

# Prediktiv analys för tidig identifiering av stödbehov hos gymnasieelever

David Adamovic

EC Utbildning

Data Science

2024–06

## Abstract

Working with data about people or human performance can be challenging. We face tasks differently and our levels of motivation and capability for solving them are divergent. This report investigates how predictive analysis can identify high school students needing support after the first year based on their grades. Using data from Kaggle's "Student Alcohol Consumption" dataset, which includes socio-economic background, behaviour, and habits of students from two Portuguese high schools. Artificial neural network models were applied to learn the patterns in the data. The results called attention to challenges about data imbalance and limited data quantity yet demonstrated a potential for identifying the risk group correctly. An accurate identification of the students in the risk group can help schools and municipalities to have a better understanding of the amount of resources needed to improve students' academic performance and well-being.

## Förkortningar och Begrepp

Feature – är en kolumn som representerar en aspekt av data som används för att skapa prediktioner.

Target – (target variabel) är en variabel som kan bestå av en eller flera klasser som predikteras med hjälp av features.

Merga – eller sammanslå är en process som kombinerar två eller flera data set till en.

Machine friendly – beskriver data som är förberedda och strukturerade för en maskininlärningsmodell.

## Innehåll

Abstract .....	2
Förkortningar och Begrepp .....	3
1 Inledning.....	1
1.1 Bakgrund .....	1
1.2 Syfte .....	1
2 Teori.....	2
2.1 Artificiella Neurala Nätverk.....	2
2.1.1 Sequential API.....	3
2.1.2 Aktiveringsfunktion .....	3
2.1.3 Keras Tuner .....	3
2.2 Utvärderingsmetoder.....	4
2.2.1 Accuracy.....	4
2.2.2 Confusion Matrix .....	4
2.2.3 Precision .....	5
2.2.4 Recall.....	5
2.2.5 F1 .....	5
3 Metod .....	6
3.1 Datainsamling .....	6
3.2 EDA.....	6
3.3 Förbehandling av data .....	7
3.4 Modellering.....	8
4 Resultat och Diskussion.....	10
5 Slutsatser .....	11
5.1 Möjlighet till vidareutveckling av projektet .....	11
Källförteckning.....	12

# 1 Inledning

## 1.1 Bakgrund

Att gå på gymnasieskola och slutföra en gymnasieexamen har många fördelar för elever. Det upplevs samtidigt en övergång från ungdomen till vuxenlivet där man tar med sig många lärdomar. I gymnasiet har elever chans att utveckla sina intressen, träffa nya vänner och lära sig om ansvaret vilket är avgörande för etablering i arbetsmarknaden. För vissa kan det vara så pass utmanande att de lämnar skolan utan examen. Enligt Skolverket måste gymnasieskolor erbjuda hjälp i form av extra anpassningar och särskilt stöd till elever i behov (Rätt att få hjälp i skolan, 2023).

I rapporten Skolinspektionen, de viktigaste faktorerna till att elever inte lyckas slutföra gymnasieexamen beror på sena upptäckten av stödbehovet, bristande mentorsrollen och elevernas ointresserade delaktighet i stödarbetet (Arbetet med individuella stödinsatser i gymnasieskolan, 2021, p.4).

## 1.2 Syfte

Den här är en data-driven analys med syftet att förutse ett tidigt stödbehov hos gymnasieelever efter första årskursen genom att använda neurala nätverk och maskininlärningsmodeller. Låga ämnesbetyg kommer vara en indikator till elevernas stödbehov.

Data som analyseras kommer från Kaggle, "Student Alcohol Consumption", och innehåller information om elevernas socioekonomiska bakgrund, beteende och vanor från två gymnasieskolor i Portugal. Det är totalt 382 elever som läser matte och portugisiska men data är uppdelad i två dataset för varsitt ämne. Anledningen till det är att vissa kolumner innehåller student-kursrelaterade information såsom betyg och frånvaro (Kaggle, Student Alcohol Consumption, 2019).

För att uppnå syftet kommer följande frågeställningen att besvaras:

1. Kan vi förutse om studenten kommer behöva extra stöd från skolan baserat på studentens socioekonomiska bakgrund och beteende i korrelation med betyg från årskurs ett?
2. Är det möjligt att få en F1 utvärdering på target klassen på över 50% utan data augmentering.

