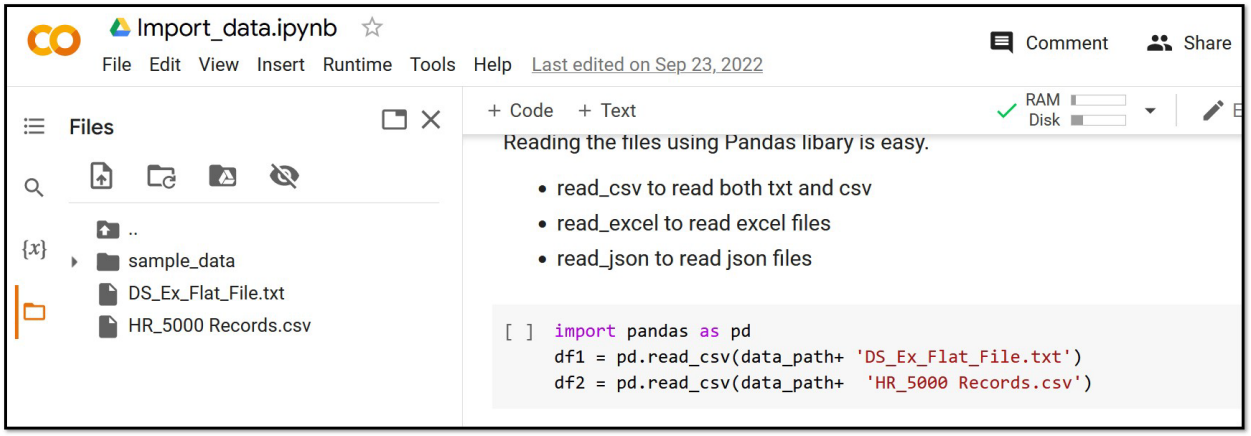
၂၂။ Reinforcement Learning ကို YouTube, Netflix ကဲ့သို့ Recommendation System များနှင့် Robotic Navigation System များတွင် အများစုတွေ့ရသည်။

Python မိတ်ဆက်

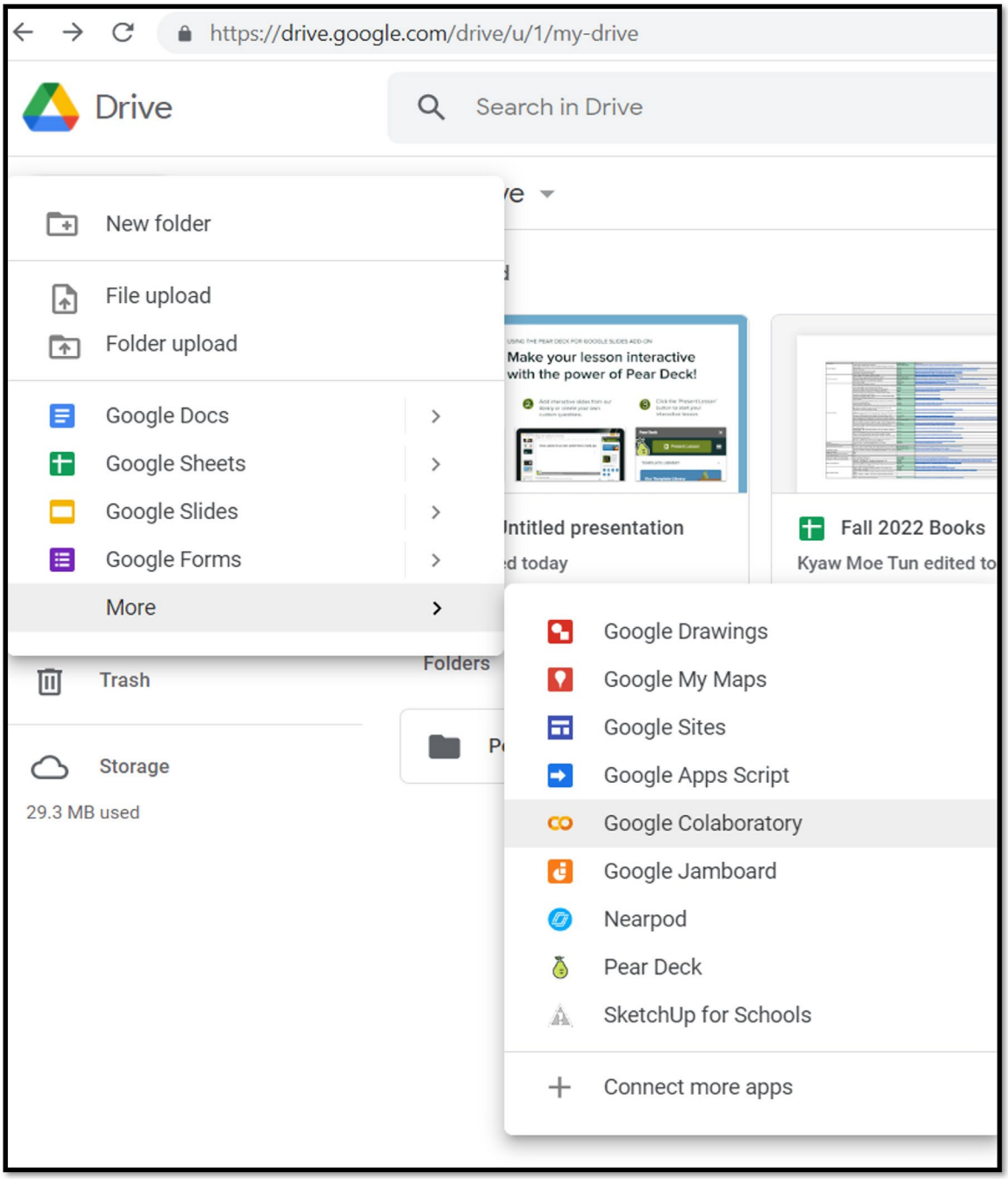
၂၃။ ယခု စာအုပ်တွင် ပါဝင်သည့် သင်ခန်းစာများကို Python Programming [၂] ကို အသုံးပြု၍ ရေးသွားမည် ဖြစ်သည်။ Python Programming ၏ အားသာချက်တစ်ခုမှာ လွယ်ကူခြင်းနှင့် အသင့် အသုံးပြုနိုင်သည့် library အမျိုးမျိုး ရှိခြင်းပင် ဖြစ်သည်။ Python Programming ကို အသုံးပြုနိုင်သည့် Platform အမျိုးမျိုးရှိရာ Jupyter Notebook [၃] သည် တစ်ခု အပါအဝင် ဖြစ်သည်။ ဤစာအုပ်တွင် ပါဝင်သည့် သင်ခန်းစာများကို Jupyter Notebook ဖြင့် ရေးသားထားခြင်းဖြစ်ပြီး ဒေါက်တာမျိုးသီတာ၏ GitHub link [၄] တွင် အခမဲ့ ရယူနိုင်ပါသည်။

၂၄။ ထို Jupyter Notebook ဖြင့် ရေးသားထားသည့် Program များကို Google Colab အသုံးပြု၍ အလွယ်တကူ စမ်းကြည့်နိုင်ပါသည်။ Google Colab သည် Google မှ ထုတ်လုပ်ထားသည့် Platform တစ်ခုဖြစ်ပြီး မိမိ၏ ကွန်ပြူတာထဲတွင် software ထည့်သွင်းရန် မလိုပဲ Google Server ကို အသုံးပြု၍ Program များ ရေးသားနိုင်သည်။

၂၅။ ပုံ - ၁(က)တွင် Google Colab ကို အသုံးပြု၍ Program ရေးသားထားခြင်းကို နမူနာ ပြသထားသည်။ Google Colab ကို Google Drive မှတဆင့် ဖွင့်သည့် နည်းလမ်းကို ပုံ ၁(ခ) တွင် ပြသထားပါသည်။



ပုံ ၁(က)။ နမူနာ Program



ပုံ ၁(ခ)။ Google Colab ကို Google Drive မှတဆင့် ဖွင့်သည့် နည်းလမ်း

၂၆။ Google Colab အသုံးပြုပုံ အသေးစိတ်ကို ဒေါက်တာမျိုးသီတာ၏ YouTube Link [၅] တွင် ဝင်ရောက် လေ့လာနိုင်သည်။

အခန်း (၂): ကိန်းဂဏန်း ခန့်မှန်းခြင်း (Regression Methods)

၁။ Regression Analysis ဆိုသည်မှာ ကြိုတင်စုဆောင်းထားသည့် အချက်အလက်များကို အခြေခံ၍ ရလဒ်နှင့် အချက်အလက်များ၏ ယေဘုယျ ဆက်သွယ်ချက်ကို ရှာဖွေခြင်း ဖြစ်သည်။ ထို့နောက် အဆိုပါ ယေဘုယျ ဆက်သွယ်ချက်ကို အသုံးပြု၍ အချက်အလက် အသစ် အတွက် ရလဒ်ကို ခန့်မှန်းခြင်း ဖြစ်သည်။ Regression Analysis ကို လေ့လာရာတွင် သိထားသင့်သည့် အသုံးအနှုန်း (၅) ခုနှင့် စတင် မိတ်ဆက်ပေးလိုပါသည်။

(က) **Training Data** ဆိုသည်မှာ ကြိုတင် စုဆောင်းထားသည့် အချက်အလက်များကို ဆိုလိုသည်။ Training Data တစ်ခုချင်းစီတွင် ရလဒ်နှင့် အဆိုပါ ရလဒ်ကို သက်ရောက်စေနိုင်သည့် အချက်အလက်များ ပါဝင်ရမည်။

(ခ) **Target Variable** ဆိုသည်မှာ ရလဒ်ကို ရည်ညွှန်းခြင်း ဖြစ်သည်။ ဥပမာ -- အိမ်ဈေးခန့်မှန်း သည့် ပုစ္ဆာတစ်ခုအတွက် အိမ်ဈေးသည် Target(ရလဒ်) ဖြစ်ပြီး အဆိုပါဈေးနှုန်းသည် တည်နေရာ၊ အကျယ်အဝန်း စသည့် အခြား အချက်အလက်များ အပေါ်တွင် မူတည်သည်။ သို့ဖြစ်ရာ Target Variable ကို Dependent Variable (မှီခို ကိန်းရှင်) အဖြစ်လည်း ဖလှယ် သုံးစွဲလေ့ ရှိသည်။

(ဂ) **Independent Variables** ဆိုသည်မှာ အထက်ပါ ရလဒ်ကို အကျိုး သက်ရောက်စေနိုင်သည့် အချက်အလက်များကို ရည်ညွှန်းသည်။ အထက်ပါ အိမ်ဈေး ခန့်မှန်းသည့် ပုစ္ဆာတွင် တည်နေရာ၊ အိမ်၏ အကျယ်အဝန်း၊ အိပ်ခန်း အရေအတွက် စသည့် အချက်အလက်များသည် Independent Variable များ ဖြစ်ကြသည်။ သတိပြုရန်မှာ ရလဒ် (ဥပမာ -အိမ်ဈေး) သည် ၁ ခု ထက်မကသည့် အချက်အလက်များ အပေါ်တွင် မူတည်သည်။

(ဃ) **Parameters** ဆိုသည်မှာ ရလဒ်နှင့် အချက်အလက်များ၏ ယေဘုယျ ဆက်သွယ်ချက်ကို ဖော်ပြနိုင်သည့် ကိန်းသေများကို ရည်ညွှန်းသည်။ ဥပမာ ရန်ကုန်တွင် တိုက်ခန်း ရောင်းချရာ၌ အိမ်အကျယ်အဝန်း ၁ ပေပတ်လည်ကို ၁ သိန်း/၁ သိန်းခွဲ စသည်ဖြင့် သတ်မှတ် ရောင်းချသည်ကို ကြုံဖူးကြပါလိမ့်မည်။ အဆိုပါ ဆက်သွယ်ချက်ကို သင်္ချာ ညီမျှခြင်းဖြင့် ဖော်ပြမည်ဆိုပါက

"တိုက်ခန်းဈေး = ကိန်းသေ x အိမ်အကျယ်အဝန်း" ဟုဖော်ပြနိုင်ပြီး အဆိုပါ ကိန်းသေကို Parameters ဟု ခေါ်ဆိုခြင်း ဖြစ်သည်။

(c) **Residuals** ဆိုသည်မှာ Machine Learning Method မှ ခန့်မှန်းလိုက်သော ရလဒ်နှင့် Training Data တွင် ပေးထားသည့် မူလရလဒ်၏ ခြားနားချက် ဖြစ်သည်။

Linear Regression

၂။ Linear Regression Analysis တွင် ရလဒ် (Target) နှင့် အချက်အလက် (Independent Variable) များသည် Linearly သို့မဟုတ် မျဉ်းဖြောင့် တစ်ကြောင်းထဲဖြင့် ဆက်သွယ် ထားသည်ဟူသော ယူဆချက်ကို အခြေခံ၍ Machine Learning Model တည်ဆောက်ခြင်း ဖြစ်သည်။

၃။ ဥပမာ - အိမ်ဈေး ခန့်မှန်းသည့် ပုစ္ဆာကို ပြန်လည် ညွှန်းဆိုရပါက အိမ် ၁ ပေ ပတ်လည်တက်သွားတိုင်း အိမ်ဈေးနှုန်းမှာ တက်သွားမည်။ တက်သွားသည့် နှုန်းမှာ တပြေးညီထဲ ဖြစ်နေမည်ဟု ယူဆ၍ “တိုက်ခန်းဈေး = ကိန်းသေ x အိမ်အကျယ်အဝန်း” ဟူသော ဆက်သွယ်ချက်ကိုအသုံးပြုပြီး ကိန်းသေ၏ တန်ဖိုးကို ရှာဖွေသည့် Method ကို Linear Regression Method ဟု ခေါ်ဆိုသည်။

၄။ Linear Regression Method တွင် ရလဒ်ကို ခန့်မှန်းသည့် အချက်အလက် အရေအတွက် အပေါ်မူတည်၍ Simple Linear Regression နှင့် Multiple Linear Regression Method ဟု အမျိုးအစား ၂ မျိုး ခွဲခြားထားသည်။

(က) ရလဒ်ကို ခန့်မှန်းရာတွင် အချက်အလက်တစ်ခုထဲကို အသုံးပြုပါက အဆိုပါ Method ကို Simple Linear Regression Method ဟု ခေါ်ဆိုပြီး

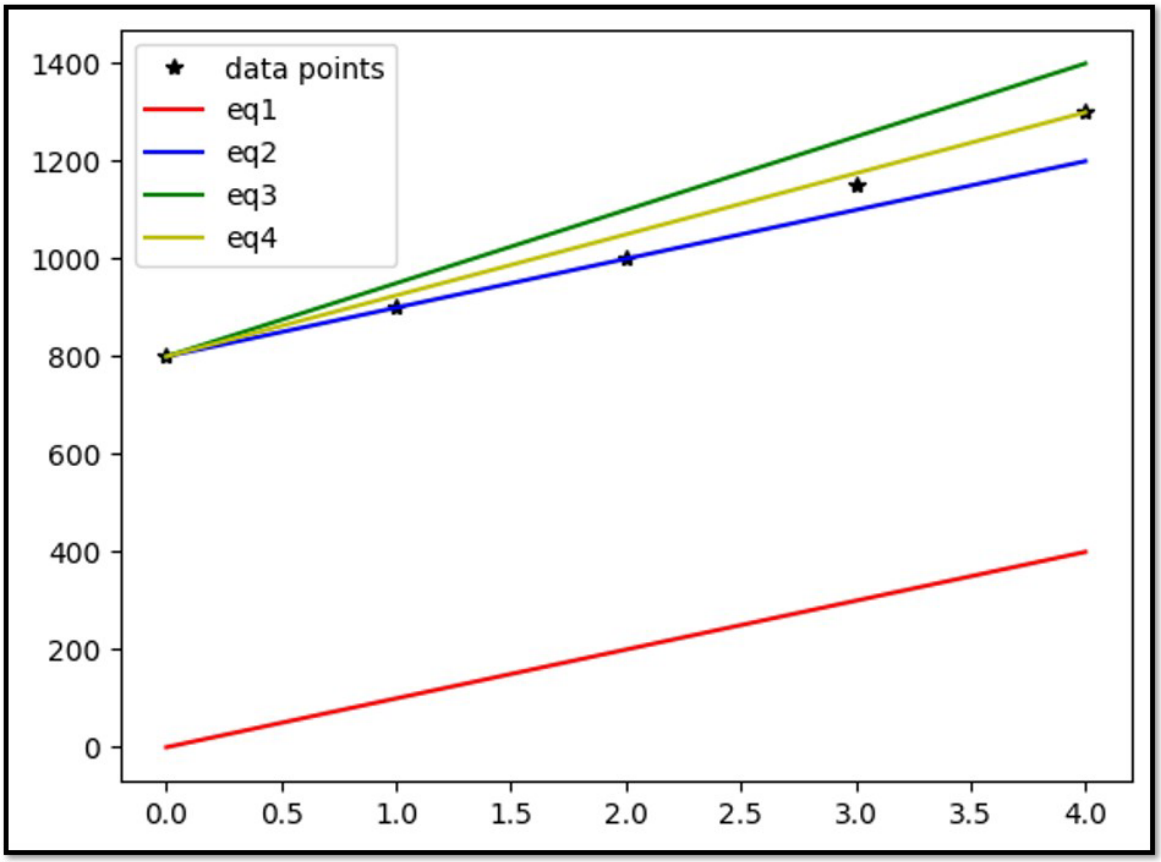
(ခ) တစ်ခုထက်ပိုသည့် အချက်အလက်များကိုအသုံးပြု၍ ရလဒ်ကို ခန့်မှန်းပါက Multiple Linear Regression Method ဟု ခေါ်ဆိုသည်။

Simple Linear Regression

၅။ Simple Linear Regression တွင် အချက်အလက်တစ်ခုထဲကို အသုံးပြု၍ ရလဒ်ကို ခန့်မှန်းရာ Target - y နှင့်Independent Variable - x ၏ ဆက်သွယ်ချက် ကို အောက်ပါ သင်္ချာညီမျှခြင်းဖြင့် ဖော်ပြနိုင်သည်။

$$y = \omega_1 x + b \qquad (၂.၁)$$

၆။ အထက်ပါ ညီမျှခြင်းကို ပုံဆွဲကြည့်မည်ဆိုပါက မျဉ်းဖြောင့်တကြောင်းကို ရရှိမည် ဖြစ်သည်။ Parameter များဖြစ်သည့် ω_1 နှင့် b ၏ နေရာတွင် ကိန်းဂဏန်းအမျိုးမျိုးကို အစားထိုးခြင်းအားဖြင့် ပုံ ၂(က) တွင် ပြထားသည့် မတူညီသည့် မျဉ်းဖြောင့်များကို ရရှိပါသည်။



ပုံ ၂ (က) Simple Linear Regression

၇။ ပုံ ၂(က) တွင် ဖော်ပြထားသည့် မျဉ်းဖြောင့် လေးကြောင်းကို နှိုင်းယှဉ်ကြည့်ပါက မျဉ်းဖြောင့်အားလုံးသည် Target - y နှင့်Independent Variable - x ၏ ယေဘုယျ ဆက်သွယ်ချက်ဖြစ်သည့် x ၏ တန်ဖိုး မြင့်တက်လာသည်နှင့် အမျှ y၏ တန်ဖိုးလည်း မြင့်တက်လာသည်ဟူသော ဆက်သွယ်ချက်ကို ဖော်ပြနိုင်သည်။ သို့သော် အနီရောင် မျဉ်းဖြောင့်ကို လေ့လာကြည့်မည်ဆိုပါက ပေးထားချက်ဖြစ်သည့် Training Data point (အမည်းရောင် ကြယ်ပွင့်လေးများဖြင့် ဖော်ပြထားသည်) တစ်ခုကိုမှ ဖြတ်သန်းသွားခြင်း မရှိသည်ကို တွေ့နိုင်ပါသည်။ တနည်းဆိုသော် အနီရောင် မျဉ်းဖြောင့်ကို ဖြစ်ပေါ်စေသည့် Machine learning Model မှ ခန့်မှန်းသည့် ရလဒ်သည် Training Data တွင် ပေးထားသည့် မူလရလဒ်နှင့် ကွာခြားနေမည်ဖြစ်သည်။