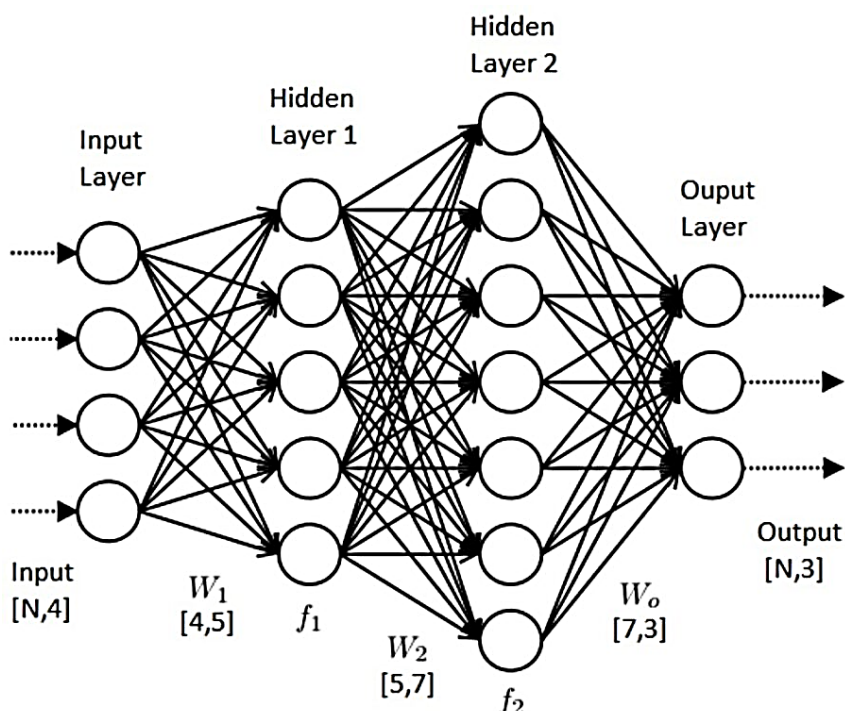
## 2 Teori

I denna del kommer all teori och beskrivningar av olika tekniska verktyg som använts i denna analys för att kunna begripa projektens struktur och metod.

### 2.1 Artificiella Neurala Nätverk

Artificiella Neurala Nätverk (ANN) är en typ av maskininkärningsmodell inspirerad av hjärnans neurala nätverk och Maskininlärningsmetoder. ANN är kraftfulla, flexibla och kan hantera många typer och mängder av data. Dessa modeller används ofta för bildigenkänning, röstigenkänning samt för vanliga regressions – och klassificeringsproblem.

ANN är byggda i lager eller layers som består av neuroner. Första lager kallas för "input layer" och ska ha en input form anpassad till antalet features. Nästa lager kallas för "hidden layers" vars roll är att bearbeta och transformera input data genom olika parametrar och aktiveringsfunktioner. Slutligen finns det ett "output layer" som är anpassad till antal klasser i "target" variabeln. Denna lager producerar prediktioner eller klassificeringar på den bearbetade information från de tidigare lagren.



Figur 1. Visar hur är ANN utformad

I figuren ovan ser vi lager och neuroner och hur de samarbetar med varandra. I de dolda lagren utförs olika beräkningar och transformation av data och resultatet skickas då vidare till neuroner i näst dolda lagret. En uppsättning av dataexempel eller så kallad "batch" används i en sådan iteration

när modellen tränas. Efter en iteration med första batchen jämförs då prediktionerna med de faktiska värden i träningsdata (Géron, Chapter 10, 2012).

#### 2.1.1 Sequential API

Sequential modell är den enklaste typen av Keras eller Tensorflow modell skapad bara för neurala nätverk. Lager är staplade på varandra sekventiellt och det är möjligt att tillägga flera lager som kommer att anslutas automatiskt (Géron, p.299, 2012).

#### 2.1.2 Aktiveringsfunktion

Utdata från ett neuron efter första iterationen läggs i en matematisk funktion som kallas för aktiveringsfunktion. Det finns olika aktiveringsfunktioner såsom Tahn, Leaky ReLU, ELU och de används för mestadels bara i hidden layers. ReLU (Rectified Linear Unit) är vanligt förekommande i de flesta neurala nätverksarkitekturer för sin snabba beräkningshastighet.

$$ReLU(x) = \max(0, x)$$

ReLU kan också användas i output layer men de vanligaste aktiveringsfunktionerna som sätts där är Softmax och Sigmoid som är dock för klassificeringsmodeller. Sigmoid sätts när output ska vara en klass och Softmax när man har multiklassificeringsproblem (Géron, p.292, 2012).

##### 2.1.2.1 Lagren

Ett enkelt neuralt nätverk består oftast av tre lager: input, dolt och output lager (se figur 1). Det dolda lagrets egenskap är att bara vissa neuroner kommer skapa en koppling med neuroner i andra lager. I artikeln om Dense layers, skriver Yugesh Verma att en "Dense" lager har en full uppkoppling mellan neuronerna. Detta påverkar dimensionaliteten från utdata och förtydligar förhållanden mellan värdena i data (Analytics India Magazine, Dense Layer, 2024).

Andra typer av lager såsom Convolution lager, Deconvolution lager och Recurrent lager används för att modellera sekventiellt data som naturligt språk, röstigenkänning och bildigenkänning.

#### 2.1.3 Keras Tuner

För att få det bästa utav modellen så måste hyperparametrarna anpassas (tunas) baserat på modellen, data och målet som ska nås. Hyperparametrar är variabler som styr träningsprocessen. Genom att vrida och vända på dem innan träningen kan vara givande för modellens prestanda. Keras Tuner är ett bibliotek med verktyg som underlättar denna process. Det finns två typer av hyperparametrar. *Modell Hyperparametrar* är de som anpassar bredd och antal hidden layers och *Algoritm Hyperparametrar* är dem som anpassar hastighet och kvalitén av inlärningsalgoritmer (Tensorflow, Introduction to the Keras Tuner, 2023).

##### 2.1.3.1 Optimering

Som en del av Keras Tuner kan optimeringsalgoritmerna inkluderas och justeras. Syftet med optimering är att minska förlustfunktion. Två mest förekommande är Adam och Nadam. Vid tuning så justeras oftast då en inlärningsgrad som styr hur snabbt ändras hyperparametrarna (Géron,

p.356–357, 2012).

#### 2.1.3.2 Regularization

Regularization är en annan teknik som gynnar modellen vid träningsprocessen så att modellen inte överfittar träningsdata. Detta uppstår när modellen anpassas för mycket för träningsdata och presterar sämre med testdata. Enligt Geoffrey Hinton anses Dropout vara mest populära regularizationsalgoritm i neurala nätverk (Geoffrey, 2012). Hemligheten till varför är dropout så pass effektiv ligger i ens hyperparameter  $p$  som reglerar sannolikheten om neuroner blir skippade i träningsprocessen.  $p$  kallas också för "dropout rate" och dess hyperparametrar kan lätt tunas med Keras Tuner (Géron, p.365, 2012).

#### 2.1.3.3 Förlustfunktion

Förlustfunktionen är lite som en utvärderingsmetod som mäter skillnaden mellan target variabeln och prediktioner. Den kan anpassas för att bättre begripa modellens prestanda men kan inte direkt påverka det. När det gäller multiklass klassificeringsmodeller där det förekommer mer än ett sannolikt resultat används det *sparse\_categorical\_crossentropy* (TensorFlow, SparseCategoricalCrossentropy, 2024).

## 2.2 Utvärderingsmetoder

Metoder för att mäta modellens prestanda delas upp i två grupper. *Mean Squared Error (MSE)*, *Mean Absolute Error (MAE)*, *Root Mean Squared Error (RMSE)* är exempel på några utvärderingsmetoder skapade för regressionsmodeller. Nedan har vi metoder som mäter prestanda till klassificeringsmodeller.

### 2.2.1 Accuracy

Vid anpassning av hyperparametrar utför man en *Cross Validation* (korsvalidering) som delar upp data i ett visst antal delar och tränas under olika hyperparametrar. Efter varje träning evalueras modellens prediktionsförmåga i skalan från 0.0 till 1.0 (0% - 100%). Korrekta prediktioner adderas och delas upp med totalt antal prediktioner (Géron, p.89, 2012).

$$Accuracy = \frac{\text{korrekta prediktioner}}{\text{antal prediktioner}}$$

### 2.2.2 Confusion Matrix

För att generera en Confusion Matrix används de prediktionerna igen och jämförs med faktiska värden. Syftet är att få en bild av hur många värden har klassats rätt och hur många har klassats fel. I figuren nedan ser vi en Confusion Matrix där vi har två klasser i target variabeln. När den faktiska värden i variabeln var *NO* blev den rättklassad för 50 gånger och felklassad för 10 gånger.