၈။ Simple Linear Regression ၏ ရည်ရွယ်ချက်မှာ Machine learning Model မှ **ခန့်မှန်းသည့် အဖြေနှင့်ရလဒ်အမှန်တို့ အကြား ကွာဟမှု** အနည်းဆုံး ဖြစ်စေသည့် Parameter (ကိန်းသေ) များကို ရှာဖွေရန် ဖြစ်သည်။

Multiple Linear Regression

၉။ Multiple Linear Regression တွင်မူ ရလဒ်ကို ခန့်မှန်းရာတွင် တစ်ခု ထက်မကသည့် အချက်အလက်များကို အသုံးပြု၍ ခန့်မှန်းသည်။ Target - y နှင့် Independent Variable များ၏ ဆက်သွယ်ချက်ကို အောက်ပါ သင်္ချာ ညီမျှခြင်းဖြင့် ဖော်ပြနိုင်သည်။

$$y = \omega_1 x_1 + \omega_2 x_2 + \cdots + b \qquad (၂.၂)$$

၁၀။ ဥပမာ Target - y သည် Independent Variable x_i အရေအတွက် ၃ ခု အပေါ်တွင် မူတည်သည်ဆိုပါက ညီမျှခြင်း ၂.၃ ကို အောက်ပါ အတိုင်း ရေးနိုင်သည်။ Independent Variable တစ်ခုစီအတွက် မြှောက်ဖော်ကိန်း တစ်ခုစီနှင့် cut-off point တစ်ခု - စုစုပေါင်း ကိန်းသေ ၄ ခုသည် Multiple Linear Regression Model ၏ Parameter (ကိန်းသေ) များ ဖြစ်သည်။

$$y = \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + b$$

၁၁။ Simple Linear Regression ကဲ့သို့ပင် Multiple Linear Regression ၏ ရည်ရွယ်ချက်မှာ **ခန့်မှန်းသည့် အဖြေနှင့် ရလဒ်အမှန်တို့ အကြား ကွာဟမှု** အနည်းဆုံး ဖြစ်စေသည့် Parameter (ကိန်းသေ) များကို ရှာဖွေရန် ဖြစ်သည်။

Cost Function

၁၂။ Cost Function ဆိုသည်မှာ Regression Model တစ်ခု၏ ခန့်မှန်းရလဒ် $(\widetilde{y_i})$ နှင့် Training Data တွင် ပေးထားသည့် ရလဒ်အမှန် (y_i) တို့ အကြား ပျမ်းမျှ ကွာဟမှုကို ခေါ်ဆိုခြင်းဖြစ်သည်။ Cost Function ကို အောက်ပါ သင်္ချာ ညီမျှခြင်းသုံး၍ ဖော်ပြနိုင်ပါသည်။

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_i - y_i) \qquad (၂.၃)$$

၁၃။ Simple Linear Regression အတွက် ခန့်မှန်းရလဒ် $(\widetilde{y_i})$ ကို ညီမျှခြင်း ၂.၁ ကို အသုံးပြု၍ တွက်ချက်နိုင်ပြီး Multiple Linear Regression အတွက် ခန့်မှန်းရလဒ် $(\widetilde{y_i})$ ကို ညီမျှခြင်း ၂.၂ သုံး၍ တွက်ချက်နိုင်သည်။

၁၄။ ညီမျှခြင်း ၂.၃ တွင် ဖော်ပြထားသည့် Cost Function ၏ အနည်းဆုံးတန်ဖိုး ဖြစ်စေမည့် Parameter များဖြစ်သည့် ω_i နှင့် b ကို ရှာဖွေရန်အတွက် Gradient Descent Method ကို အသုံးပြုလေ့ ရှိသည်။

Feature Scaling

၁၅။ Simple Linear Regression နှင့် Multiple Linear Regression တို့၏ အဓိက ကွာခြားချက်မှာ ရလဒ်ကို ခန့်မှန်းရာတွင် Independent Variable တစ်ခုထဲကို အသုံးပြုခြင်းနှင့် Independent Variable အများအပြားကို အသုံးပြုခြင်း ဖြစ်သည်။ တစ်ခုထက် ပိုသော Independent Variable များကို အသုံးပြုရာတွင် သတိပြုရမည့် အချက်မှာ Independent Variable တစ်ခု နှင့် တစ်ခုကြား ပမာဏ မတူညီခြင်း ဖြစ်သည်။

၁၆။ ဥပမာ တိုက်ခန်းတစ်ခန်း၏ ဈေးနှုန်းသည် အကျယ်အဝန်းနှင့် အိပ်ခန်း အရေအတွက်ပေါ် မူတည်သည်ဟူသောအချက်ကို အခြေခံ၍ ယေဘုယျ ဆက်သွယ်ချက် ရှာဖွေကြမည် ဆိုပါစို့။ တိုက်ခန်းတစ်ခန်း၏ အကျယ်အဝန်းမှာ ပေ ၅၀၀ မှ ပေ ၁၀၀၀ ကျော် ပတ်လည်ထိ ရှိနိုင်သော်လည်း အိပ်ခန်း အရေအတွက်မှာမူ ၁၀ ခန်းထက် မကျော်နိုင်ပါ။ မတူညီသည့် တန်ဖိုး ၂ ခုကို သင်္ချာ ညီမျှခြင်းတစ်ကြောင်းထဲကို အသုံးပြု၍ ဖြေရှင်းမည်ဆိုပါက တိကျ မှန်ကန်နိုင်ခြင်း မရှိပါ။

၁၇။ သို့ဖြစ်ရာ Multiple Linear Regression ကို အသုံးပြု မည်ဆိုပါက အချက်အလက် (Independent Variable) များကို တူညီသည့် တန်ဖိုး တစ်ခုထဲတွင် ရှိစေရန် ညှိပေးရမည် ဖြစ်သည်။ ထိုသို့ ပြုလုပ်ခြင်းကို Feature Scaling ဟု ခေါ်ဆိုပါသည်။ ဤစာအုပ်တွင် အသုံးများသည့် Feature Scaling နည်းလမ်း ၂ သွယ်ကို ဖော်ပြသွား ပါမည်။

Min-Max Scaling

၁၈။ Min-Max Scaling ဆိုသည်မှာ Independent Variable တစ်ခုချင်းစီ၏ တန်ဖိုးကို သုည နှင့် တစ် အကြား ရောက်အောင် ညှိယူခြင်း ဖြစ်သည်။ အထက်ပါ တိုက်ခန်း ဈေးနှုန်း ဥပမာတွင် တိုက်ခန်း၏ အနည်းဆုံး အကျယ်အဝန်းမှာ ပေ ၅၀၀ ပတ်လည်ဖြစ်ပြီး အကျယ်ဆုံးတိုက်ခန်းမှာ ပေ၁၅၀၀ ပတ်လည်ဟု ဆိုကြပါစို့။တိုက်ခန်းများ၏ အကျယ် အဝန်း ခြားနားချက်မှာ ပေ တစ်ထောင်ပတ်လည် (၁၅၀၀ - ၅၀၀ = ၁၀၀၀) ဖြစ်သည်ကို

တွေ့ရမည်။ ထို့နောက် Training Data-setရှိ တိုက်ခန်းတစ်ခန်းချင်းစီ၏ အကျယ်အဝန်းမှ အနည်းဆုံးတန်ဖိုး (၅၀၀) ကို နှုတ်၍ အကျယ်အဝန်းခြားနားချက် (၁၀၀၀) ဖြင့် စားပေးရမည်။

၁၉။ Min-Max Scaling ကို အသုံးပြု၍ Scaled ပြုလုပ်ထားသည့် တိုက်ခန်းများ၏ အကျယ်အဝန်းမှာ အကျဉ်းဆုံး တိုက်ခန်း အတွက် အနိမ့်ဆုံး တန်ဖိုး သုည (၀) နှင့် အကျယ်ဆုံး တိုက်ခန်း အတွက် အမြင့်ဆုံး တန်ဖိုး တစ် (၁) ရသည်ကို တွေ့ရ မည်။

၂၀။ Min-Max Scaling ကို သန့်ရှင်းမှုခြင်း အသုံးပြု၍ အောက်ပါ အတိုင်း ဖော်ပြနိုင်သည်။

$$x_{scale} = \frac{x - x\_min}{x\_\max - x\_min} \qquad (၂.၄)$$

၂၁။ Min-Max Scaling ကို Neural Network, Deep Learning Method များတွင် အများဆုံး အသုံးပြုသည်ကို တွေ့ရသည်။

Z-score Standardization

၂၂။ Z-score Standardization ဆိုသည်မှာ Independent Variable များ၏ တန်ဖိုးနှင့် ပျမ်းမျှ တန်ဖိုးအကြား ခြားနားချက်ကို မူတည်၍ Scaling ပြုလုပ်ခြင်း ဖြစ်သည်။ အထက်ပါ တိုက်ခန်းဈေးနှုန်း ခန့်မှန်းသည့် ဥပမာတွင် Training Data-setရှိ တိုက်ခန်းအားလုံး၏ ပျမ်းမျှ အကျယ်အဝန်းမှာ ပေ ၈၀၀ ပတ်လည်ဖြစ်ပြီး standard scalar တန်ဖိုးမှာ ပေ ၃၅၀ ပတ်လည် ဟု ယူဆကြပါစို့။ Z-score Standardization ကို အသုံးပြု၍ Scaling ပြုလုပ်ရန် Training Data-setရှိ တိုက်ခန်းတစ်ခန်းချင်းစီ ၏ အကျယ်အဝန်းမှ ပျမ်းမျှတန်ဖိုးကို နှုတ်၍ standard scalar ဖြင့် စားပေးရမည်။

၂၃။ Z-score Standardization ဖြင့် Scaled ပြုလုပ်ထားသည့် တိုက်ခန်းများ၏ အကျယ်အဝန်း တန်ဖိုးများ၏ ပျံ့နှံ့မှုမှာ Gaussian distribution အတိုင်း တည်ရှိနေမည်။ ပျမ်းမျှတန်ဖိုး သုည (၀) နှင့် standard scalar တန်ဖိုး တစ် (၁) ရသည်ကို တွေ့ရမည်။

၂၄။ Z-score Standardization ကို သန်းခေါင်စာရင်းဖြင့် အောက်ပါအတိုင်း ဖော်ပြနိုင်သည်။

$$x_{scale} = \frac{x-\mu}{\sigma} \qquad (၂.၅)$$

၂၅။ Z-score Standardization ကို Linear Regression, Logistic Regression စသည့် Method များတွင် အများဆုံး အသုံးပြုသည်ကို တွေ့ရသည်။

Assumptions

၂၆။ Linear Regression ကို အသုံးပြု၍ Machine Learning Model တည်ဆောက်ရာတွင် အထူးသတိပြုရမည့်အချက်မှာ Target- ရလဒ်နှင့် Independent Variable များအကြား ဆက်သွယ်ချက်မှာ Linear ဖြစ်နေရမည် ဖြစ်သည်။ တနည်းအားဖြင့် ရလဒ်နှင့် Independent Variable များ၏ ဆက်သွယ်ချက်ကို ပုံဖြင့် ဖော်ပြမည် ဆိုပါက မျဉ်းဖြောင့်တစ်ကြောင်းဖြင့် ဖော်ပြနိုင်ရမည် ဖြစ်သည်။

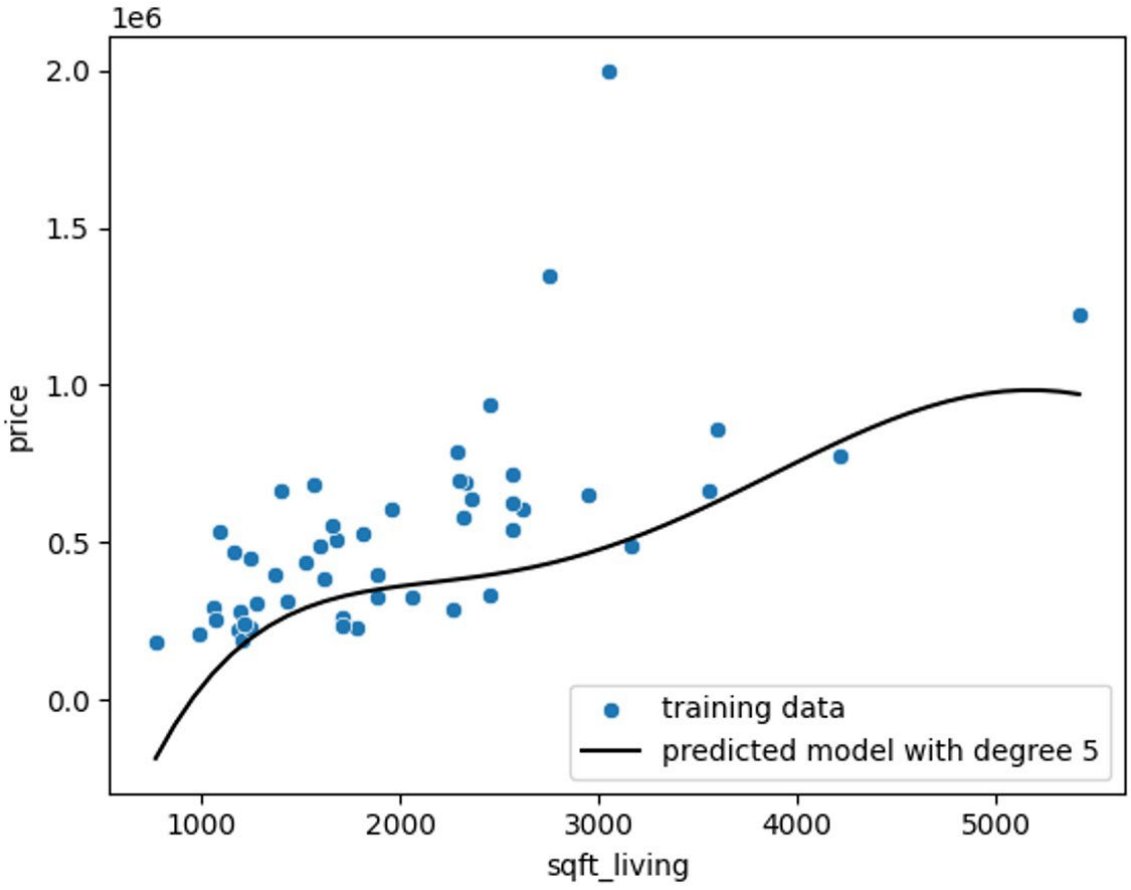
၂၇။ ထို့အပြင် Multiple Linear Regression တွင် ရလဒ်ကို ခန့်မှန်းရာ၌ အသုံးပြုသည့် အချက်အလက်များမှာ တစ်ခုနှင့် တစ်ခု မှီခိုခြင်း နည်းနိုင်သရွေ့နည်းရန် ဖြစ်သည်။ အကယ်၍ အထက်ပါ ယူဆချက်များနှင့် မကိုက်ညီပါက Linear Regression မှ ရလာသည့် ရလဒ်များမှာ တိကျမှု အားနည်းနေမည် ဖြစ်သည်။

၂၈။ သို့ရာတွင် Target- ရလဒ်နှင့် Independent Variable ဆက်သွယ်ချက်မှာ Linear ဖြစ်ရမည် ဟူသော ယူဆချက်မှာ အမြဲမှန်ကန်နိုင်ခြင်း မရှိပါ။ ဥပမာ - တိုက်ခန်းဈေးနှုန်းကို ပြန်လည်စဉ်းစားကြည့်ပါက တိုက်ခန်း အကျယ်အဝန်း ကြီးလာသည်နှင့် အမျှ ဈေးနှုန်း ပြောင်းလဲမှုနှုန်းသည် တပြေးညီ မဟုတ်တော့ပဲ လျော့ကျသွားခြင်းမျိုး ရှိနိုင်ပါသည်။ သို့ဖြစ်ရာ Target- ရလဒ်နှင့် Independent Variable ၏ ဆက်သွယ်ချက်မှာ Linear ဖြစ်သည် ဟူသော ယူဆချက် မမှန်ကန်နိုင်တော့ပါ။ ထိုကဲ့သို့ အခြေအနေမျိုးတွင် Linear Regression Model အစား Non-Linear Regression Model (သို့မဟုတ်) Polynomial Regression Model ကို အစားထိုး အသုံးပြုရမည် ဖြစ်သည်။

Polynomial Regression

၂၉။ Polynomial Regression Analysis ဆိုသည်မှာ ရလဒ်နှင့် အချက်အလက်များ၏ ဆက်သွယ်ချက် ကို မျဉ်းကွေးဖြင့် ဖော်ပြခြင်းဖြစ်သည်။ Polynomial Regression Model ကို သချိၤညီမျှခြင်းသုံး၍ အောက်ပါ အတိုင်း ဖော်ပြနိုင်သည်။

$$y = \omega a_1 x^n + \omega_2 x^{\{n-1\}} + \omega_3 x^{\{n-2\}} + \ldots + \omega_n x + b \qquad (၂.၆)$$



ပုံ ၂ (ခ) Illustration of Polynomial Regression

၃၀။ အထက်ပါ ညီမျှခြင်းကို ပုံဆွဲကြည့်မည်ဆိုပါက ပုံ ၂ (ခ) တွင် ဖော်ပြထားသည့် မျဉ်းကွေးကို ရရှိပါသည်။ ညီမျှခြင်း ၂.၆ ကို Polynomial Equation ဟု ခေါ်ပြီး

ညီမျှခြင်း၏ Degree (သို့မဟုတ်) Order ဖြစ်သည့် n ၏ တန်ဖိုးပေါ်တွင် မူတည်၍ မျဉ်းကွေး၏ ပုံအနေအထားမှာ ပြောင်းလဲသွားမည် ဖြစ်သည်။ n ၏ တန်ဖိုးကို တစ်(၁) ဟု သတ်မှတ်ပါက ညီမျှခြင်း (၂.၆) သည် Linear Regression Model ၏ ညီမျှခြင်းနှင့် ထပ်တူကျမည်ဖြစ်ပြီး Target ရလဒ်နှင့် Independent Variable အကြား ဆက်သွယ်မှုကို ပုံဆွဲပါက မျဉ်းဖြောင့်ကို ရရှိမည်။

၃၁။ Polynomial Equation ၏ Degree (n) တန်ဖိုး မြင့်တက်လာသည်နှင့် အမျှ ခန့်မှန်းရလဒ် (မျဉ်းကွေး)နှင့် Training Data-setအတွင်းရှိ Target ၏ မူလတန်ဖိုး (အပြာအစက်များ) အကြား ကွာဟမှု နည်းပါးလာမည် ဖြစ်သည်။ သို့သော် အခြားတဖက်ကမူ Polynomial Equation ၏ Degree (n) ကြီးလာသည်နှင့် အမျှ ရှာဖွေရမည့် ကိန်းသေ (Parameter) အရေအတွက်လည်း များပြားလာမည် ဖြစ်သည်။

Model Implementation

၃၂။ Regression Model ကို Python Library များ အသုံးပြု၍ အလွယ်တကူ တည်ဆောက်နိုင်ပါသည်။ သို့သော် Machine Learning Model တစ်ခု မတည်ဆောက်မီ ပထမဦးစွာ အရေးကြီးသည့်အဆင့်မှာ ဒေတာများကို အမှားအယွင်းများ မရှိရန် ပြင်ဆင် စစ်ဆေးခြင်း ဖြစ်ပါသည်။ မူလ အချက်အလက်များ မှားယွင်းနေပါက ရလာသည့် ရလဒ် သည်လည်း မည်သို့မျှ မတိကျနိုင်ပါ။ ထို့အပြင် ရလဒ်ကို ခန့်မှန်းရာတွင် အသုံးပြုမည့် အချက်အလက်များကို ရွေးချယ်ရာတွင်လည်း မှန်ကန်သည့် အချက်အလက်များ ဖြစ်ရန် လိုအပ်သည်။

၃၃။ ဥပမာ ကုမ္ပဏီတစ်ခုသည် ကုန်ပစ္စည်းများကို ရုပ်မြင်သံကြား၊ရေဒီယိုနှင့် သတင်းစာ များတွင် ကြော်ငြာလေ့သည် ဆိုကြပါစို့။ ရုပ်မြင်သံကြား၊ ရေဒီယိုနှင့် သတင်းစာ ကြော်ငြာများအနက် မည်သည့် အစီအစဉ်သည် ကုမ္ပဏီ၏ ကုန်ပစ္စည်း ရောင်းအားကို တိုးတက်မှု အများဆုံး ဖြစ်စေသည်ကို သိရှိရန်အတွက် Machine Learning Model အသုံးပြု၍ ဖြေရှင်းနိုင်သည်။