	Predicted:		
	NO	YES	
Actual: NO	TN = 50	FP = 10	60
Actual: YES	FN = 5	TP = 100	105
55		110	



*TN = True Negatives*

*FN = False Negatives*

*TP = True Positives*

*FP = False Positives*

### 2.2.3 Precision

Precision kallas också för accuracy till alla positiva prediktioner både rätt och fel klassade. Den mäter modellens förmåga att undvika FP prediktioner.

$$Precision = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

### 2.2.4 Recall

Recall fokuserar på de verkliga positiva prediktioner vilket inkluderar fel klassade negatives.

$$Recall = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

### 2.2.5 F1

F1 utvärderingsmetoden är en kombination av Recall och Precision där resultatet kallas för *Harmonic mean value* vilket lägger mer vikt på låga värden. Om både Recall och Precision har en högt värde så kommer F1 vara hög (Géron, p.90–91, 2012).

$$F1 = \frac{2}{\frac{1}{precision} + \frac{1}{recall}} = 2 \times \frac{precision \times recall}{precision + recall} = \frac{2TP}{2TP + FN + FP}$$

## 3 Metod

Följande underrubriker beskriver metoden som används i problemlösandet. Funktionerna som har förklarats under "teoridelen" kommer nu till användning.

### 3.1 Datainsamling

Datasetet "Student Alcohol Consumption" som analyseras i detta projekt kommer från Kaggle och har tidigare använts för att förutspå elevernas beteende och prestanda på gymnasieskolor i förhållande till alkoholkonsumtion (jovi.org, Classification of Alcohol Consumption among Secondary School Students, 2016).

Data innehåller socioekonomisk information om studenter samt deras beteende och prestanda i ämnen portugisiska och matte på två gymnasieskolor i Portugal. Data är uppdelat i två tabeller enligt ämnet de läser. Totalt är det 382 studenter i båda tabeller trots att antal rader sammanlagt blir 1044. Tabellen nedan innehåller alla kolumner och beskrivningar av kolumnernas innehåll

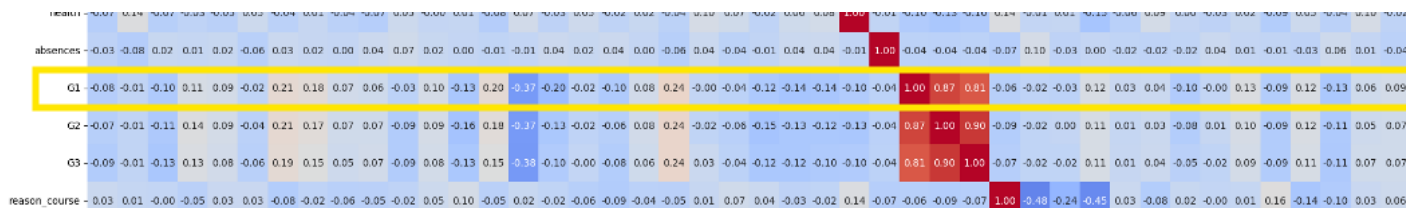
kolumner och deras innehåll
1. school - student's school (binary: 'GP' - Gabriel Pereira or 'MS' - Mousinho da Silveira)
2. sex - student's sex (binary: 'F' - female or 'M' - male)
3. age - student's age (numeric: from 15 to 22)
4. address - student's home address type (binary: 'U' - urban or 'R' - rural)
5. famsize - family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)
6. Pstatus - parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
7. Medu - mother's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
8. Fedu - father's education (numeric: 0 - none, 1 - primary education (4th grade), 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education)
9. Mjob - mother's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
10. Fjob - father's job (nominal: 'teacher', 'health' care related, civil 'services' (e.g. administrative or police), 'at_home' or 'other')
11. reason - reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
12. guardian - student's guardian (nominal: 'mother', 'father' or 'other')
13. traveltime - home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
14. studytime - weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
15. failures - number of past class failures (numeric: n if 1<=n<3, else 4)
16. schoolsup - extra educational support (binary: yes or no)
17. famsup - family educational support (binary: yes or no)
18. paid - extra paid classes within the course subject (Math or Portuguese) (binary: yes or no)
19. activities - extra-curricular activities (binary: yes or no)
20. nursery - attended nursery school (binary: yes or no)
21. higher - wants to take higher education (binary: yes or no)
22. internet - Internet access at home (binary: yes or no)
23. romantic - with a romantic relationship (binary: yes or no)
24. famrel - quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
25. freetime - free time after school (numeric: from 1 - very low to 5 - very high)
26. goout - going out with friends (numeric: from 1 - very low to 5 - very high)
27. dalc - weekday alcohol consumption (numeric: from 1 - very low to 5 - very high)
28. walc - weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
29. health - current health status (numeric: from 1 - very bad to 5 - very good)
30. absences - number of school absences (numeric: from 0 to 93)
<b>Betyg</b>
1. G1 - first period grade (numeric: from 0 to 20)
2. G2 - second period grade (numeric: from 0 to 20)
3. G3 - final grade (numeric: from 0 to 20, output target)