Data Collection

၃၄။ အထက်ပါ ပြဿနာကို ဖြေရှင်းနိုင်ရန်အတွက် ဦးစွာပထမ နေ့စဉ် (သို့မဟုတ်) အပတ်စဉ်အတွက် ကြော်ငြာတစ်မျိုးစီ၏ ကုန်ကျစရိတ် (Independent Variable) နှင့် ၎င်းကာလ အတွင်း ရောင်းရသည့် ရောင်းအား (target) များ ပါဝင်သည့် Data-setတစ်ခုကို ဦးစွာတည်ဆောက်ရမည် ဖြစ်သည်။ ထိုသို့ ကောက်ယူရာတွင်လည်း တစ်ပတ်စာ ၊ တစ်လစာ ကောက်ယူရုံဖြင့် မလုံလောက်ဘဲ အနည်းဆုံး တစ်နှစ်စာ ပါဝင်သည့် ဒေတာ အချက်အလက်များကို ကောက်ယူရန်လိုအပ်ပါသည်။

Data Pre-processing

၃၅။ ထိုသို့ ကောက်ယူရာတွင် အချက်အလက်များ မှားယွင်းစာရင်း သွင်းခြင်း မရှိရန် စီစစ်ရမည် ဖြစ်သည်။ ဥပမာ ကြော်ငြာတစ်ခု၏ အပတ်စဉ် ပျမ်းမျှတန်ဖိုးသည် တစ်သောင်း (သို့မဟုတ်) နှစ်သောင်းဖြစ်နေရာမှ တစ်သိန်း ဖြစ်နေပါက ထိုအချက်အလက် မှန်ကန်မှု ရှိ/မရှိ ပြန်လည် စီစစ်ရန် လိုအပ်သည်။ ထို့အပြင် အစီအစဉ် တစ်မျိုးနှင့် တစ်မျိုးအကြား ကုန်ကျသည့် ကြော်ငြာ စရိတ်၏ ပမာဏများ မညီမျှပါက အထက်တွင် ဆွေးနွေးခဲ့သည့်အတိုင်း Feature Scaling ပြုလုပ်ပေးရန် လိုအပ်သည်။

Train-Test Split

၃၆။ supervised Regression Method သည် ကြိုတင် စုဆောင်းထားသည့် အချက်အလက်များကို အသုံးပြု၍ ရလဒ်နှင့် အချက်အလက်ကြား ဆက်သွယ်မှုကို ဖော်ပြနိုင်မည့် ကိန်းသေ (Parameter) များကို ရှာဖွေခြင်း ဖြစ်သည်။ Machine Learning Model တည်ဆောက်ရာတွင် စုဆောင်းထားသည့် အချက်အလက်အားလုံးကို အသုံးပြုခဲ့မည် ဆိုပါက Model ၏ လုပ်ဆောင်ချက်ကို ပြန်လည် ဆန်းစစ်ရန် ခက်ခဲပါလိမ့်မည်။

၃၇။ သို့ဖြစ်ရာ ကြိုတင် စုဆောင်းထားသည့် အချက်အလက်များ အနက်မှ တချို့ တဝက်ကို Model တည်ဆောက်ရာတွင် အသုံးမပြုဘဲ ချန်လှပ်ထားရန် လိုအပ်ပါသည်။ ဥပမာ အထက်ပါ ကုမ္ပဏီမှ ရက် ၃၀၀ စာ ဒေတာ ကောက်ယူထားသည် ဆိုပါက ရက် ၂၀၀ စာ ဒေတာကို Machine Learning Model တည်ဆောက်ရာတွင် အသုံးပြု၍ ကျန် ရက် ၁၀၀ စာကို Model ၏ လုပ်ဆောင်ချက်ကို ပြန်လည် ဆန်းစစ်ရန်အတွက် အသုံးပြုနိုင်ပါသည်။ အဆိုပါ ရက် ၂၀၀ စာ ဒေတာကို Training Data Set ဟု ခေါ်ဆို၍ ကျန် ရက် ၁၀၀ စာ ဒေတာကို Testing Data Set ဟု ခေါ်ဝေါ် သုံးစွဲသည်။

၃။ အချက်အလက်များ အဆင်သင့်ဖြစ်ပြီးဆိုပါက Training Data Set ကို အသုံးပြု၍ Machine Learning Model စတင် တည်ဆောက်နိုင်မည် ဖြစ်ပါသည်။ ထို့နောက် Testing Data Set ကို အသုံးပြု၍ Machine Learning Model ၏ လုပ်ဆောင်ချက်ကို ပြန်လည် ဆန်းစစ်ရပါမည်။ သို့မှသာ Machine Learning Model ကို လက်တွေ့ အသုံးချရာတွင် ရရှိနိုင်မည့် ရလဒ်ကို ကြိုတင် ခန့်မှန်းနိုင်မည် ဖြစ်သည်။

Model Evaluation

၃၉။ Machine Learning Model ၏ လုပ်ဆောင်ချက်ကို ဆန်းစစ်ရာတွင် အောက်ပါ Metric များကို အသုံးပြု၍ တိုင်းတာလေ့ ရှိသည်။

(က) Mean Absolute Error (MAE) - - ခန့်မှန်း ရလဒ်နှင့် မူလ တန်ဖိုးကြား ခြားနားမှု၏ ပျမ်းမျှ ယူခြင်း ဖြစ်သည်။ လက္ခဏာများကို ထည့်သွင်း စဉ်းစားခြင်း မပြုရပါ။ Mean Absolute Error (MAE) ကို အောက်ပါအတိုင်း တွက်ချက်နိုင်သည်။

$$mae = \frac{1}{N}\sum_{i=1}^{N}|\widetilde{y_i} - y_i|$$

(ခ) Mean Squared Error (MSE) -- ခန့်မှန်း ရလဒ်နှင့် မူလ တန်ဖိုးကြား ခြားနားမှု၏ ၂ ထပ် ကိန်းကို ယူခြင်း ဖြစ်သည်။ Mean Squared Error MSE) ကို အောက်ပါအတိုင်း တွက်ချက် နိုင်သည်။

$$mse = \frac{1}{N}\sum_{i=1}^{N}(\widetilde{y_i} - y_i)^2$$

(ဂ) R2-score -- Machine Learning Model ၏ ခန့်မှန်းချက်သည် ပျမ်းမျှတန်ဖိုးနှင့် မည်မျှ ကွာခြားသည်ကို တွက်ချက်ခြင်း ဖြစ်ပြီး R2-score ၏ တန်ဖိုးကို အောက်ပါ အတိုင်း သတ်မှတ်ထားသည်။

$$R^2\ Score = 1 - \frac{RSS}{TSS}$$

RSS သည် ခန့်မှန်းတန်ဖိုးနှင့် ယေဘုယျတန်ဖိုး ခြားနားချက် ဖြစ်ပြီး TSS မှာ ခန့်မှန်းတန်ဖိုးနှင့် မူလတန်ဖိုး၏ ခြားနားချက်ဖြစ်သည်။ R2-score ကို ပိုမို နားလည်ရန် ဇယား (၂-က) နှင့် ဇယား (၂-ခ)တွင် ပြထားသည့် ဥပမာ ကို ကြည့်ပါ။

y_i	$\tilde{y}_i$	$(y_i - \tilde{y}_i)^2$	$(y_i - \bar{y}_i)^2$
10	10	0	100
20	20	0	0
30	30	0	100
$\bar{y}_i = 20$		RSS = 0	TSS = 200

ဇယား (၂-က) ခန့်မှန်း တန်ဖိုး = မူလတန်ဖိုး

ဇယား ၂ (က) တွင် မူလတန်ဖိုးနှင့် Machine Learning Model မှ ခန့်မှန်းသည့် ခန့်မှန်း တန်ဖိုးများတူနေသည်ကို တွေ့ရမည်။ သို့ဖြစ်ရာ RSS တန်ဖိုး သုည ဖြစ်ပြီး R2-score တန်ဖိုး တစ် (၁) ကို ရရှိပါမည်။

y_i	$\tilde{y}_i$	$(y_i - \tilde{y}_i)^2$	$(y_i - \bar{y}_i)^2$
10	20	100	100
20	20	0	0
30	20	100	100
$\bar{y}_i = 20$		RSS = 200	TSS = 200

ဇယား (၂-ခ) ခန့်မှန်း တန်ဖိုး = ပျမ်းမျှ တန်ဖိုး

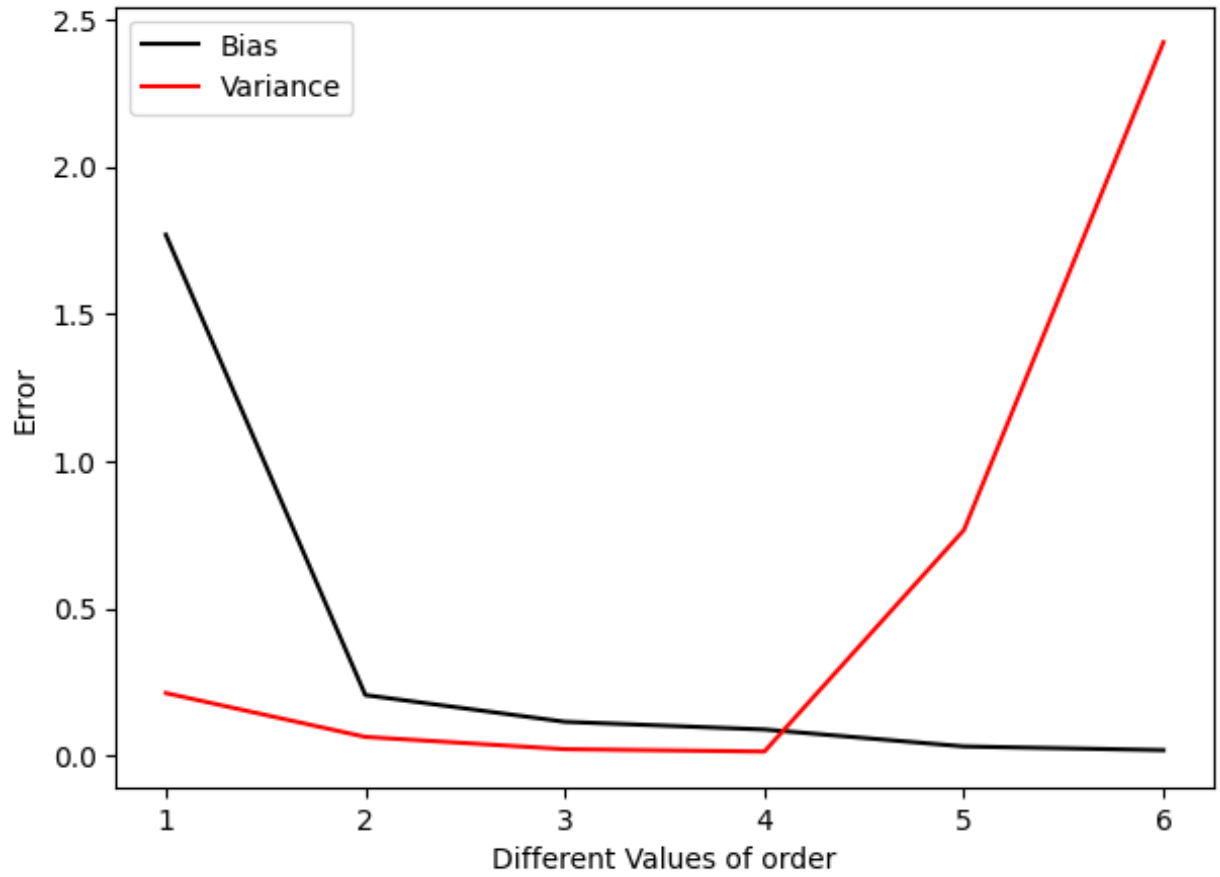
ဇယား ၂ (ခ) တွင် Machine Learning Model မှ ခန့်မှန်းသည့် ခန့်မှန်းတန်ဖိုးသည် Training Data ၏ ပျမ်းမျှတန်ဖိုးနှင့် တူညီနေသည်ကို တွေ့ရမည်။ သို့ဖြစ်ရာ RSS တန်ဖိုး နှင့် TSS တန်ဖိုး တူညီပြီး R2-score တန်ဖိုး သုည (၀) ကို ရရှိပါမည်။

၄၀။ အထက်ပါ ဥပမာ - ၂ ခုကို လေ့လာခြင်းအားဖြင့် Machine Learning Model ၏ ခန့်မှန်းချက်သည် မူလတန်ဖိုးနှင့် နီးစပ်ပါက R2-score ၏ တန်ဖိုး မြင့်တက်လာသည်ကို တွေ့ရသည်။ Model သည် မည်သည့် Independent Variable အတွက် မဆို Training Data ၏ ရလဒ် -Target ပျမ်းမျှတန်ဖိုးကိုသာ ပြန်ပေးသည် ဆိုပါက R2-score ၏ တန်ဖိုးသည် သုည ဖြစ်ပြီး ရံဖန်ရံခါတွင် အနှုတ်တန်ဖိုးလည်း ရရှိနိုင်သည်။

Bias-Variance Trade-off

၄၁။ Machine Learning Model ၏ လုပ်ဆောင်ချက်ကို ဆန်းစစ်ရာတွင် Training နှင့် Testing Data ၂ ခုလုံးကို အသုံးပြု၍ ဆန်းစစ်ရမည် ဖြစ်သည်။ ယေဘုယျအားဖြင့် mae / mse တန်ဖိုးကို လျှော့ချနိုင်ပြီး R2-score ၏ တန်ဖိုးမှာ တစ် (သို့မဟုတ်) ၁၀၀ ရာခိုင်နှုန်း နီးပါး ရရှိပါက Machine Learning Model ၏ လုပ်ဆောင် ချက်မှာ ကောင်းမွန်သည်ဟု ဆိုနိုင်မည် ဖြစ်သည်။ သို့သော် Training ဒေတာ အတွက် အကောင်းဆုံး ရလဒ် ရရှိရန် တည်ဆောက်ထားသည့် Model သည် Testing Data အတွက် ကောင်းမွန်မည်ဟု မဆို နိုင်ပါ။

၄၂။ ပုံ (၂- ဂ) တွင် Polynomial Regression တွင် Equation ၏ Degree မြင့်တက် လာသည်နှင့် အမျှ Machine Learning Model ၏ လုပ်ဆောင်ချက် မည်သို့ ပြောင်းလဲ သွားသည်ကို ဖော်ပြထားခြင်း ဖြစ်သည်။ Bias ဆိုသည်မှာ Model ၏ ခန့်မှန်း ရလဒ်နှင့် မူလတန်ဖိုး အကြား ခြားနားချက် ဖြစ်သည်။ Variance မှာ Training နှင့် Testing Data ကြား လုပ်ဆောင်ချက် ကွာဟမှုကို ရည်ညွှန်းခြင်း ဖြစ်သည်။



ပုံ(၂- ဂ) Bias-Variance Trade off

၄၃။ ပုံ(၂- ဂ) ကို သုံးသပ်ကြည့်ပါက Polynomial Regression Model ၏ Degree (complexity) မြင့်တက်လာသည်နှင့် အမျှ Bias တန်ဖိုး ကျလာသည်ကို တွေ့နိုင်ပါသည်။ သို့သော်တစ်ဖက်မှာမူ Training နှင့် Testing Data ကြား ကွာဟမှုမှာ မြင့်တက်လာသည်ကို တွေ့ရလိမ့်မည်။ Machine Learning Model တစ်ခုအနေဖြင့် Bias ကို လျော့ချနိုင်ရုံမက Training နှင့် Testing Data ကြား လုပ်ဆောင်ချက် ကွာဟမှု နည်းပါးစေရန်လည်း သတိပြုရမည် ဖြစ်သည်။

Regularization

၄၄။ Training Data အတွက် Model ၏ လုပ်ဆောင်ချက်မှာ အထူးကောင်းမွန်နေသော်လည်း Testing Data နှင့် အသစ်ဝင်လာသည့် ဒေတာများအတွက် လုပ်ဆောင်ချက် ကျဆင်းသွားခြင်းကို over-fitting problem ဟု ခေါ်ဝေါ်ပါသည်။ over-fitting ပြဿနာသည် Machine Learning Model များတွင် ကြုံတွေ့ရလေ့ ရှိပြီး အဆိုပါ ပြဿနာကို ဖြေရှင်းနိုင်ရန်အတွက် ညီမျှခြင်း ၂.၂ တွင် ဖော်ပြထားခဲ့ သည့် Cost Function တွင် Regularization term ကို ထည့်သွင်း အသုံးပြုလေ့ ရှိကြသည်။ Regularization term ပါဝင်သည့် Regression Model ၂ မျိုးမှာ အောက်ပါ အတိုင်း ဖြစ်သည်။

(က) Lasso Regression or L1 Regularization

(ခ) Ridge Regression or L2 Regularization

၄၅။ Lasso Regression တွင် over-fitting ကို လျှော့ချနိုင်ရန်အတွက် Independent Variable များ၏ မြောက်ဖော်ကိန်းဖြစ်သည့် ω_i ၏ ပကတိတန်ဖိုးကို ညီမျှခြင်း ၂.၂ တွင် ဖော်ပြထားသည့် cost function တွင် ထည့်ပေါင်း ခြင်း ဖြစ်သည်။

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_i - y_i) + \propto \sum_{i=1}^{N}|\omega_i|$$

၄၆။ Ridge Regression တွင်မူ ω_i ၏ ပကတိ တန်ဖိုးအစား ၂ထပ်ကိန်းကို ထည့်ပေါင်းပါသည်။ သို့ဖြစ်၍ Lasso Regression ကို L1 Regularization ဟု ခေါ်ဆိုပြီး Ridge Regression ကို L2 regularization ဟု ခေါ်ဆိုကြခြင်း ဖြစ်သည်။

$$\epsilon = \frac{1}{N}\sum_{i=1}^{N}(\tilde{y}_i - y_i) + \propto \sum_{i=1}^{N}\omega_i^2$$

၄၇။ Regularization term တွင်ပါဝင်သည့် $\propto$ ကို အချို့ စာအုပ်များတွင် C သင်္ကေတဖြင့်လည်း ဖော်ပြလေ့ရှိပြီး $\propto$ တန်ဖိုးသည် Regularization ၏ ပမာဏကို ထိန်းညှိပေးပါသည်။ $\propto$ တန်ဖိုးများပါက မူလ Parameter(ကိန်းသေ)များ၏ တန်ဖိုးသည် လျော့ကျလာပြီး ရလဒ်ကို ဆုံးဖြတ်ရာတွင် သက်ရောက် မှု နည်းစေသဖြင့် over-fitting ကို လျှော့ချနိုင်စေပါသည်။ Lasso Regression တွင်မူ $\propto$ တန်ဖိုးကြီးပါက ရလဒ်ကို ဆုံးဖြတ်ရာတွင် အချို့ Independent Variable များ၏ မြှောက်ဖော်ကိန်းတန်ဖိုးကို သုည အထိ လျှော့ချပေးရာ Independent Variable အရေအတွက်များနေသည့် ပုစ္ဆာများကို ဖြေရှင်းရာတွင် ပိုမို အသုံးဝင်သည်ကို တွေ့ရသည်။

၄၈။ အောက်ပါ Linear Equation သည် Multiple Linear Regression Model တစ်ခု၏ Equation ဖြစ်သည်။ y သည် Model ၏ ရလဒ်ကို ရည်ညွှန်းပြီး $x_1, x_2,$ နှင့် x_3 တို့မှာ Independent Variable များ ဖြစ်ကြသည်။ မြှောက်ဖော်ကိန်းတန်ဖိုးများကို နှိုင်းယှဉ်ကြည့်မည်ဆိုပါကရလဒ်ကို ဆုံးဖြတ်ရာတွင် အဓိက အရေးအပါဆုံး Independent Variableမှာ x_1 ဖြစ်ပြီး x_3၏ မြှောက်ဖော်ကိန်းတန်ဖိုးမှာ အနည်း ဆုံးဖြစ်သည်ကို တွေ့နိုင်ပါသည်။

$$y = 3.893\,x_1 + 3.420\,x_2 + 2.943\,x_3 + 13.788$$

၄၉။ အထက်ပါ Multiple Linear Regression Model တွင် L1 or L2 Regularization term များ ထည့်သွင်းခြင်း၏ အကျိုးကျေးဇူးကို အောက်ပါ ဇယားတွင် ဖော်ပြထားပါသည်။ $\propto$ တန်ဖိုး များလာ သည်နှင့် အမျှ Lasso Regression ရှိ မြှောက်ဖော်ကိန်းတန်ဖိုးများမှာ တဖြည်းဖြည်း နည်းလာပြီး $\propto = 0.5$ ဖြစ်သည့်အချိန်တွင် x_3၏ မြှောက်ဖော်ကိန်း တန်ဖိုးမှာ သုည ဖြစ်သွားသည်ကို တွေ့ရမည်။ Ridge Regression တွင်မူ x_3၏ မြှောက်ဖော်ကိန်းတန်ဖိုးမှာ မြင့်တက်လာပြီး x_1, x_2 တို့၏ မြှောက်ဖော်ကိန်း တန်ဖိုးများမှာ နည်းလာသည်ကို တွေ့နိုင်သည်။

	$\alpha = 0$	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 10$
ω_1 (L)	3.893	3.420	2.943	0
ω_2 (L)	2.761	2.310	1.832	0
ω_3 (L)	0.067	0	0	0
ω_1 (R)	3.893	3.879	3.865	3.625
ω_2 (R)	2.761	2.749	2.738	2.546
ω_3 (R)	0.067	0.072	0.078	0.163

ဇယား (၃) ၊ Lasso Regression နှင့် Ridge Regression ကို နှိုင်းယှဉ်ခြင်း

၅၀။ အချုပ်ဆိုရသော် over-fitting ပြဿနာကို ကျော်လွှားနိုင်ရေးအတွက် Regularization term ထည့်သွင်းခြင်းသည် ကောင်းမွန်သည့် အလေ့အထ တစ်ခုဖြစ်သည်။ သို့သော် Regularization ၏ ပမာဏ ကို ထိန်းချုပ်ပေးမည့် hyper-Parameter တစ်ခုဖြစ်သည့် ∝ တန်ဖိုးကို သတိထား ရွေးချယ်ပေးရန် လိုအပ် သည်။

Project: Sale Amount Prediction

၅၁။ အောက်ပါ Program သည် ကုမ္ပဏီ တစ်ခု၏ ရောင်းအားကို ခန့်မှန်းနိုင်ရန်အတွက် Multiple Linear Regression Model တစ်ခုကို တည်ဆောက်သည့် အဆင့်ဆင့် ဖြစ်သည်။ ယခု Program တွင် အသုံးပြုထားသည့် Data Set ကို kaggle.com [၇] မှ ရယူသည်။ အဆိုပါ Data Set တွင် ကြော်ငြာ အစီအစဉ်တစ်မျိုးစီ၏ ကုန်ကျစရိတ် (Independent Variable) နှင့် ၎င်းကာလ အတွင်း ရောင်းရသည့် ကုမ္ပဏီ၏ ရောင်းအားများ ပါဝင်သည်။ ကြော်ငြာ အစီအစဉ်များမှာ ရုပ်မြင်သံကြား၊ရေဒီယိုနှင့် သတင်းစာတို့ဖြစ်ပြီး Column တစ်ခုချင်းစီတွင် သက်ဆိုင်ရာ အစီအစဉ်၏ ကုန်ကျစရိတ်များ ပါဝင်သည်။ နောက်ဆုံး Column မှာ ကုမ္ပဏီ၏ ရောင်းအား ပမာဏ ဖြစ်သည်။ ယခု ပုစ္ဆာတွင် ကုမ္ပဏီ၏ ရောင်းအားမှာ Target ဖြစ်ပြီး Independent Variable – ၃ ခု ပါဝင်သည်။ Target နှင့် Independent Variable များသည် Linearly ဆက်သွယ်နေသည်ဟု ယူဆ၍ Multiple Linear Regression Model ကို တည်ဆောက်မည် ဖြစ်သည်။

အဆင့် ၁။ ။ ပထမဆုံး အဆင့်အနေဖြင့် လိုအပ်သည့် Python Library များကို import လုပ်ပြီး Data-setမှ target နှင့် Independent Variable များကို ခွဲထုတ်ရန် ဖြစ်သည်။

```
# ================================================#
import pandas as pd

df=pd.read_csv('..\\data\Advertising.csv')

X=df[['TV', 'radio', 'newspaper']].values
y=df['sales'].values

# ================================================#
```

အဆင့် ၂။ ။ ဒုတိယ အဆင့်အနေဖြင့် Training Data-set နှင့် Testing Data-set ခွဲရန် ဖြစ်သည်။ မူလ Data-set ၏ ၃ ပုံ-၂ ပုံကို Training အတွက် အသုံးပြုပြီး ကျန် တစ်ပုံကို Testing အတွက် အသုံးပြုမည် ဖြစ်သည်။

```
# ================================================#
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y,
                                    test_size = 0.33,
                                    random_state=1)
# ================================================#
```

အဆင့် ၃။ ။ လိုအပ်သည့် Hyper-Parameter ဖြစ်သည့် Degree တန်ဖိုး ရှာဖွေခြင်း နှင့် Model implementation အဆင့် ဖြစ်သည်။

```
# ================================================#
from sklearn.model_selection import GridSearchCV
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
from sklearn.pipeline import Pipeline

steps = [('scaler', StandardScaler()),
         ('poly', PolynomialFeatures(degree = 2,
                                     include_bias=False)),
         ('liReg', LinearRegression())]
parameters = {"poly__degree":[2, 3, 4, 5, 7, 9]}
pipeline = Pipeline(steps)
poly_grid = GridSearchCV(pipeline, parameters,
                         cv=5,
                        scoring='neg_mean_squared_erro
                        verbose= True)
poly_grid.fit(X_train, y_train)
print ('best order is :', poly_grid.best_params_)
# ================================================#
```

အဆင့် ၄။ ။ Model ၏ လုပ်ဆောင်ချက်ကို ဆန်းစစ်သည့် အဆင့်ဖြစ်သည်။

```
# ================================================#
rom sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score

# Evaluation on the Tesing data set
ytest_pred = poly_grid.predict(X_test)
mae = mean_absolute_error(y_test, ytest_pred)
mse = mean_squared_error(y_test, ytest_pred,
                         squared= True)
r2 = r2_score(y_test, ytest_pred)
#Evaluation on the Training data set
ytr_pred = poly_grid.predict(X_train)
maeT = mean_absolute_error(y_train, ytr_pred)
mseT = mean_squared_error(y_train, ytr_pred,
                         squared= True)
r2T = r2_score(y_train, ytr_pred)
#Keep all results in the tabular form
result = pd.DataFrame({'mae': [maeT, mae],
                         'mse': [mseT, mse],
                         'r2': [r2T, r2]})
result.index = ['Training', 'Testing']
# ================================================#
```

၅၂။ ကြော်ငြာ အစီအစဉ် ၃ ခု အနက် ရုပ်မြင်သံကြားမှ ကြော်ငြာခြင်းသည် အထိရောက်ဆုံး ဖြစ်သည်ကို တွေ့ရသည်။ ရုပ်မြင်သံကြား အစီအစဉ်အတွက် ၁ ဒေါ်လာ တိုး၍ အသုံးပြုတိုင်း ရောင်းအားပမာဏ ၄ ဒေါ်လာ နီးပါး တိုးလာသည်ကို တွေ့ရသည်။ Model ၏ R2-Score သည် Training Data နှင့် Testing Data ၂ ခု လုံးအတွက် ၉၉ ရာခိုင်နှုန်း ရရှိသည်။

အခန်း (၃): အမျိုးအစား ခွဲခြားခြင်း (Classification Methods)

၁။ Classification Method များ၏ အဓိက လုပ်ဆောင်ချက်မှာ ပေးလာသည့် အချက်အလက်များကို အမျိုးအစား ခွဲခြားပေးခြင်း ဖြစ်သည်။ ဥပမာ -- Face-authentication စနစ်တွင် ကင်မရာမှ ဖတ်လိုက်သည့် မျက်နှာကို Accept (လက်ခံသည်) သို့မဟုတ် Reject (ငြင်းပယ်သည်) ဟု ရလဒ် ထုတ်ပေးရသည်။ ရလဒ်မှာ လက်ခံသည် နှင့် ငြင်းပယ် သည် ဟူသည့် အဖြေ နှစ်မျိုးထဲမှ တစ်မျိုးသာ ဖြစ်သည့် အတွက် ထိုကဲ့သို့ Classification Method ကို Binary Classification Method ဟုလည်း ခေါ်ဝေါ်သုံးစွဲကြသည်။

၂။ Binary Classification Method ကို အသုံးပြုသည့် နမူနာ အချို့မှာ အောက်ပါ အတိုင်း ဖြစ်သည်။

(က) Spam Email ဟုတ်/မဟုတ် စီစစ်ပေးခြင်း။

(ခ) အကျိတ်သည် ကင်ဆာ ဟုတ်/မဟုတ် စီစစ်ပေးခြင်း။

(ဂ) ငွေချေးထားသည့် client တစ်ဦးသည် အကြွေးပြန်ဆက်ခြင်း/ မဆက်ခြင်း။

၃။ Multi-class Classification Method တွင်မူ Input ပေးလိုက်သည့် Independent Variable အတွက် ခန့်မှန်းရမည့် ရလဒ် အမျိုးအစားမှာ ၂ ခု အနက်မှ တစ်ခု မဟုတ်တော့ပဲ နှစ်ခုထက်ပိုသော ရလဒ်များအနက်မှ တစ်ခုကို ခန့်မှန်းပေးခြင်း ဖြစ်သည်။ ဥပမာ - နံမည် တစ်ခုကို တိုင်းရင်းသား အမျိုးအစား ခွဲခြားပေးခြင်း၊ ဓါတ်ပုံ တစ်ပုံကို ခွေး၊ကြောင်၊ ယုန် စသည်ဖြင့် အမျိုးအစား ခွဲခြားပေးခြင်း ဖြစ်သည်။

၄။ ထို့အပြင် Input ပေးလိုက်သည့် Independent Variable တစ်ခုအတွက် အမျိုးအစားတစ်ခုထက်မကသည့် အခြေအနေမျိုးလည်း ရှိနေနိုင်သေးသည်။ ဥပမာ အစားအသောက်ကို ရိုက်ထားသည့် ဓါတ်ပုံ တစ်ပုံကို အမျိုးအစားခွဲခြားရာတွင်

အစားအသောက် အမျိုးအစား တစ်ခုထက်မကသော အဖြေကို ပေးနိုင်သည်။ ထိုကဲ့သို့သော Classification Method ကိုမူ Multi-label Classification Method ဟု ခေါ်ဆိုသည်။

၅။ Regression Method (အခန်း ၂) တွင် ဆွေးနွေးခဲ့သော Training data, Target, Independent variable, Residual, Parameter စသည့် အသုံးအနှုန်းများသည် Classification Method များတွင်လည်း အကျုံးဝင်သည်။

Performance Evaluation Metric for Classification

၆။ Classification Method များ၏ လုပ်ဆောင်ချက်ကို ဆန်းစစ်ရာတွင် အသုံးပြုသည့် Evaluation Metric အများအပြားရှိပြီး Classification ပုစ္ဆာ အပေါ်မူတည်၍ သုံးစွဲရသည် ဖြစ်ရာ မည်သည့် Classification Method ကိုမှ မဆွေးနွေးမီ ယခု အခန်းတွင် အသုံးများသည့် Evaluation Metric များကို ဆွေးနွေးတင်ပြသွားမည် ဖြစ်သည်။

၇။ အောက်ပါ အခန်းငယ်များတွင် ဆွေးနွေးသွားမည့် Evaluation Metric များသည် Binary Classification ပြဿနာကို အခြေခံ၍ တင်ပြသွားမည် ဖြစ်သည်။ သို့သော် အဆိုပါ ဆွေးနွေးချက်များအားလုံးကို Binary Classification ပြဿနာများအတွက်လည်း အသုံးပြုနိုင်သည်။

Confusion Matrix

၈။ Binary Classification ပြဿနာများတွင် အဖြေမှာ Positive (ဟုတ်သည်/ ရှိသည်) သို့မဟုတ် Negative (မဟုတ်/ မရှိ) - ၂မျိုး ထဲမှ တစ်မျိုးသာ ဖြစ်သည်။ Program ရေးရာတွင် Positive Class ကို တစ် (၁) နှင့် ဖော်ပြ၍ Negative Class ကို သုည (၀) နှင့် ဖော်ပြလေ့ ရှိသည်။

၉။ Binary Classification Model ၏ ခန့်မှန်းချက်နှင့် ပတ်သက်၍ ဖြစ်ပေါ်လာနိုင်သည့် အခြေအနေ ၄ မျိုး ရှိသည်။

(က) Positive Class ကို Model မှ **Positive** ဟု ခန့်မှန်းခြင်း (True **Positive**)

(ခ) Positive Class ကို Model မှ **Negative** ဟု ခန့်မှန်းခြင်း (False **Negative**)

(ဂ) Negative Class ကို Model မှ **Positive** ဟု ခန့်မှန်းခြင်း (False **Positive**)

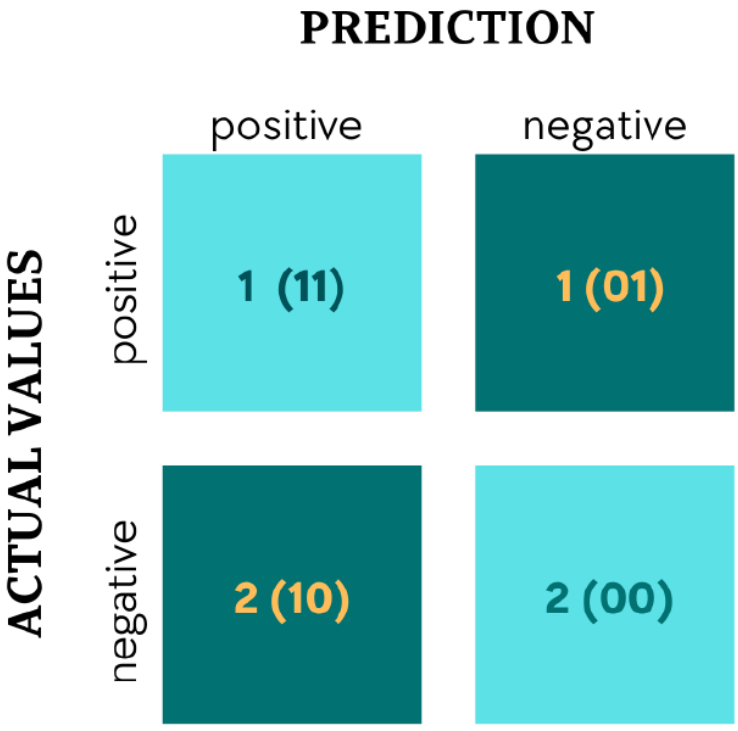
(ဃ) Negative Class ကို Model မှ **Negative** ဟု ခန့်မှန်းခြင်း (True **Negative**)

၁၀။ အထက်ပါ အခြေအနေ လေးမျိုးအနက် True **Positive** နှင့် True **Negative** မှာ မှန်ကန်သည့် ခန့်မှန်းချက်များ ဖြစ်ပြီး False **Negative** နှင့် False **Positive** မှားယွင်းသည့် ခန့်မှန်းချက်များ ဖြစ်သည်။

၁၁။ ဇယား (၃-က) တွင် Binary Classification Model တစ်ခု၏ ခန့်မှန်း ရလဒ်နှင့် မူလ တန်ဖိုးကို ယှဉ်၍ ပြသထားသည်။ အဆိုပါ ဇယား (၃-က) အရ True **Positive - တစ်ခု** နှင့် True **Negative - နှစ်ခု** ရှိပြီး False Negative - တစ်ခု နှင့် False **Positive** နှစ်ခု ရှိသည်ကို တွေ့နိုင်သည်။ အထက်ပါ ရလဒ်ကို Squared Matrix အတွင်း ထည့်လိုက်ပါက ပုံ ၃-က တွင် ဖော်ပြထားသည့် Confusion Matrix ကို ရရှိပါသည်။

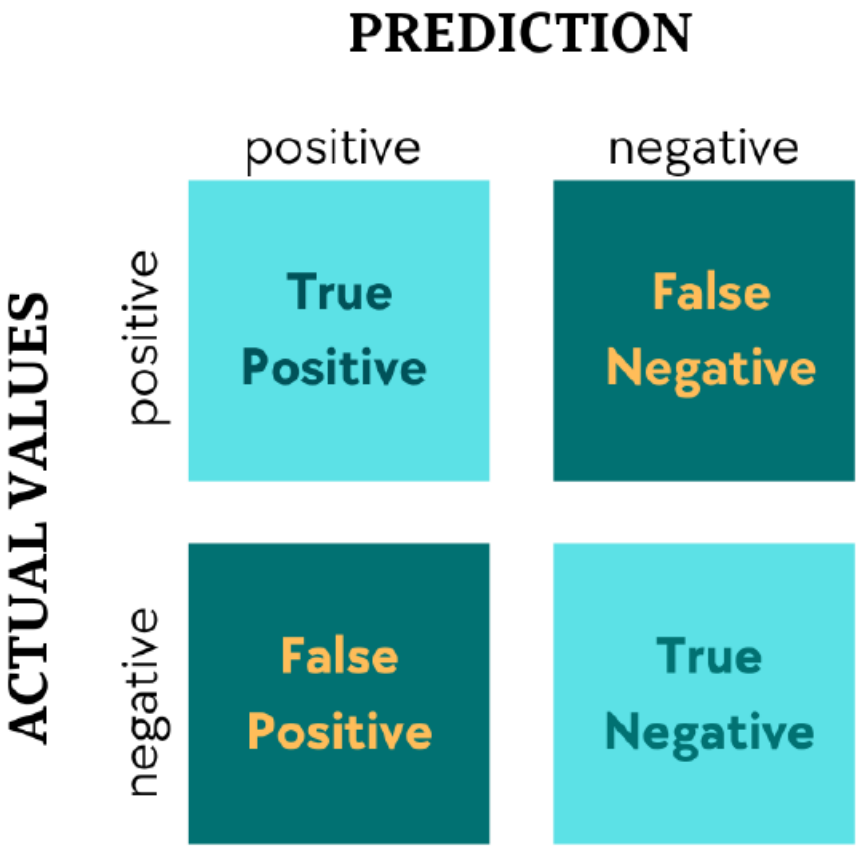
Predicted output	0	1	0	0	1	1
Actual Label	0	0	1	0	0	1

ဇယား (၃-က) ခန့်မှန်း ရလဒ်နှင့် မူလ တန်ဖိုး



ပုံ (၃-က) Confusion Matrix (ဇယား ၃ -က၏ ရလဒ်)

၁၂။ Confusion Matrix ဆိုသည်မှာ True **Positive** ၊ True **Negative** ၊ False **Negative** နှင့် False **Positive** တို့ကို Squared Matrix ဖြင့် ဖော်ပြခြင်း ဖြစ်ပြီး ယေဘုယျ Confusion Matrix မှာ ပုံ (၃-ခ)တွင် ဖော်ပြထားသည့် အတိုင်း ဖြစ်သည်။