Figur 2

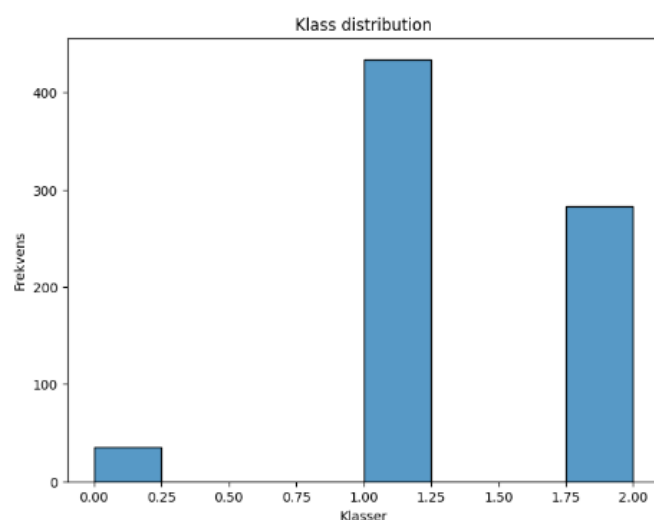
### 3.2 EDA

Utforskande dataanalys (EDA) hjälper att förstå dataset och vart man ska utgå ifrån för att uppnå sitt mål. Det första som behövs göra är att ta inspektera kolumner och värden, antal tomma värden, datatypen, antal dubbla värden och korrelation mellan kolumnerna. På grund av kategoriska kolumner och icke kompatibla datatyper var det nödvändigt att förbehandla data innan det var möjligt att få fram en korrelations matrix. Efter dataförbehandlingen var det möjligt att inspektera

korrelationen mellan kolumner och då var det synligt att många features har ingen påverkan på "G1" som kommer bli en target variabel (se figur 3).



Figur 3. Visar korrelation matrix mellan features och target variabeln G1



Figur 4.

Det var tydligt att den största utmaningen kommer vara ojämnt fördelade klasser i G1, särskilt klass 0 som står för de lägsta betygen vilket representerar studenter som kan behöva undersökas för möjligt stödbehov (Se figur 4).

### 3.3 Förbehandling av data

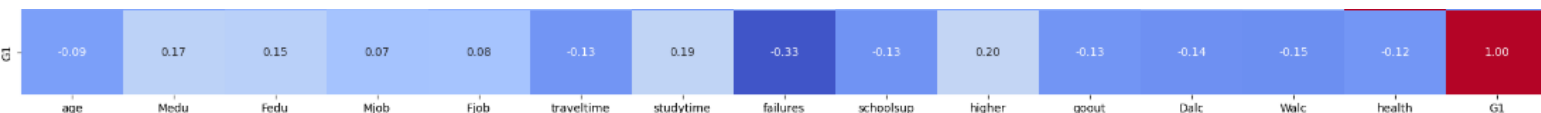
För att få ihop två dataset, 649 och 395 måste båda dataset sammanslås enligt de här kolumnerna ("school", "sex", "age", "address", "famsize", "Pstatus", "Medu", "Fedu", "Mjob", "Fjob", "reason", "nursery", "internet"). Detta kommer resultera till att antal kolumner i nya joinade datasetet kommer öka pga. resten av kolumner som är ämnesrelaterade och har olika värden. Men antal rader förblir samma (382).