ပုံ (၃-ခ) Confusion Matrix

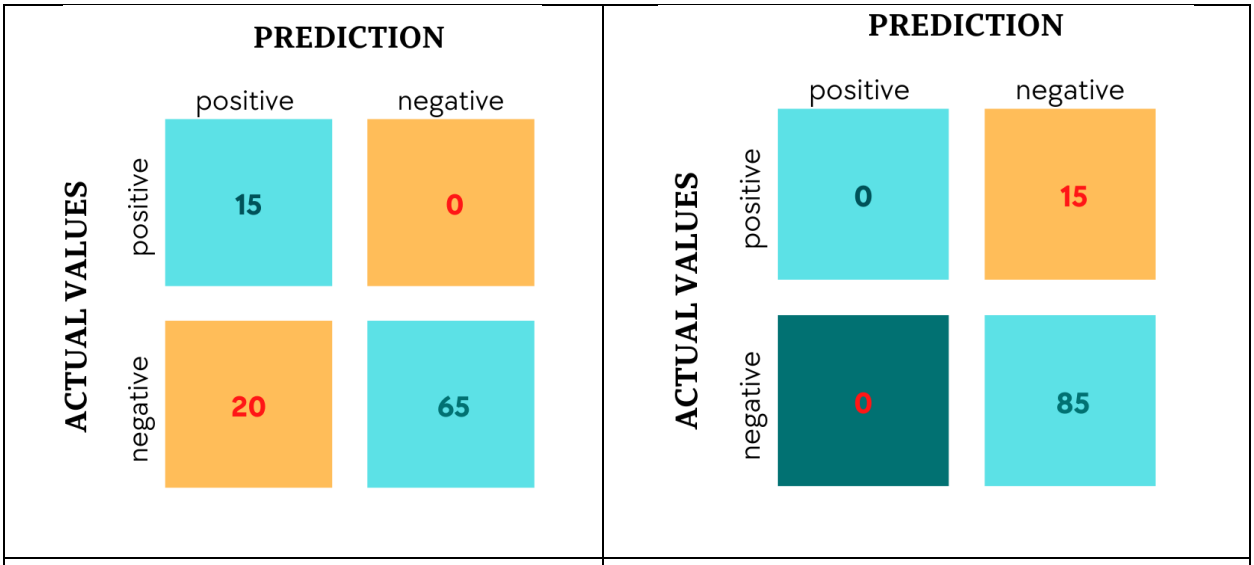
၁၃။ Confusion Matrix သည် အခြား Evaluation Metric များကို သင်ယူရန် အတွက် အရေးကြီးသည့် အခြေခံ သဘောတရား တစ်ခုဖြစ်သည်။ ဤ စာအုပ်တွင် Confusion Matrix ၏ Column သည် ခန့်မှန်းတန်ဖိုးများကို ရည်ညွှန်းပြီး Row သည် မူလတန်ဖိုးကို ရည်ညွှန်းသည်။ Confusion Matrix ကို သတ်မှတ်နိုင်သည့် အခြားနည်းလမ်းတစ်ခုမှာ Row ကို ခန့်မှန်းတန်ဖိုးဟု သတ်မှတ်၍ Confusion Matrix ကို မူလတန်ဖိုး ဟု သတ်မှတ်ခြင်း ဖြစ်သည်။

Accuracy

၁၄။ Classification Model တစ်ခု၏ ခန့်မှန်းချက် တိကျမှုကို တိုင်းတာခြင်းဖြစ်ပြီး သရုပ်ဖော်အားဖြင့် အောက်ပါ အတိုင်း တွက်ချက်နိုင်သည်။

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

၁၅။ Accuracy တန်ဖိုးသည် Balanced Data-set များအတွက် အသုံးဝင်သည့် Metric တစ်ခု ဖြစ်သော်လည်း Imbalanced Data-set များအတွက်မူ တိကျမှု မရှိနိုင်ပါ။ ဥပမာ - Negative Case ၈၅ ခု နှင့် Positive Case ၁၅ ခု ရှိသော Data-set တစ်ခု ကို စဉ်းစားကြည့်ပါ။Model 1 ၏ Accuracy တန်ဖိုးမှာ ၈၀ ရာခိုင်နှုန်းသာရှိပြီး Model – 2 ၏ Accuracy တန်ဖိုးမှာ ၈၅ ရာခိုင်နှုန်း ရှိသည်ကို တွေ့ရမည်။



ပုံ (၃-၈) Confusion Matrix for Model 1 and Model 2

၁၆။ သို့သော် Model 1 နှင့် Model – 2 ၏ Confusion Matrix တန်ဖိုးများကို သေချာလေ့လာကြည့်မည်ဆိုပါက Model – 2 သည် Positive Case တစ်ခုကိုမှ ရှာဖွေတွေ့ရှိ ခြင်းမရှိသည်ကို တွေ့နိုင်ပါသည်။ အကယ်၍ အထက်ပါ Classification Model များသည် ကင်ဆာ ဖေဒနာကို စမ်းသပ်သည့် Model ဖြစ်ပါက Model – 2 သည် လာသမျှ လူနာကို ကင်ဆာ မရှိပါဟု နံမည်တပ်ပေးလိုက်ခြင်း ဖြစ်ရာ ကင်ဆာ ဖေဒနာသည်များအတွက် အန္တရာယ် အလွန်များသော Model တစ်ခုဖြစ်သည်။

၁၇။ သို့ဖြစ်ရာ Imbalanced Data-set များဖြစ်ပါက အခြား Evaluation Metric များကို လည်း ထည့်သွင်းစဉ်းစားရန် လိုအပ်သည်။

Precision

၁၈။ Precision သည် Model မှ Positive ဟု ခန့်မှန်းပေးလိုက်သည့် Case များအနက် ရာခိုင်နှုန်း မည်မျှ မှန်ကန်သည်ကို တွက်ချက်ခြင်း ဖြစ်သည်။ Precision ကို သချိာ ညီမျှခြင်း အားဖြင့် အောက်ပါ အတိုင်း တွက်ချက်နိုင်သည်။

$$Precision = \frac{TP}{TP + FP}$$

၁၉။ ပုံ (၃-ဂ) တွင် Model 1 သည် လူနာ ၃၅ဦး ကို Positive ဟု သတ်မှတ်ပေးလိုက်ရာ အဆိုပါ လူနာ ၃၅ ဦးအနက် ၁၅ ဦးသာ အမှန်ဖေဒနာ ခံစားနေရခြင်း ဖြစ်သည်။ သို့ဖြစ်ရာ Model 1 ၏ Precision တန်ဖိုးသည် ၄၃ ရာခိုင်နှုန်းသာ ရှိသည်ကို တွေ့ရမည်။ Positive ဟု မှားယွင်းစွာ သတ်မှတ်ခံလိုက်ရသည့် လူနာများမှာ စိတ်ဆင်းရဲ ကိုယ်ဆင်းရဲ ဖြစ်စေနိုင်သော်လည်း ကင်ဆာလူနာကို ကင်ဆာမရှိဟု သတ်မှတ်လိုက်သည်ထက်တော့ ဘေးကင်းသေးသည်ကို တွေ့နိုင်ပါသည်။ Model – 2 ၏ Precision တန်ဖိုးမှာ သုည ဖြစ်သည်။

Recall

၂၀။ Recall သည် Positive Case များအနက်မှ ရာခိုင်နှုန်းမည်မျှကို ရှာဖွေနိုင်သည်ကို တွက်ချက်ခြင်း ဖြစ်သည်။Recallကို သင်္ချာညီမျှခြင်းအားဖြင့် အောက်ပါအတိုင်း တွက်ချက် နိုင်သည်။

$$Recall(\text{sensitivity}) = \frac{TP}{TP + FN}$$

၂၁။ ပုံ (၃-ဂ) တွင် Model 1 သည် Positive ပေဒနာရှင် ၁၅ ဦးလုံးကို မှန်ကန်စွာ ရှာဖွေနိုင်ရာ Model 1 ၏ Recall တန်ဖိုးသည် ၁၀၀ ရာခိုင်နှုန်း ဖြစ်ပြီး Model – 2 ၏ Recall တန်ဖိုးမှာ သုည ဖြစ်သည်။

True Negative Rate (Specificity)

၂၂။ True Negative Rate (Specificity) သည် Negative Case များအနက်မှ ရာခိုင်နှုန်း မည်မျှကို မှန်ကန်စွာရှာဖွေနိုင်သည်ကို တွက်ချက်ခြင်း ဖြစ်သည်။ True Negative Rate (Specificity) ကို သင်္ချာညီမျှခြင်းအားဖြင့် အောက်ပါအတိုင်း တွက်ချက် နိုင်သည်။

$$TNR(\text{Specificity}) = \frac{TN}{FP + TN}$$

၂၃။ ပုံ (၃-ဂ) တွင် Model 1 သည် Negative Case ၈၅ ဦးအနက်မှ ၆၅ဦးကိုသာ မှန်ကန်စွာ ရှာဖွေနိုင်ရာ Model 1 ၏ Specificity တန်ဖိုးသည် ၇၆ ရာခိုင်နှုန်း ဖြစ်ပြီး Model – 2 ၏ Specificity တန်ဖိုးမှာ ၁၀၀ ရာခိုင်နှုန်း ဖြစ်သည်။

F-score

၂၄။ **F-score သည်** Recall နှင့် Precision - ၂ ခုလုံးကို ထည့်သွင်းစဉ်းစားထားသည့် Evaluation Metric တစ်ခု ဖြစ်သည်။ F-score ၏ သင်္ချာ ညီမျှခြင်း မှာ အောက်ပါအတိုင်း ဖြစ်သည်။

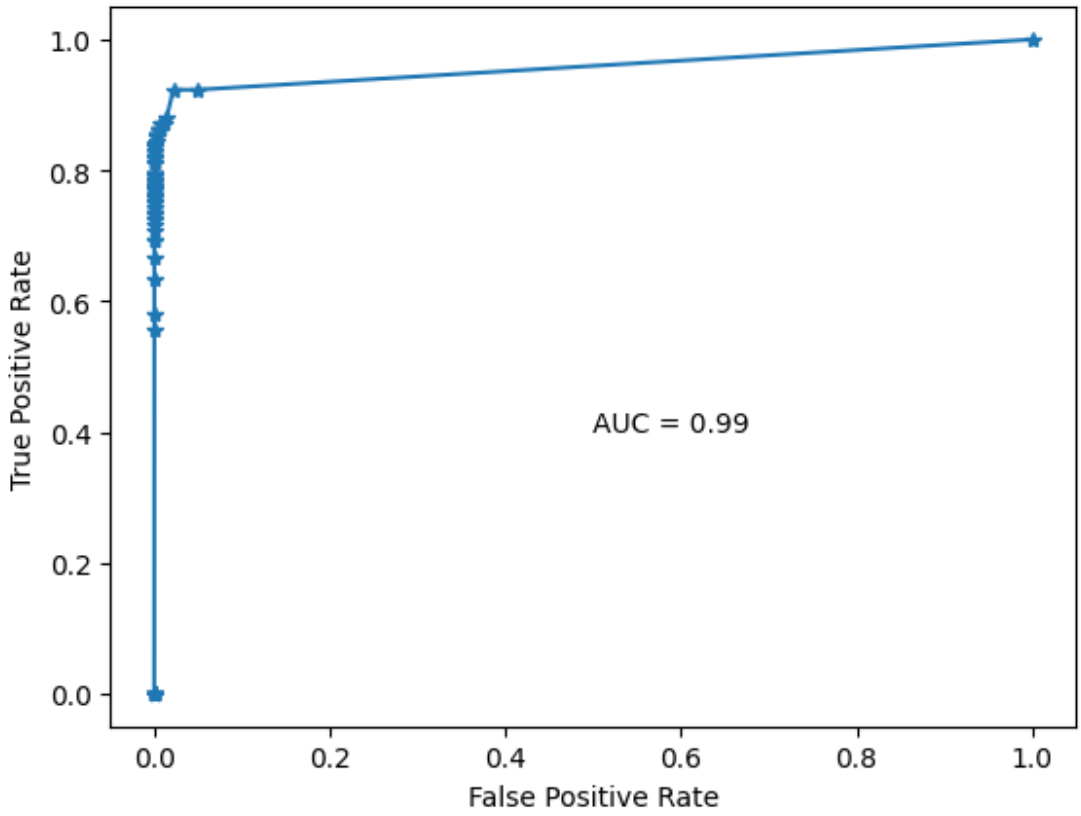
$$F-score = \frac{2 * Recall * Precision}{Recall + Precision}$$

၂၅။ အထက်ပါ ညီမျှခြင်းအရ Recall နှင့် Precision - ၂ ခုလုံးမြင့်မှသာ F-score တန်ဖိုး မြင့်မည် ဖြစ်သည်။ သို့ဖြစ်ရာ Positive နှင့် Negative Case - ၂ ခုလုံး အရေးကြီးသည့် ပုစ္ဆာများတွင် F-score တန်ဖိုးကို သုံး၍ တိုင်းတာသင့်သည်။

Area under the ROC (AUROC or AUC)

၂၆။ အခြားအသုံးများသည့် Evaluation Metric တစ်ခုမှာ ROC (receiver operating characteristic) curve ၏ စုစုပေါင်း ဧရိယာကို တွက်ချက်ခြင်းဖြစ်သည်။ ROC curve ဆိုသည်မှာ True Positive Rate (Recall) နှင့် False Positive Rate (1-Specificity) ၏ ဆက်သွယ်ချက်ကို ဖော်ပြသည့် curve ဖြစ်သည်။

၂၇။ Classifier အတော်များများတွင် ရလဒ် အဖြေမှာ Positive ဖြစ်နိုင်သည့် ရာခိုင်နှုန်း ဖြစ်ပြီး အဆိုပါ ရာခိုင်နှုန်းသည် နံပါတ်တစ်ခု (Threshold) ထက် ကျော်လွန်ပါက Positive Case ဟု သတ်မှတ်ခြင်း ဖြစ်သည်။ သို့ဖြစ်ရာ Threshold ၏ တန်ဖိုးကို မူတည်၍ Positive နှင့် Negative အရေအတွက် ကွာသွားမည်ဖြစ်သည်။ Threshold တန်ဖိုး များလာသည် နှင့်အမျှ True Positive အရေအတွက် လျော့ကျလာပြီး True Negative အရေအတွက် မြင့်တက်သွားမည် ဖြစ်သည်။



ပုံ (၃-ဃ) Example ROC curve

၂၈။ ROC curve သည် Threshold တန်ဖိုး အမျိုးမျိုးအတွက် ရရှိသည့် True Positive Rate နှင့် False Positive Rate ၏ အရေအတွက်ကို ဖော်ပြသည့် curve ဖြစ်သည်။

- Point (၀,၀) တွင် မည်သည့် Positive case ကိုမှ ရှာဖွေနိုင်ခြင်းမရှိပါ။ သို့သော် Negative case အားလုံးကို ရှာဖွေနိုင်သည်။
- Point (၀,၁) တွင် Positive နှင့် Negative case အားလုံးကို ၁၀၀ ရာခိုင်နှုန်း ရှာဖွေနိုင်သည်။ (**အကောင်းဆုံး အနေအထား ဖြစ်သည်**)။
- Point (၁,၁) တွင် Positive case အားလုံးကို ၁၀၀ ရာခိုင်နှုန်း ရှာဖွေနိုင်သော်လည်း Negative case ကို ရှာဖွေနိုင်ခြင်း မရှိပါ။
- Point (၁,၀) တွင် Positive နှင့် Negative case တစ်ခုကိုမှ ရှာဖွေနိုင်ခြင်း မရှိပါ။

၂၉။ Classifier ၏ ROC curve သည် Point (၀,၀) , Point (၀,၁) နှင့် Point (၁,၁) ကို ဖြတ်သွားပါက ROC curve အောက်ရှိ စုစုပေါင်း ဧရိယာ ပမာဏသည် တစ် (သို့မဟုတ်) ၁၀၀ ရာခိုင်နှုန်း ဖြစ်သည်။ True Positive Rate နှင့် False Positive Rate - ၂ ခုလုံးကို တစ် ဖြစ်စေမည့် Threshold တန်ဖိုးကို အသုံးပြုပါက Classifier သည် Positive နှင့် Negative case အားလုံးကို ၁၀၀ ရာခိုင်နှုန်း ရှာဖွေနိုင်သည်။

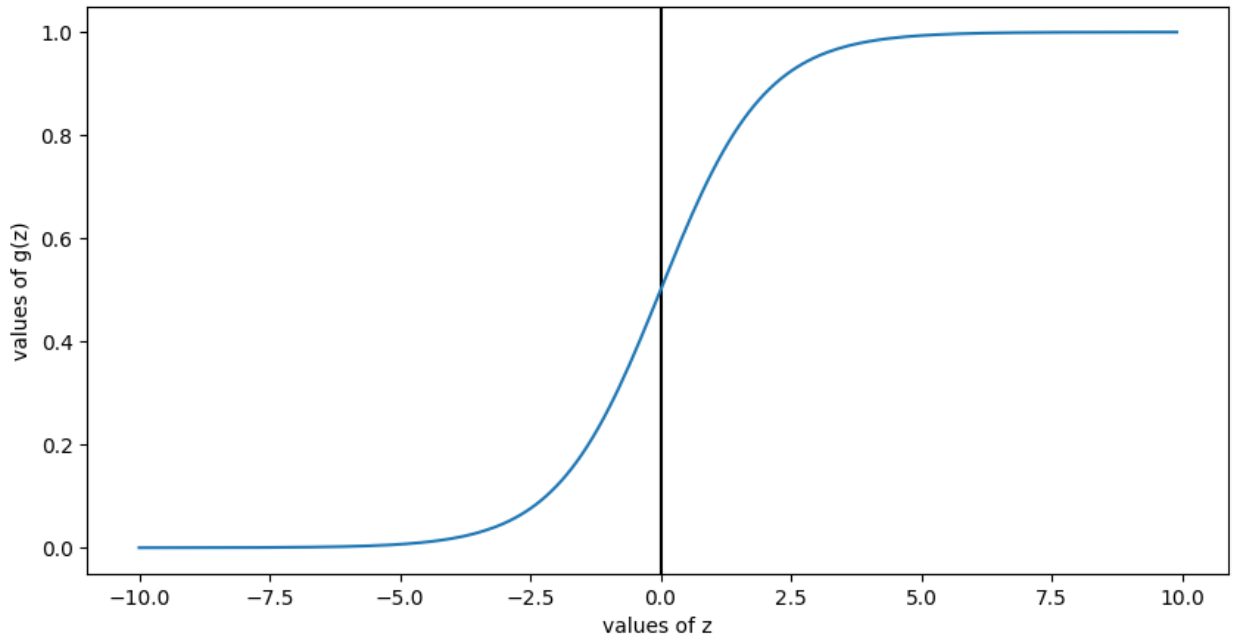
၃၀။ ပုံ ၃-ဃ တွင် ဖော်ပြထားသည့် နမူနာ ROC curve ကို ကြည့်မည်ဆိုပါက ROC curve သည် Point (၀,၁) ကို မဖြတ်သွားသော်လည်း ထိလုနီးပါး ရှိသည်ကို တွေ့ရမည် ဖြစ်သည်။ တနည်းဆိုသော် ပုံ ၃-ဃ တွင် ဖော်ပြထားသော Classifier သည် Threshold တန်ဖိုး တစ်ခု၌ Positive နှင့် Negative case အားလုံးနီးပါးကို ရှာဖွေနိုင်သည်။ ပုံ ၃-ဃ အတွက် ROC curve အောက်ရှိ စုစုပေါင်း ဧရိယာ ပမာဏသည် ၀.၉၉ (သို့မဟုတ်) ၉၉ ရာခိုင်နှုန်း ဖြစ်သည်။

Logistic Regression

၃၁။ Classification Method များအနက် အသုံးအများဆုံး Method တစ်ခုမှာ Logistic Regression Classifier ဖြစ်သည်။ Logistic Regression တွင် အောက်ပါ Sigmoid Functionကို အသုံးပြု၍ Target ($\tilde{y}$) ၏ တန်ဖိုးကို ခန့်မှန်းသည်။

$$\tilde{y} = g(\theta^T x) = \frac{1}{1 + \exp(-\theta^T x)},$$

၃၂။ Linear or Polynomial Regression မှ ရရှိလာသည့် တန်ဖိုးသည် Sigmoid Function ၏ input value ဖြစ်ပြီး Logistic Regression မှ ခန့်မှန်းလိုက်သည့် ($\tilde{y}$) ၏ တန်ဖိုးမှာ သုည နှင့် တစ် အကြားတွင် ရှိမည် ဖြစ်သည်။ Sigmoid Function ၏ ရလဒ် (သို့မဟုတ်) ($\tilde{y}$) ၏ တန်ဖိုးကို အောက်ပါ ပုံ(၃-c) တွင် ဖော်ပြထားသည်။



ပုံ (၃-c) ။ Output of Sigmoid Function

၃၃။ Binary Classification ပုစ္ဆာအတွက် Logistic Regression မှ ရရှိလိုသည့် output တန်ဖိုးမှာ သုည သို့ တစ် ဖြစ်ရာ Logistic Regression အတွက် Cost Function ကို အောက်ပါ အတိုင်း သတ်မှတ်သည်။ z ၏ တန်ဖိုးမှာ Linear or Polynomial မှ ရရှိသည့် ရလဒ်ဖြစ်ပြီး $z = \theta^T X_i$ ဖြစ်သည်။

$$J(\theta) = -y_i \log(g(z)) - (1 - y_i) * \log(1 - g(z))$$

၃၄။ အကယ်၍ Logistic Regression မှ ခန့်မှန်းလိုက်သည့် $\tilde{y} = g(z)$ ၏ တန်ဖိုးနှင့် မူလတန်ဖိုး y_i တူပါက အထက်ပါ cost function ၏ တန်ဖိုးသည် သုညကို ရရှိပါမည်။ ဇယား ၃-ခ တွင် တွက်ချက်ပုံ အဆင့်ဆင့်ကို ဖော်ပြထားသည်။

Actual	Estimated label	cost
		$J(\theta) = -\log(g(z))$
$y = 1$	$g(z) = 1$	$J(\theta) = -\log(1) = 0$
$y = 1$	$g(z) = 0$	$J(\theta) = -\log(0) = 1$
		$J(\theta) = -\log(1 - g(z))$
$y = 0$	$g(z) = 1$	$J(\theta) = -\log(1 - 1) = -1$
$y = 0$	$g(z) = 0$	$J(\theta) = -\log(1 - 0) = 0$

ဇယား ၃ -ခ ၊ Results of Cost Function for Logistic Regression

Implementation of Logistic Regression in Python

၃၅။ အောက်ပါ Program သည် ငွေချေးသူများအနေဖြင့် default ဖြစ်မည် / မဖြစ်မည် ကို Logistic Regression သုံး၍ ခန့်မှန်းတွက်ချက်ပုံ အဆင့်ဆင့်ကို ဖော်ပြထားခြင်း ဖြစ်သည်။ ယခု ပုစ္ဆာအတွက် Fraud Data-set ကို Public GitHub Repo [8] မှ ရယူထားပြီး အဆိုပါ Fraud Data Set သည် Client ပေါင်း ၂၁,၆၉၃ ဦး ၏ အချက်အလက်များကို စုဆောင်းထားပြီး အချက်အလက် များမှာ Independent Variable - ၂၉ ခုနှင့် Target Column (default or non-default) တစ်ခု ၊ စုစုပေါင်း Column အရေ အတွက် ၃၀ ပါဝင်သည်။

၃၆။ အခန်း (၂) တွင် ရှင်းပြခဲ့သည့်အတိုင်း Project Implementation တွင် data preparation, train-test-split, model implementation နှင့် model evaluation အဆင့် တို့ ပါဝင်ပါသည်။

```
# ================================================#
import pandas as pd

df=pd.read_csv('..\\data\\fraud.csv')
y = df['Class'].values
X = df.drop(columns = 'Class').values

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X,y,
                                   test_size = 0.33,
                                   random_state=1)
```

```
#-------------------------------------------------
## Using piepline to implement Logistic regression ##
#-------------------------------------------------
from sklearn.preprocessing import StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.pipeline import Pipeline

steps = [('scaler', StandardScaler()),
         ('logReg', LogisticRegression())]

clf_pipeline = Pipeline(steps)
clf_pipeline.fit(X_train, y_train)
#-------------------------------------------------
## Model Evaluation ##
#-------------------------------------------------
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score

ypred_test = clf_pipeline.predict(X_test)
mat_clf = confusion_matrix(y_test, ypred_test)
report_clf = classification_report(y_test, ypred_test)

ypred_testP = clf_pipeline.predict_proba(X_test)
auc = roc_auc_score(y_test, ypred_testP[:,1])
# ================================================#
```

၃၇။ Logistic Regression Classifier ကို တည်ဆောက်ရန် အတွက် Python sklearn Library မှ Linear Model Module အောက်ရှိ Logistic Regression Function ကို အသုံးပြုထားပြီး Evaluation Metric များကို တွက်ချက်ရန် အတွက် Metrics Module အောက်ရှိ 'classification_report', 'confusion_matrix', 'roc_auc_score' Function များကို အသုံးပြုထားပါသည်။

KNN Classifier

၃၈။ KNN Classifier သည် နားလည်ရန် အလွယ်ကူဆုံး Machine Learning Classifier တစ်ခု ဖြစ်ပါသည်။ မြန်မာ စကားပုံတစ်ခု ဖြစ်သည့် “သူတော်ချင်းချင်း သတင်းလွှဲလွှဲ ပေါင်းဖက်တွေ့ “ဆိုသည့် စကားပုံကို နားလည်သည်ဟု ဆိုပါက KNN Classifier ကို နားလည်ရန် မခက်ခဲပါ။

၃၉။ KNN ၏ အရှည်မှာ K-Nearest Neighbor Classifier ဖြစ်ပြီး K မှာ အနီးဆုံး အိမ်နီးချင်း အရေအတွက်ကို ရည်ညွှန်းသည်။ ဥပမာ - 5-NN ဟု ဆိုပါက အရင်းနှီးဆုံး အိမ်နီးချင်း ၅ ဦးဟု ဆိုရပါမည်။ KNN Classifier တွင် Parameter များကို တွက်ချက်ခြင်း မပြုပဲ အသစ်ဝင်လာသည့် ဒေတာအတွက် အိမ်နီးချင်းများကို ဦးစွာ ရှာဖွေပါသည်။ အကယ်၍ 5-NN Classifier တစ်ခုကို တည်ဆောက်မည်ဆိုပါက အိမ်နီးချင်းများထဲမှ အနီးဆုံး ၅ဦးကို ရွေးချယ်ပါမည်။ အကယ်၍ အနီးဆုံး အိမ်နီးချင်း ၅ ဦး အနက် ၃ ဦးမှာ လူကောင်းဖြစ်ပြီး ကျန် နှစ်ဦးမှာ လူဆိုးဖြစ်နေပါက အသစ်ဝင်လာသည့် ဒေတာကို လူကောင်းဟု သတ်မှတ်မည် ဖြစ်သည်။

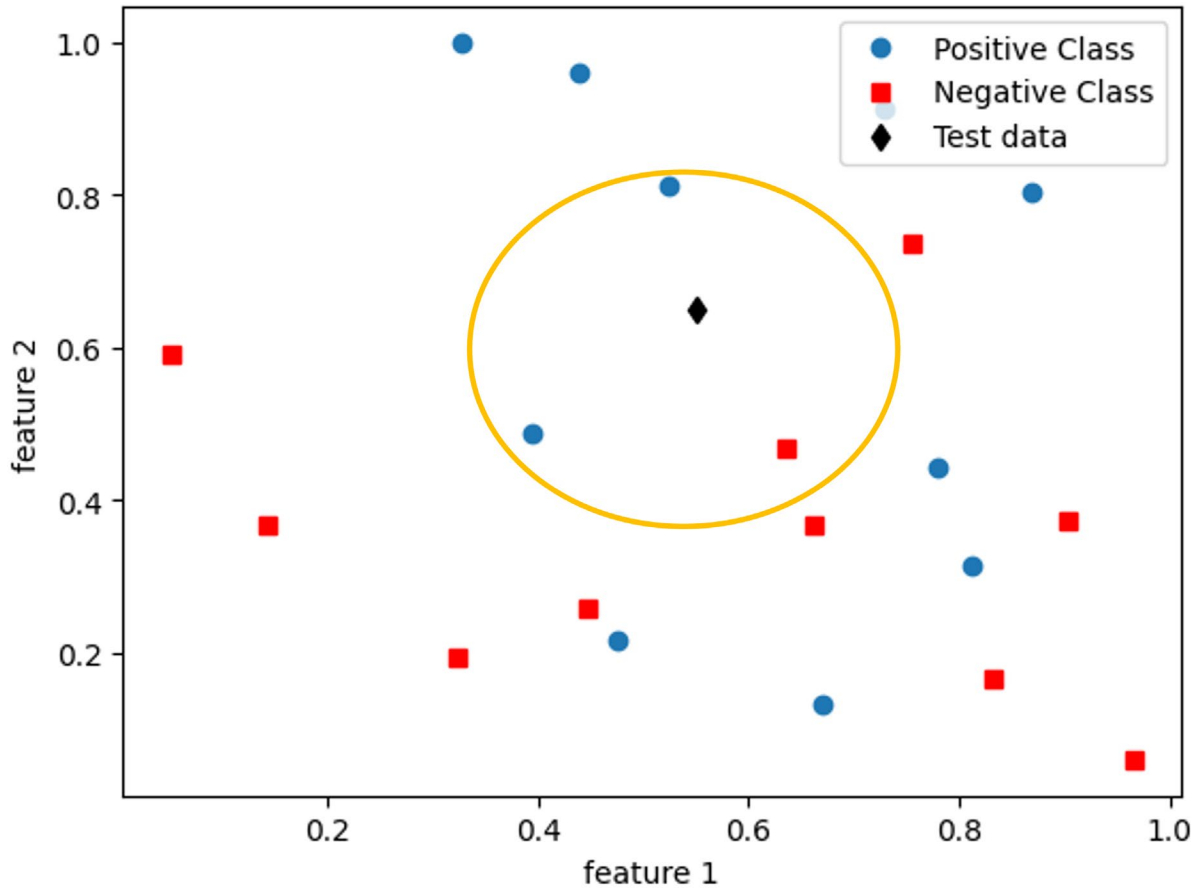
၄၀။ အချုပ်ဆိုရသော် K-Nearest Neighbor Classifier ကို တည်ဆောက်ရာတွင် အောက်ပါ အဆင့် (၄) ဆင့် ပါဝင်ပါသည်။

(က) အဆင့်(၁) တွင် အသစ်ဝင်လာသည့် ဒေတာနှင့် ကြိုတင်စုဆောင်းထားသည့် Training ဒေတာများ၏ အကွာအဝေးကို တွက်ချက် ရပါမည်။

(ခ) အဆင့် (၂) တွင် Training data များထဲမှ အသစ်ဝင်လာသည့် ဒေတာနှင့် အနီးဆုံး ဖြစ်သော ဒေတာ K အရေ အတွက်ကို ရှာဖွေရပါမည်။

(ဂ) အဆင့် (၃) တွင် အနီးဆုံး K ဒေတာ၏ Class အမျိုးအစားများကို Training data ရှိပေးထားချက်နှင့် တိုက်၍ ရှာဖွေရမည်။

(ဃ) အဆင့် (၄) တွင် အသစ်ဝင်လာသည့် ဒေတာကို အဖွဲ့အဝင် အရေအတွက် ပိုများသည့် Class ဟု သတ်မှတ်မည် ဖြစ်သည်။



ပုံ (၃-စ) Demonstration of K-NN Classifier (K = 3)

၄၁။ ပုံ (၃-စ) တွင် KNN Classifier ၏ လုပ်ဆောင်ပုံကို တင်ပြထားပါသည်။ အသစ်ဝင်လာသည့် အမည်းရောင် ဒေတာအတွက် အနီးဆုံး အိမ်နီးချင်း ၃ ဦးကို ရှာဖွေရာ ၂ ဦးမှာ Positive Class မှ ဖြစ်ပြီး ကျန် တစ်ဦးမှာ Negative Class မှ ဖြစ်သည်။ Positive Class အဖွဲ့ဝင် အရေအတွက်သည် Negative Class အဖွဲ့ဝင် အရေအတွက်ထက် ပိုများသည် ဖြစ်ရာ အသစ်ဝင် လာသည့် အမည်းရောင် ဒေတာသည် Positive Class ဝင် ဖြစ်သည်။

၄၂။ KNN Classifier တွင် အသစ်ဝင်လာသည့် ဒေတာနှင့် ကြိုတင်စုဆောင်းထားသည့် Training ဒေတာများ၏ အကွာအဝေးကို တွက်ချက်ရာတွင် Euclidean သို့မဟုတ် Manhattan Distance ကို အသုံးပြုနိုင်သည်။

၄၃။ KNN Classifier တွင် K ၏ တန်ဖိုးကို ကြိုတင် သတ်မှတ်ထားရန် လိုအပ်သည်။ K ၏ တန်ဖိုးသည် Classifier ၏ လုပ်ဆောင်ချက်ကို အများဆုံး သက်ရောက်မှု ရှိသည်။အကယ်၍ K ၏ တန်ဖိုးကို တစ်ဟု သတ်မှတ်ခဲ့ပါက အနီးဆုံး ဒေတာတစ်ခု အပေါ်တွင်သာ မူတည်၍ အသစ်ဒေတာ၏ Class အမျိုးအစားကို ဆုံးဖြတ်မည် ဖြစ်ရာ လက်ရှိ Training Data အတွက် အဖြေမှန်ကို ရရှိနိုင်သော်လည်း အခြား ဒေတာများအတွက်မူ တိကျမှု အားနည်းသွားနိုင်ပါသည်။

၄၄။ ယေဘုယျ ဆိုရသော် သေးငယ်သည့် K တန်ဖိုးသည် အခန်း(၂) – Regression Method များတွင် ဆွေးနွေးခဲ့သည့် Over-Fitting ပြဿနာကို ဖြစ်စေနိုင်ပြီး ကြီးမားသည့် K တန်ဖိုးသည် Under-fitting ကို ဖြစ်စေနိုင်ပါသည်။ သို့ဖြစ်ရာ သင့်တော်သည့် K တန်ဖိုးကို ရွေးချယ်ရန် လိုအပ်သည်။

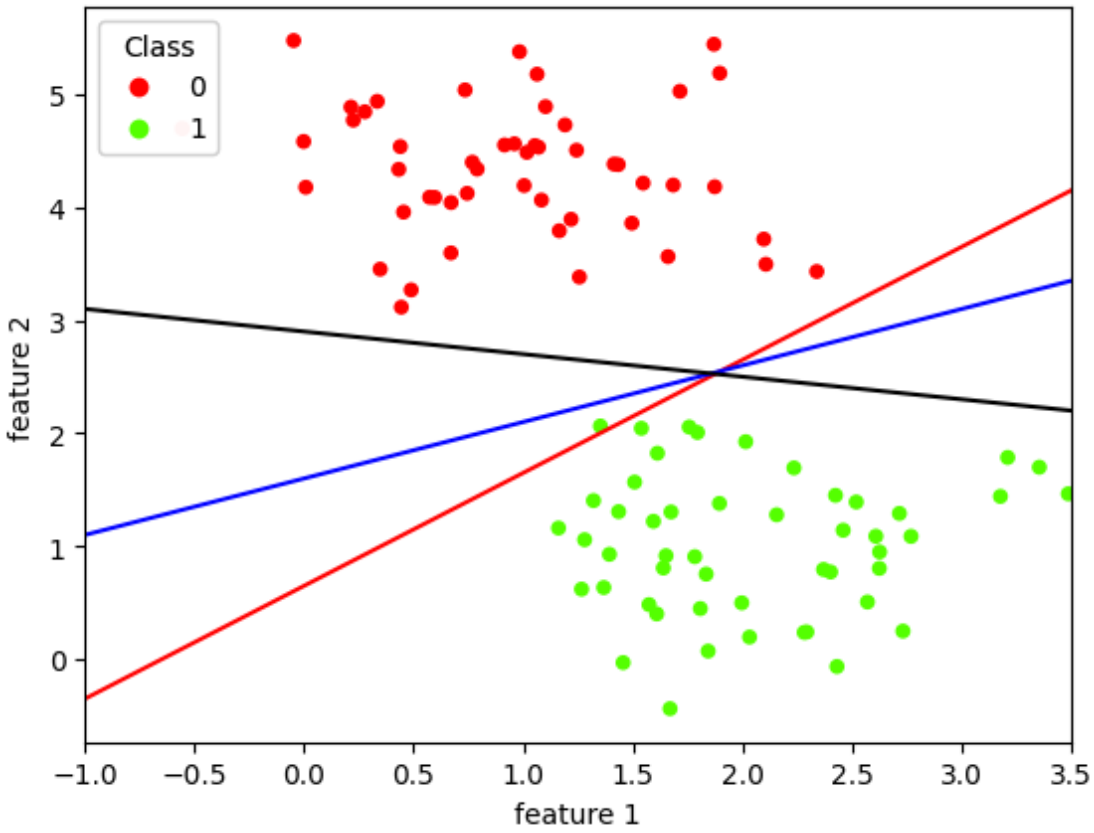
၄၅။ K တန်ဖိုး ရွေးချယ်ရာတွင် သတိပြုရမည့် အချက်မှာ K ၏တန်ဖိုးသည် ‘မ’ ကိန်း ဖြစ်ရမည်ဆိုသော အချက်ပင် ဖြစ်သည်။ အကယ်၍ စုံကိန်းဖြစ်ခဲ့ပါက Positive Class ၏ အဖွဲ့ဝင် အရေအတွက်နှင့် Negative Class ၏ အဖွဲ့ဝင် အရေအတွက် တူညီနေသည့် အခြေအနေမျိုးကို ကြုံရနိုင်သဖြင့် K ၏တန်ဖိုးကို ရွေးချယ်ရာတွင် ‘မ’ ကိန်းကို ရွေးချယ်ရန် လိုသည်။

၄၆။ အောက်ပါ Program သည် ငွေချေးသူများမှ ငွေကြေး ပြန်လည်ဆပ်မည့် အခြေအနေ မရှိ (default) / အခြေအနေ ရှိ (non-default) ကို KNN Classifier သုံး၍ ခန့်မှန်းတွက်ချက်ပုံ အဆင့်ဆင့်ကို ဖော်ပြထားခြင်း ဖြစ်သည်။ ယခု ပုစ္ဆာတွင် K ၏တန်ဖိုးကို ၅ ဟု သတ်မှတ်ထားသည်။

```
# ================================================#
#------------------------------------------------
## Using piepline to implement k-nn classifier ##
#------------------------------------------------
from sklearn.neighbors import KNeighborsClassifier

steps = [('scaler', StandardScaler()),
         ('knn', KNeighborsClassifier(n_neighbors = 5))]

knn_pipeline = Pipeline(steps)
knn_pipeline.fit(X_train, y_train)
#------------------------------------------------
## Model Evaluation ##
#------------------------------------------------
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
from sklearn.metrics import roc_auc_score


ypred_test = knn_pipeline.predict(X_test)
mat_clf = confusion_matrix(y_test, ypred_test)
report_clf = classification_report(y_test, ypred_test)

print(mat_clf)
print(report_clf)

ypred_testP = knn_pipeline.predict_proba(X_test)
auc = roc_auc_score(y_test, ypred_testP[:,1])
print(auc)
# ================================================#
```

SVM Classifier

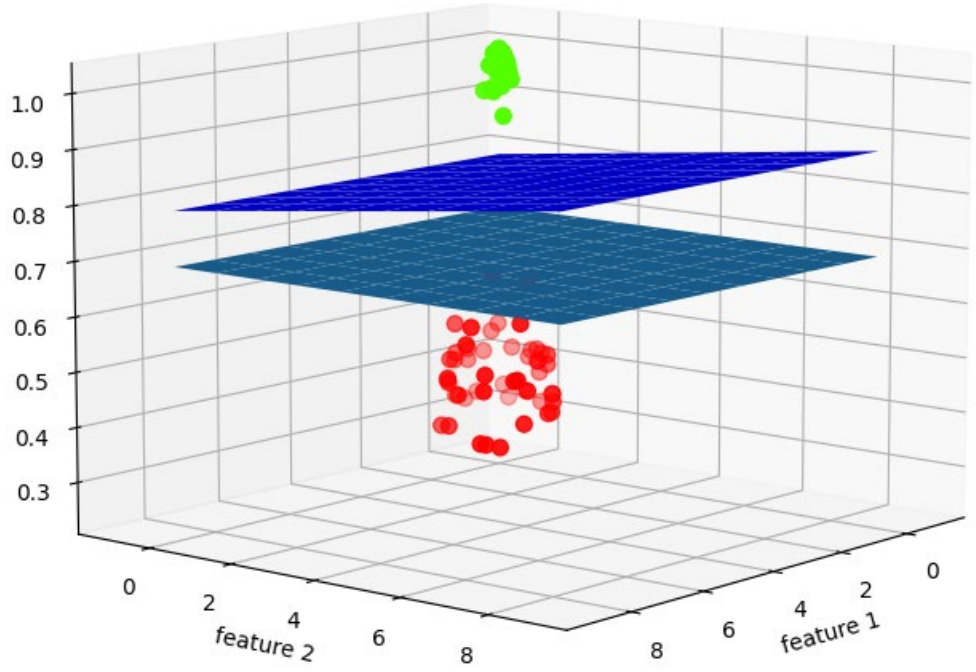
၄၇။ Support Vector Machine (SVM) သည် လူသုံးအများဆုံး Classifier တစ်ခု ဖြစ်သည်။ SVM ကို Classification ပြဿနာအတွက် အဓိက သုံးစွဲကြသော်လည်း Regression ပြဿနာများ အတွက်လည်း သုံးစွဲနိုင်သည်။