Ämnesrelaterade kolumner separeras från datasetet och skapas då en ny dataset som är nu också ämnesrelaterad fast innehåller samma antal studenter som i förra datasetet. I denna del så här vi raderat dubbla rader och nu adderas de igen för att ha en hel dataset med dubbla socioekonomiska information om elever men olika ämnesrelaterade information.

De flesta kolumner är kategoriska förutom betyg och ålder. Antal kategoriska värden varierar men hamnar under 5 – 2 kategorier per kolumn. De med 2 kategorier transformerades med *label*

encoding medan resten som var string data typ gick igenom en *one hot encoding*. Nu var det möjligt att se korrelation mellan kolumnerna i figur 3.

Kolumnen med korrelation under 10% ska tas bort från datasetet för att minska överflödiga information som kan störa modelleringen. Se en slutlig dataset på *figur 5*.



Figur 5. Slutlig version av datasetet och Korrelation till target variabeln

Target variabeln (G1) representerar betyg i skalan från 0–20 så det bästa lösningen skulle vara då att gruppera betygen i 3 jämna klasser (Se figur 6). Ju mindre klasser desto lättare blir det för modellen att upptäcka underliggande mönster och samband i data. Klass 0 ska innehålla betygen från 0–9.99 och den representerar elever med lägsta betyg och ett eventuellt stödbehov.

Sista steget i dataförbehandlingen är att transformera data så att värdena är *machine friendly*. Detta innebär att utföra encoding där det behövs och ändra datatypen till int64.

Scale	Scale Description
20.00	Very Good with Distinction & Honors
18.00 – 19.99	Excellent
16.00 – 17.99	Very Good
14.00 – 15.99	Good
10.00 – 13.99	Sufficient
7.00 – 9.99	Poor (Conditional) / Fail
1.00 – 6.99	Fail

Figur 6. Betygsskalan på gymnasieskolor i Portugal

### 3.4 Modellering

Data ska modelleras med neurala nätverk eller mer specifikt med *Sequential* modell från Tensorflow som passar bäst för enkel tabulär data såsom *Student Alcohol Consumption* dataset i det här fallet.

Första steget är då att importera modellen och alla nödvändiga tekniska verktyg och börja stapla lagar med tanke på att bygga det så enkelt som möjligt för analysera modellens prestanda och förstå vad som ska optimeras.

*Innan data hamnade i nuläget efter sista förbehandlingen så har den redan modellerats flertal gånger och resulterat med dåliga utvärderingar.*

Andra steget i modelleringen inkluderar Keras Tuner för att anpassa modellen till data vid träningen. Hyperparameter tuning delas upp i två delar som nämnt tidigare i rapporten och första delen är då att tuna *modellens hyperparametrar* som anpassar bredd och antal hidden layers. Meningen med den här var rent experimentellt eftersom data var i enkel tabulär form med ca 700 rader innan den splittrades till tränings och testdata. Modellen borde kunna hantera det med få antal lager i sin enklaste form. Antal dolda lager var mellan 1 –5 medan lägsta antal neuroner efter input lager ökade från 32 till 512 med 32 neuroner mer i varje steg.

Därefter så tunas *algoritmen hyperparametrar* som anpassar tränings hastighet och kvalitén av inlärningsalgoritmer. Dropout introducerades nu till träningen och justerades men olika inlärningsgrader mellan 0.1 och 0.5.

Sista delen av hyperparameter tuning var optimeringsalgoritmer som kan analysera modellens förlustfunktion och ge några insikter om modellens prestanda.

Nämnda hyperparametrar optimeras för att förbättra modellens valideringsnoggrannhet (*val\_accuracy*), vid varje träning med 30% av träningsdata. När bästa modellen har valts då tränas den igen och evalueras med testdata. Slutligen utförs det en prediktion och utvärderas modellens prestanda baserat på sin prediktionsförmåga med testdata. Resultaten visualiseras med hjälp av plottar och grafer för en tydligare insikt.