ပုံ (၃-ဆ) SVM Classifier in 2 Dimensional Feature Space

၄၈။ SVM Classifier ၏ လုပ်ဆောင်ချက်ကို အောက်ပါ ပုံ (၃-ဆ) တွင် လေ့လာနိုင်သည်။ SVM Classifier ၏ ရည်ရွယ်ချက်မှာ Positive နှင့် Negative Class ကို ခွဲခြားပေးနိုင်မည့် Boundary တစ်ခုကို ရှာဖွေခြင်း ဖြစ်သည်။ ထိုနယ်မြေပိုင်းခြားပေးနိုင်သည့် Boundary ကို **Decision Boundary** ဟု ခေါ်သည်။ ပုံ (၃-ဆ) အရ Decision Boundary သည် မျဉ်းဖြောင့် တစ်ကြောင်းဖြစ်သည်ကို တွေ့နိုင်သည်။

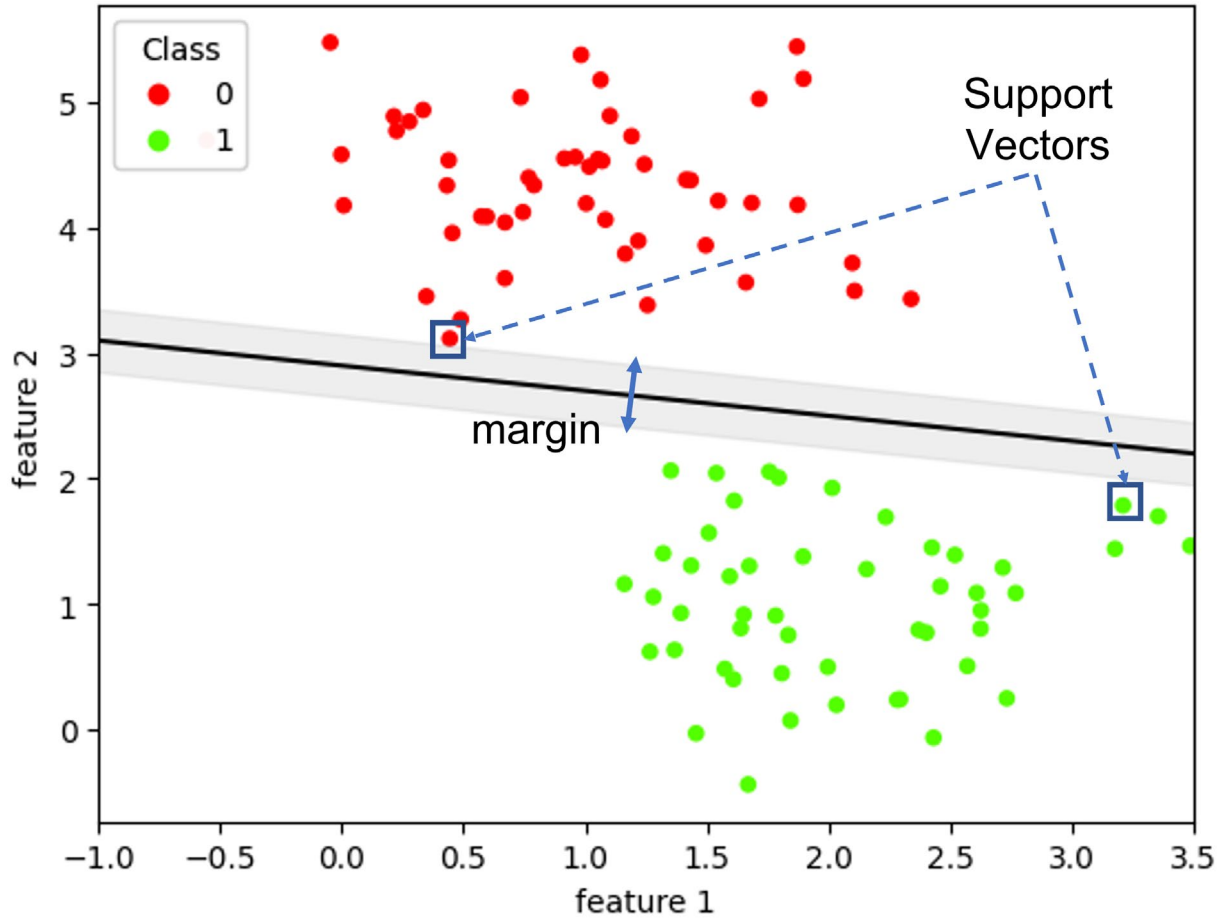
၄၉။ Independent Variable or Feature ၂ ခုထဲကို သာထည့်သွင်းစဉ်းစားရန် လိုအပ်သည့် ပုစ္ဆာများတွင် Boundary သည် မျဉ်းတစ်ခု ဖြစ်သော်သည်း Feature ၂ ခုထက်ပိုလာပါက Plane ဖြင့်သာ ပိုင်းခြားနိုင်မည်။ Classification ပုစ္ဆာများတွင် Feature အရေအတွက် အများအပြားကို ထည့်သွင်း စဉ်းစား လေ့ရှိသည်။ သို့ဖြစ်၍ အဆိုပါ Boundary ကို SVM တွင် **Hyperplane** ဟုသာ သုံးစွဲသည်။ 3-Dimensional Feature Space (Feature ၃ ခုကို အသုံးပြုထားသော) Decision Boundary ကို ပုံ (၃-၈) တွင် ပြသထားသည်။ ၃ ခုထက်ပိုသော Feature Space များအတွက် ပုံဖြင့် ဖော်ယူရန် မလွယ်ကူပါ။



ပုံ (၃-၈)။ SVM Classifier in 3-Dimenstional Feature Space

၅၀။ ပုံ (၃-ဆ) နှင့် ပုံ (၃-၈) တို့ကို လေ့လာကြည့်ပါက Positive နှင့် Negative Class ကို ခွဲခြားပေး နိုင်မည့် Boundary သည် တစ်ခုထက်မက ရှိနိုင်သည်ကို သိနိုင်ပါသည်။ သို့ဖြစ်ရာ SVM Classifier တွင် Boundary နှင့် Positive နှင့် Negative Class - ၂ ခုလုံးမှ

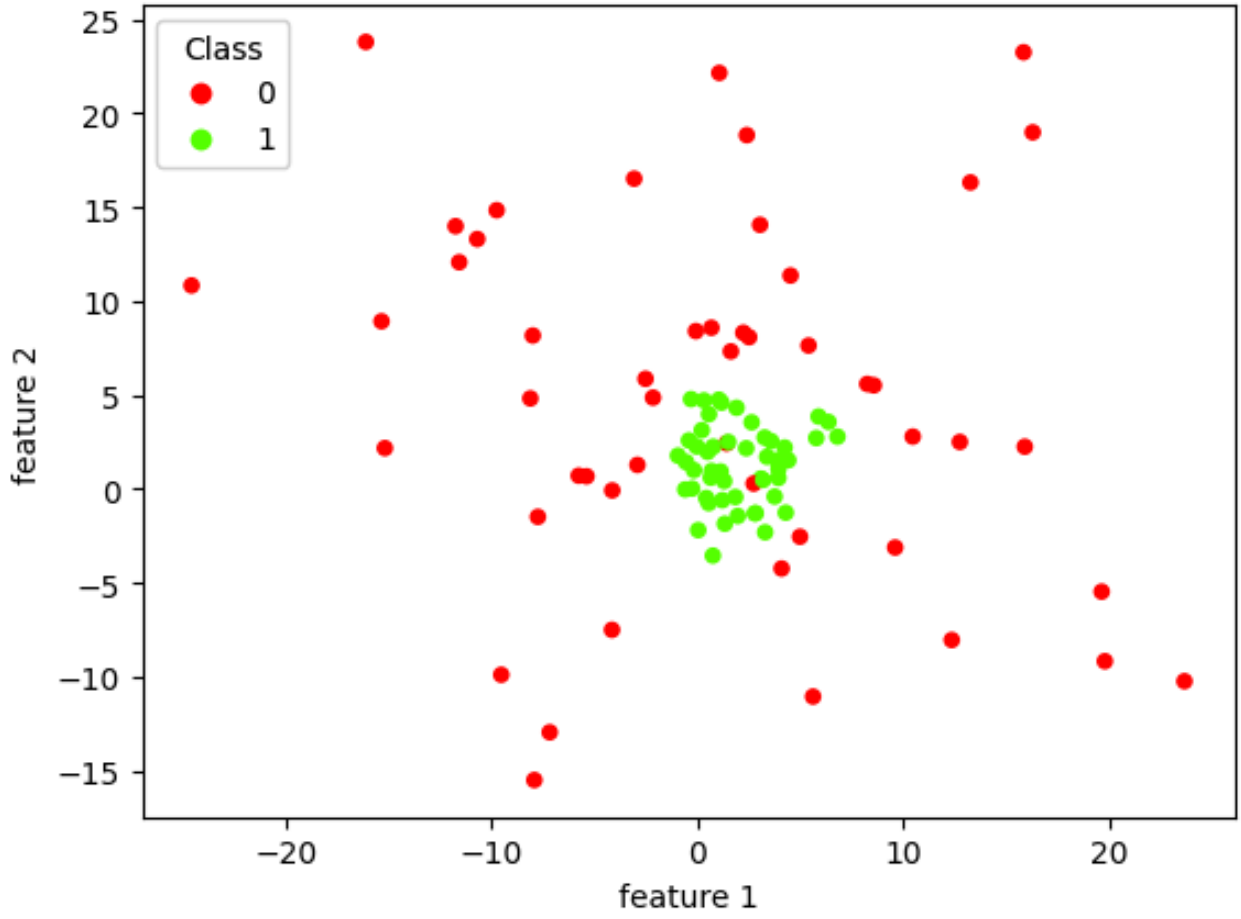
ဒေတာများ အကြားရှိသည့် အကွာအဝေး အကြီး ဆုံး ဖြစ်နိုင်မည့် Boundary ကို Decision Boundary ဟု သတ်မှတ်သည်။ ပုံ (၃-၈) တွင် ခဲရောင်ဖြင့် ပြသထားသည့် Region ကို **Margin** ဟု ခေါ်ပြီး SVM Classifier ၏ တာဝန်မှာ အဆိုပါ Margin ကို အကြီးဆုံး ဖြစ်စေမည့် Boundary ကို ရှာဖွေရန် ဖြစ်သည်။



ပုံ (၃- ၈)။ SVM Classifier : Decision Boundary, Margin and Support Vector

၅၁။ Decision Boundary နှင့် အနီးဆုံး ဖြစ်နေသည့် ဒေတာများကို Support Vector ဟု ခေါ်ပြီး ပုံ(၃- ၈) တွင် လေးထောင့်ကွက်များဖြင့် ပြသထားသည်။ အဆိုပါ Support Vector များကို ဖယ်လိုက်ပါက Decision Boundary ၏ Direction ပြောင်းလဲ သွားမည် ဖြစ်သည်။

၅၂။ ပုံ(၃-ဆ၊ ၃- ၈) တွင် ပြသထားသည့် ဒေတာများသည် Linearly ခွဲခြား၍ ရသည့် ဒေတာများ ဖြစ်သည်။ သို့သော် လက်တွေ့တွင်မူ Linearly ခွဲခြား၍ မရသည့် ဒေတာများ ပါဝင်သည့် ပုစ္ဆာ အများအပြားလည်း ရှိသည်။ ဥပမာ ပုံ (၃-ည) ကို ကြည့်ပါ။ အနီရောင် နှင့် အစိမ်းရောင် ဒေတာများကို မျဉ်းဖြောင့် တစ်ကြောင်းဖြင့် ခွဲခြား၍ မရနိုင်ပါ။



ပုံ (၃-ည)။ Linearly non-separable data set

SVM Kernels

၅၃။ SVM Kernel ဆိုသည်မှာ မူလ Feature Space ရှိ ဒေတာများကို အလွယ်တကူ ခွဲခြားနိုင်မည့် အခြား Higher dimensional space သို့ ရွှေ့ပြောင်းခြင်း ဖြစ်သည်။ အသုံးပြုသည့် Kernel အမျိုးအစား အပေါ်မူတည်၍ အမျိုးအစားများ ကွဲပြားသွားသည်။

Linear Kernel

၅၄။ Linear Kernel အမျိုးအစားကို အောက်ပါအတိုင်း သတ်မှတ်ထားသည်။ Linear Kernel အသုံးပြုသည့် SVM ကို Linear SVM ဟု ခေါ်ဝေါ်၍ အခြား Kernel အမျိုးအစား အသုံးပြုသည့် SVM ကို Non-Linear SVM ဟု ခေါ်ဝေါ်ကြသည် ။ Linear SVM သည် Linearly ခွဲခြား၍ ရနိုင်သော ဒေတာများအတွက်သာ အသုံးဝင်သည်။

$$K(X_1, X_2) = X_1^T X_2$$

၅၅။ Linear SVM ကို Python sklearn Library ရှိ SVM Module ကို အသုံးပြု၍ အလွယ်တကူ တည်ဆောက်နိုင်သည်။ အောက်ပါ Program သည် Linear SVM တစ်ခု၏ တည်ဆောက်ပုံ ဖြစ်သည်။

```
# ================================================#
#------------------------------------------------
## Using piepline to implement SVM classifier ##
#------------------------------------------------
from sklearn.svm import SVC

steps = [('scaler', StandardScaler()),
         ('svc', SVC(kernel = 'linear'))]

svc_pipeline = Pipeline(steps)
svc_pipeline.fit(X_train, y_train)

#------------------------------------------------
```

Polynomial Kernel

၅၆။ Polynomial Kernel တွင် Polynomial equation ကို အသုံးပြု၍ ဒေတာများကို Higher dimensional space သို့ ရွှေ့ပြောင်းသည်။ သင်္ကေတ d သည် Polynomial equation ၏ degree ကို ရည်ညွှန်း၍ သည် ကိန်းသေတစ်ခုဖြစ်သည်။ ကိန်းသေ တန်ဖိုးကို သုည (သို့မဟုတ်) တစ် ဟု သတ်မှတ် လေ့ရှိပြီး d သည် Hyperparameter တစ်ခုဖြစ်သည်။ Polynomial SVM Classifier ကို မတည်ဆောက်မီ Bias နှင့် Variance ကို Balance ဖြစ်စေမည့် degree or d ကို ရှာဖွေရန် လိုအပ်သည်။

$$K(X_1, X_2) = (X_1^T X_2 + c)^d$$

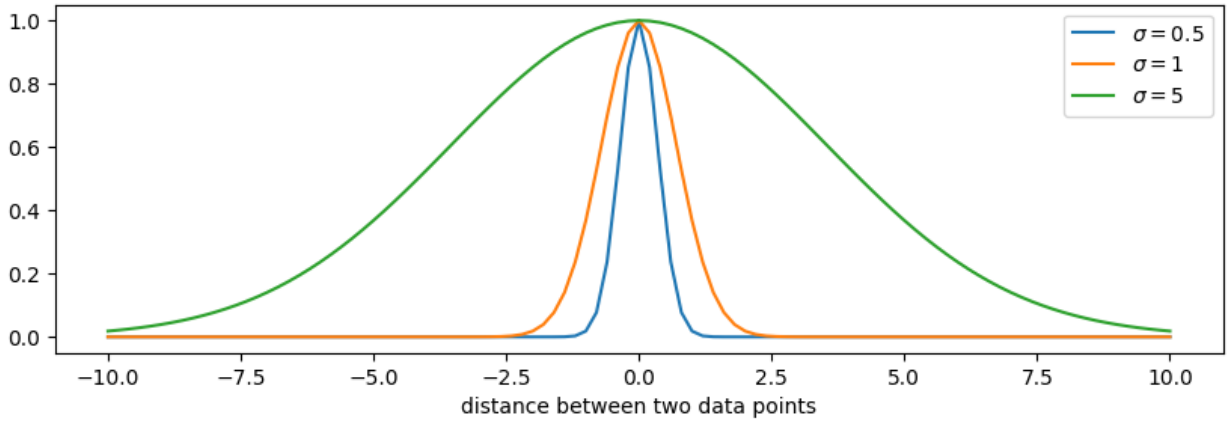
၅၇။ အောက်ပါ Program သည် Python sklearn Library ရှိ SVM Module ကို အသုံးပြု၍ Polynomial SVM တစ်ခုကို တည်ဆောက်ပုံ ဖြစ်သည်။ ယခု Polynomial SVM တွင် degree or d ၏ တန်ဖိုးကို ၅ ဟု သတ်မှတ်ထားသည်။

```
# ================================================#
#------------------------------------------------
## Using piepline to implement SVM Poly classifier ##
#------------------------------------------------
from sklearn.svm import SVC

steps = [('scaler', StandardScaler()),
         ('svc', SVC(kernel = 'poly', degree = 5))]

svc_pipeline = Pipeline(steps)
svc_pipeline.fit(X_train, y_train)

#------------------------------------------------
```

Radial Bias Function (RBF) kernel

၅၈။ RBF Kernel တွင် Gaussian Function ကို အသုံးပြု၍ Kernel ကို သတ်မှတ်ထားသည်။ သင်္ကေတ γ သည် Hyperparameter တစ်ခုဖြစ်ပြီး Kernel ၏ အကျယ်အဝန်းကို ထိန်းချုပ်သည်။ ပုံ (၃-၌) တွင် သင်္ကေတ γ ကို မူတည်၍ Kernel ၏ အကျယ်အဝန်း ပြောင်းလဲသွားပုံကို ပြသထားသည်။

$$K(X_1, X_2) = \exp(-\gamma \| X_1 - X_2 \|^2)$$



ပုံ (၃-၌) သင်္ကေတ γ ကို မူတည်၍ Kernel ၏ အကျယ်အဝန်း ပြောင်းလဲသွားပုံ

၅၉။ သင်္ကေတ γ နှင့် သင်္ကေတ σ သည် ပြောင်းပြန် အချိုးကျသည်။ σ တန်ဖိုး ကြီးလာသည်နှင့် အမျှ γ ၏ တန်ဖိုး ငယ်သွားမည် ဖြစ်သည်။ ပုံ (၃-၌) ကို ကြည့်ပါက σ တန်ဖိုး ကြီးလာသည်နှင့် အမျှ Kernel ၏ အကျယ်အဝန်း ကျယ်လာသည်ကို တွေ့ရမည် ဖြစ်သည်။ တနည်းဆိုသော် γ တန်ဖိုးနှင့် Kernel ၏ အကျယ်အဝန်းသည် ပြောင်းပြန် အချိုးကျသည်။

$$\gamma \propto \frac{1}{\sigma}$$

၆၀။ အောက်ပါ Program တွင် RBF Kernel ကို အသုံးပြု၍ Non-Linear SVM တစ်ခုကို တည်ဆောက်ထားသည်။ Python sklearn Library ရှိ SVM Module တွင် ၏ γ ၏ တန်ဖိုးကို အောက်ပါအတိုင်း သတ်မှတ်သည်။ Nသည် Training Data-set အတွင်းရှိ ဒေတာ အရေအတွက် ဖြစ်သည်။

$$\gamma \propto \frac{1}{N\sigma}$$

```
# ================================================#
#------------------------------------------------
## Using piepline to implement SVM RBF classifier ##
#------------------------------------------------
from sklearn.svm import SVC

steps = [('scaler', StandardScaler()),
         ('svc', SVC(kernel = 'rbf', gamma = 'scale'))]

svc_pipeline = Pipeline(steps)
svc_pipeline.fit(X_train, y_train)

#------------------------------------------------
```

Project: Fraud Detection

၆၁။ ယခု အခန်းတွင် Fraud Detection ပုစ္ဆာ (ငွေချေးသူများအနေဖြင့် default ဖြစ် (သို့မဟုတ်) မဖြစ်) ကို Classifier အမျိုးမျိုးကို အသုံးပြု၍ တွက်ပြ သွားမည် ဖြစ်သည်။ စာပိုဒ် (၃၅) တွင် ရှင်းပြခဲ့သည့် အတိုင်း ယခု ပုစ္ဆာအတွက် Fraud Data-set ကို Public GitHub Repo [8] မှ ရယူထားပြီး အဆိုပါ Fraud Data Set သည် Client ပေါင်း ၂၁,၆၉၃ ဦး ၏ အချက်အလက်များကို စုဆောင်းထားပါသည်။ ပါဝင်သည့် အချက်အလက် အရေအတွက် မှာ Independent Variable - ၂၉ ခုနှင့် Target (default or non-default) တစ်ခု ၊ စုစုပေါင်း Column အရေ ၃၀ ပါဝင်သည်။

၆၂။ Logistic Regression , K-NN Classifier, Linear SVM, Non-Linear SVM (Poly) နှင့် Non-Linear SVM (RBF) တို့ကို တည်ဆောက်ထားပြီး Classifier အသီးသီးအတွက် အသုံးပြုထားသည့် Hyperparameter များကို ဇယား - (၃-၈) တွင် ဖော်ပြထားသည်။

Name	Hyper-parameters	
LR	C = 1.0	
KNN	k = 5	
SVM-Linear	C = 1.0	
SVM-Poly	C = 1.0	degree = 3
SVM-RBF	C = 1.0	$\gamma = \frac{1}{N\sigma_{Xtr}}$

ဇယား (၃-၈) Classifier အသီးသီးအတွက် အသုံးပြုထားသည့် Hyperparameter များ

၆၃။ Classifier အားလုံးအတွက် Implementation ပြုလုပ်ရာတွင် Traing Data-set ကို အသုံးပြုထားပြီး Testing Data-set မှ အချက်အလက်များ မပါဝင်စေရန် ကာကွယ်သည့် အနေဖြင့် Pipeline Function ကို အသုံးပြုထားသည်။

၆၄။ အောက်ပါ Program သည် Logistic Regression Classifier နှင့် K-NN Classifier တို့ကို တည်ဆောက်ထားခြင်းဖြစ်သည်။

```
# ================================================#
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
#-------------------------------------------------
## ------------Logistic Regression----------------##
#-------------------------------------------------

from sklearn.linear_model import LogisticRegression

steps = [('scaler', StandardScaler()),
         ('logReg', LogisticRegression(penalty = "l2",
                                       C = 1.0))]

LR_pipeline = Pipeline(steps)
LR_pipeline.fit(X_train, y_train)
#-------------------------------------------------
## ----------- K-NN Classifier ------------------##
#-------------------------------------------------

from sklearn.neighbors import KNeighborsClassifier

steps = [('scaler', StandardScaler()),
         ('knn', KNeighborsClassifier(n_neighbors = 5))]

knn_pipeline = Pipeline(steps)
knn_pipeline.fit(X_train, y_train)
```

၆၅။ အောက်ပါ Program သည် SVM Classifier - ၃ မျိုးကို တည်ဆောက်ထားခြင်း ဖြစ်သည်။

```
#-------------------------------------------------
## ------------ SVM Classifier ------------------##
#-------------------------------------------------
from sklearn.svm import SVC
## Linear Kernel  ---------------
steps = [('scaler', StandardScaler()),
         ('svc', SVC(kernel = 'linear',
                     class_weight='balanced'))]

svcL_pipeline = Pipeline(steps)
svcL_pipeline.fit(X_train, y_train)
## Polynomial Kernel ----------------------
steps = [('scaler', StandardScaler()),
         ('svc', SVC(kernel = 'poly', degree = 3,
                     class_weight='balanced'))]

svcPoly_pipeline = Pipeline(steps)
svcPoly_pipeline.fit(X_train, y_train)
## RBF Kernel ----------------------
steps = [('scaler', StandardScaler()),
         ('svc', SVC(kernel = 'rbf', gamma = 'scale',
                     class_weight='balanced'))]

svcRBF_pipeline = Pipeline(steps)
svcRBF_pipeline.fit(X_train, y_train)
# ================================================#
```

၆၆။ အောက်ပါ Program သည် Classifier များ၏ လုပ်ဆောင်ချက်ကို ပြန်လည် ဆန်းစစ်ရန် အတွက် Training နှင့် Testing Data-set ၂ ခု လုံးကို အသုံးပြု၍ စမ်းသပ် ထားခြင်း ဖြစ်သည်။ ယခု ပုစ္ဆာတွင် အသုံးပြုထားသည့် Data-set တွင် Non-Default (Negative Class) ၂၁,၃၃၇ ခုနှင့် Default (Positive Class) ၃၅၆ ခု ပါဝင်သည်။ Positive နှင့် Negative Class ကြား ဒေတာ အရေအတွက် ပါဝင်မှု မညီမျှသည့် Imbalanced Data-set ဖြစ်သည်ကို သတိပြုသင့်သည်။

၆၇။ ယခု ပုစ္ဆာတွင် Default ဖြစ်မည့် Client တစ်ယောက်ကိုမှ လွတ်ထွားမသွားစေရန် အရေးကြီးသည်။ တနည်းအားဖြင့် False Negative (Default ဖြစ်မည့် Client ကို လွတ်သွားခြင်း) သည် False Alarm/False Positive (Client ကို Default ဖြစ်မည်ဟု မှားယွင်း သံသယဝင်ခြင်း) ထက် အရေးကြီး သည်။ သို့ဖြစ်ရာ Recall value မြင့်နိုင်သရွေ့ မြင့်ရမည်ဖြစ်သည်။ Classifier များ၏ Performance ကို နှိုင်းယှဉ်ရန်အတွက် Accuracy, Recall နှင့် Area Under the Curve – Metric ၃ ခုကို အသုံးပြုထားသည်။

```
# ================================================#
from sklearn.metrics import classification_report
from sklearn.metrics import roc_auc_score

result_df = pd.DataFrame(columns = ['Tr_accuracy',
                                    'Test_accuracy',
                                    'Tr_recall',
                                    'Test_recall',
                                    'Train_auc',
                                    'Test_auc'])

model_name = [LR_pipeline, knn_pipeline, svcL_pipeline,
              svcPoly_pipeline, svcRBF_pipeline]
```

```
for idx, model in enumerate(model_name):
    ## for training data
    ypred_train = model.predict(X_train)
    report_clf = classification_report(y_train,
                                       ypred_train,
                                       output_dict=True)
    df_r = pd.DataFrame(report_clf).transpose()
    acc_tr = df_r.loc['accuracy', 'recall'].round(3)
    recall_tr = df_r.iloc[1,1].round(3)
    auc_tr = roc_auc_score(y_train, ypred_train)
    ## for testing data
    ypred_test = model.predict(X_test)
    report_clf = classification_report(y_test,
                                       ypred_test,
                                       output_dict=True)
    df_r = pd.DataFrame(report_clf).transpose()
    acc = df_r.loc['accuracy', 'recall'].round(3)
    recall = df_r.iloc[1,1].round(3)
    auc = roc_auc_score(y_test, ypred_test)

    result_df.loc[idx,:]=[acc_tr, acc, recall_tr,
                          recall, auc_tr.round(3),
                          auc.round(3)]
result_df.index = ['LR', 'K-NN', 'SVM-Linear',
                   'SVM-Poly', 'SVM-RBF']
# ================================================#
```

ဖ၈။ ဇယား (၃-ဃ) တွင် Classifier များ၏ Accuracy တန်ဖိုးကို လည်းကောင်း၊ ဇယား (၃-c) တွင် Recall တန်ဖိုးကို လည်းကောင်း၊ ဇယား (၃-စ) တွင် Area under the ROC Curve (AUC) တန်ဖိုးကို လည်းကောင်း နှိုင်းယှဉ်ပြထားသည်။

	Train	Test
LR	0.996	0.996
K-NN	0.997	0.996
SVM-Linear	0.972	0.97
SVM-Poly	0.996	0.99
SVM-RBF	0.991	0.986

ဇယား (၃-ဃ) Classifier များ၏ Accuracy တန်ဖိုး

	Train	Test
LR	0.79	0.788
K-NN	0.822	0.796
SVM-Linear	0.913	0.883
SVM-Poly	0.973	0.796
SVM-RBF	0.991	0.774

ဇယား (၃-င) Classifier များ၏ Recall တန်ဖိုး

	Train	Test
LR	0.895	0.894
K-NN	0.911	0.898
SVM-Linear	0.943	0.927
SVM-Poly	0.984	0.895
SVM-RBF	0.991	0.882

ဇယား (၃-စ) Classifier များ၏ AUC တန်ဖိုး

၆၈။ ဇယား (၃-ဃ) ရှိ Accuracy တန်ဖိုးကို နှိုင်းယှဉ်ကြည့်ပါက Logistic Regression Classifier သည် Training and Testing Data-set ၂ ခု လုံးအတွက် ၁၀၀ ရာခိုင်နှုန်းနီးပါး Accuracy ရှိသည်ကို တွေ့ရသည်။ သို့သော် ဇယား (၃-ဇ) ရှိ Recall တန်ဖိုးကို ယှဉ်ကြည့်ပါက Logistic Regression Classifier ၏ Performance သည် သိသိသာသာ ကျနေသည်ကို တွေ့ရမည်။ တနည်းဆိုသော် Logistic Regression Classifier သည် Default ဖြစ်မည့်သူ၏ ၈၀ ရာခိုင်နှုန်း နီးပါးကိုသာ Detect ပြုလုပ်နိုင်သည်။

၆၉။ SMV-RBF Classifier သည် Training Data-set အတွင်းရှိ Default ဖြစ်မည့်သူ၏ ၉၉ ရာခိုင်နှုန်းကို Detect ပြုလုပ်နိုင်သည်။ သို့သော် Testing Data-set အတွက်မူ Default ဖြစ်မည့်သူ၏ ၇၇ - ရာခိုင်နှုန်းကိုသာ Detect ပြုလုပ်နိုင်သည်။ တနည်းဆိုသော် SMV-RBF Classifier သည် Training Data-set အတွက် Over-fit ဖြစ်နေသည် ။

၇၀။ ဇယား (၃-ဇ) ရှိ Recall တန်ဖိုးနှင့် ဇယား (၃-စ) ရှိ AUC တန်ဖိုးများကို နှိုင်းယှဉ် ကြည့်ပါက Linear SVM ၏ Performance သည် Training နှင့် Testing Data-set ၂ ခုလုံးအတွက် အမြင့်ဆုံး ဖြစ်သည်ကို တွေ့ရမည်။ ယခု Fraud Detection ပုစ္ဆာအတွက် ပေးထားသော Data-set အရ Linear SVM သည် အသင့်တော်ဆုံး Classifier တစ်ခု ဖြစ်သည်ဟု ဆိုနိုင်ပါသည်။ အထူးသတိပြုရန်မှာ ယခု စာအုပ်တွင် တင်ပြခြင်း မရှိသည့် Gradient Boosting Classifier, Tree-Based Classifiers, Random Forest Classifier, နှင့် Deep Learning Methods ဆိုင်ရာ Classifier များစွာ ရှိသေးပါသည်။ Classifier (တနည်းအားဖြင့် Machine Learning Method)တစ်ခုကို ရွေးချယ်ရာတွင် ဖြေရှင်းရမည့် ပြဿနာ၊ ရရှိနိုင်သည့် ဒေတာ ပမာဏနှင့် အရည်အသွေး၊ Computational Resources စသည့် အချက်များကို ထည့်သွင်း စဉ်းစားရန် လိုအပ်သည်။

အခန်း (၄): ChatGPT မှ အကြံပြုသည့် လေ့လာသင့်သည်များ

၁။ ChatGPT ဆိုသည်မှာ အခြား နည်းပညာဆိုင်ရာ Tool များကဲ့သို့ပင် အသုံးချတတ်လျင် အကျိုးများစေမည့် နည်းပညာတစ်ခု ဖြစ်သည်။ Machine Learning ဘာသာရပ်ကို စတင်လေ့လာသူ အများစု မေးလေ့ရှိသည့် သိထားသင့်သော သနှံ့ာ ခေါင်းစဉ်များ၊ တတ်သင့်သည့် သင်တန်းများနှင့် ဖတ်သင့်သည့် စာအုပ်များကို ChatGPT အား မေးမြန်း၍ ဤအခန်းတွင် တင်ပြပေးလိုက်ပါသည်။

ChatGPT မှ အကြံပြုသော သနှံ့ာ ခေါင်းစဉ်များ

၂။ Machine Learning အပါအဝင် ဘာသာရပ် တော်တော်များများတွင် သနှံ့ာသည် အခြေခံ တစ်ခု အနေဖြင့် ရှိနေပါသည်။ သို့ဖြစ်ရာ Machine Learning ကို လေ့လာရန် သနှံ့ာ သိထားရန် လိုအပ်ပါသလား ဟု မေးမြန်းပါက လိုအပ်ပါသည်ဟု အမြဲပင် ဖြေကြားခဲ့ပါသည်။ Machine Learning ဘာသာရပ် အတွက် သာမက နေ့စဉ် ကြုံတွေ့နေရသည့် လူမှုဘဝတွင်လည်း သနှံ့ာကို အသုံးပြုနေကြရပါသည်။ သို့ဖြစ်ရာ သနှံ့ာကို ကြောက်ရမည့် ဘာသာရပ် တစ်ခု အဖြစ် မမြင်ပဲ စိတ်ဝင်စားစရာ ကောင်းသော ဘာသာရပ်တစ်ခု အဖြစ် မြင်ပါက လေ့လာရာတွင် ပို၍ လွယ်ကူစေမည် ဖြစ်ပါသည်။

၃။ ChatGPT မှ အကြံပြုသော သနှံ့ာ ခေါင်းစဉ်များမှာ အောက်ပါအတိုင်း ဖြစ်ပါသည်။

(က) Linear Algebra

(ခ) Calculus

(ဂ) Probability and Statistics

(ဃ) Optimization

(င) Information theory and entropy

ChatGPT မှ အကြံပြုသော အွန်လိုင်း သင်တန်းများ

၄။ အွန်လိုင်း သင်တန်းများနှင့် ပတ်သက်လျင် ChatGPT ၏ အကြံပြုချက်မှာ ၁၀၀ ရာခိုင်နှုန်း ဘက်လိုက်မှု ကင်းသည် ဟု အာမ မခံနိုင်သော်လည်း ChatGPT အကြံပြုသည့် သင်တန်းများသည် သင်တန်းကောင်းများ ဖြစ်သည်မှာ ငြင်းဆို၍ မရနိုင်ပါ။ စတင်လေ့လာသူများအနေဖြင့် ယခုအခန်းတွင် အကြံပြုထားသည့် သင်တန်းများကို စမှတ်အဖြစ် အသုံးပြု၍ စတင်လေ့လာနိုင်ပါသည်။ ထိုသို့ လေ့လာရင်းမှ မိမိ၏ သင်ကြားမှု ပုံစံနှင့် ကိုက်ညီသော သင်တန်းများကို ကိုယ်ပိုင် ဆုံးဖြတ်ချက်နှင့် ရွေးချယ်လာနိုင်မည်ဟု ယုံကြည်ပါသည်။

၅။ ChatGPT မှ အကြံပြုသောအွန်လိုင်းသင်တန်းများမှာ အောက်ပါအတိုင်း ဖြစ်ပါသည်။

- "Introduction to Machine Learning" by Andrew Ng on Coursera
- "Machine Learning" by Georgia Tech on Udacity
- "Deep Learning" by Andrew Ng on Coursera
- "Applied Data Science with Python" on Coursera
- "Introduction to Machine Learning with Python" by Sarah Guido and Andreas Müller on Coursera
- "Machine Learning A-Z: Hands-On Python and R In Data Science" by Kirill Eremenko and Hadelin de Ponteves on Udemy
- "Machine Learning for Data Science and Analytics" by Columbia University on edX

ChatGPT မှ အကြံပြုသော စာအုပ်များ

၆။ အွန်လိုင်းသင်တန်းများသည် ဘာသာရပ်တစ်ခုကို လျင်လျင်မြန်မြန်နှင့် အလွယ်တကူ လေ့လာရန် အထောက်အပံ့ကောင်းများ ဖြစ်ပါသည်။ အခမဲ့ သင်တန်းများ ရှိသကဲ့သို့ အခကြေးငွေယူ၍ သင်ကြား ပေးနေသည့် သင်တန်းများလည်း အများအပြား ရှိပါသည်။ သို့သော် အရည်အသွေး ပြည့်ဝကောင်းမွန်သည့် သင်တန်းများနည်းတူ အရည်အသွေး မပြည့်မီသည့် သင်တန်းများလည်း ရှိနေသည်ကို သတိပြုရမည် ဖြစ်သည်။ အရည်အသွေး မပြည့်မီသည့် သင်တန်းများကြောင့် စာဖတ်သူ၏ အချိန်နှင့် ခွန်အားကို ဆုံးရှုံးစေရုံမက သင်ယူလိုစိတ်ကိုပါ ဖျက်ဆီးစေနိုင်ရာ သင်တန်းကောင်းများကို ရွေးချယ်တတ်ရန် လိုအပ်ပါသည်။

၇။ စာအုပ်များသည် ဘာသာရပ်တစ်ခု၏ သီအိုရီနှင့်သဘောတရားကို ပြည့်ပြည့်ဝဝ နားလည်နိုင်စေမည့် အဖိုးတန်အထောက်အပံ့များ ဖြစ်ပါသည်။အွန်လိုင်းမှတဆင့် သင်ယူရန် အခက်အခဲရှိနေသူများအတွက် စာအုပ်များမှ လေ့လာသင်ယူနိုင်ပါသည်။

၉။ ChatGPT မှ အကြံပြုသည့် Machine Learning ဆိုင်ရာ စာအုပ်များမှာ အောက်ပါအတိုင်း ဖြစ်ပါသည်။

(က) "Pattern Recognition and Machine Learning" by Christopher M. Bishop

(ခ) Deep Learning" by Yoshua Bengio, Ian Goodfellow, and Aaron Courville

(ဂ) "Machine Learning: A Probabilistic Perspective" by Kevin P.Murphy

(ဃ) "Introduction to Machine Learning" by Alex Smola and S.V.N. Vishwanathan

(င) "The Hundred-Page Machine Learning Book" by Andriy Burkov

(စ) "Python Machine Learning" by Sebastian Raschka and VahidMirjalili

(ဆ) "Machine Learning for Dummies" by John Paul Mueller

Bibliography

[1] OpenAI. Chatbot, 2022. developed by openAI.

[2] Guido van Rossum. Python, 1991. Python Software Foundation.

[3] Fernando Pérez. Juypter notebook, 2012. born from IPython project.

[4] Google Research Team. Google colab, 2018. Free to use.

[5] Microsoft. Visual studio code, 2015. cross-platform code editor.

[6] Myo Thida. Introduction to python, jupyter and google colab, 2022. Free source to learn Python Programming.

[7] kaggle.com. Advertising and sales, 2017. Release with CC0: Public Domain.

[8] 'Yony cherkos. Fraud Data-set. "https://github.com/yonycherkos"

[9] David Cournapeau and Matthieu Brucher. scikitlearn, 2007. Google Summer of Code project.

[10] Chih-Chung Chang Chih-Wei Hsu and Chih-Jen Lin. A practical guide to support vector classification.

ကျေးဇူးတင်လွှာ

ယခု စာအုပ်ဖြစ်မြောက်လာရေးအတွက် စာအုပ်ရေးပေးပါရန် တိုက်တွန်းခဲ့ကြသည့် နိုင်ငံတကာမှ တပည့်များကို အထူးပင် ကျေးဇူးတင်ရှိကြောင်း မှတ်တမ်းတင်အပ် ပါသည်။ ထို့အပြင် မြန်မာစာ စာလုံးပေါင်း အခက်အခဲများကို ကူညီဖြေရှင်း ပေးခဲ့သော ၊ စာကြမ်းမှ စာချော ဖြစ်သည်အထိ စာပြင်ပေးခဲ့ကြသော ပါမောက္ခ ဒေါက်တာ လပြည့်လင်း (YTU) နှင့် ဒေါ်ဇင်မိမိကျော် (SMVTI) တို့ကို အထူးကျေးဇူးတင်ရှိပါသည်။