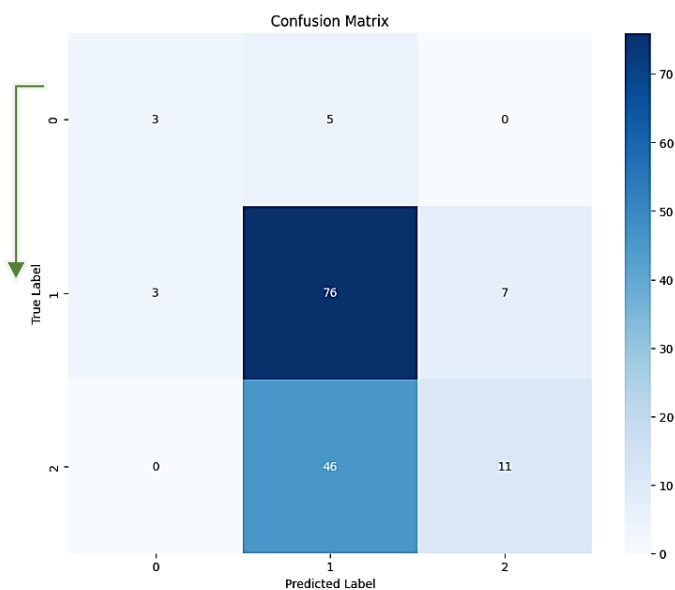
## 4 Resultat och Diskussion

I tabellen nedan ser vi olika modeller och deras prediktionsförmåga med testdata. Modeller som hade 15 klasser i target variabeln hade svårt att klassa ut de korrekt i slutet och därför grupperades klasserna i tre jämna klasser enligt deras unika värden. Det resulterade med tre lika klasser med olika antal exemplar. Klass 0 som är en grupp med lägsta betyg, hade 8 exemplar i testdata och blev alltid felklassad som 1.

Högsta accuracy visade modell 6 dock har den predikerat klassen 0 dåligt enligt F1 utvärderingen (36%). F1 var däremot mycket högre med modell 7 på hela 43% trots en lägre accuracy på alla klasser.

Modeller	Klasser som target	precision	recall	F1 score	Accuracy
Sequential 1 - med skalad data	15				13%
Sequential 2	15				19%
Sequential 5	3	67%	25%	36%	63%
Sequential 6 Keras Tuner	3	67%	25%	36%	66%
Sequential 7 Keras Tuner	3	50%	38%	43%	60%
Sequential 8 Keras Tuner	3	25%	12%	17%	64%

Figur 7: Modeller och prestanda



Figur 9: Confusion Matrix: Modell 7

	precision	recall	f1-score	support
0	0.50	0.38	0.43	8
1	0.60	0.88	0.71	86
2	0.61	0.19	0.29	57
accuracy			0.60	161

Figur 8: Modell 7

Som de såg tidigare, klass 0 hade totalt 8 exemplar i testdata. Modell 7 har predikterat nollan korrekt 3 gånger och fel 5 gånger där nollan misstogs för en etta. En etta misstogs 3 gånger för en nolla (se figur 7).

## 5 Slutsatser

Resultatet var inte tillfredställande för att fortsätta bygga på det här projektet under dessa omständigheter. Av 8 elever med ett stödbehov så 5 av dem har klassats fel. Majoriteten av elever i riskgruppen kan riskera att inte bli upptäckta av modellen.

Det finns många sätt att få en hög resultat men sättet som valdes för detta projekt har många begränsningar. Något som kunde provas fram är träna andra modeller såsom Logistisk Regression, Random Forest Klassificerare eller SVM. Kanske en mer fördjupad feature engineering hade påverkat modellens prestanda men det hade inte löst den största utmaningen och det är ha en target variabeln med en tydlig och stor klassobalans. Detta problem kunde lösas bäst med data augmentation vilket skulle innebära att man artificiellt skapar flera exemplar för den underrepresenterade klassen.

För att svara på frågeställningen, det hade kunnat vara möjligt att upptäcka stödbehov bland elever baserat på deras socioekonomiska information, beteende och vanor om det fanns tillräckligt med data. Mer robust data och andra metoder kunde ge ett tillfredställande resultat men trots detta så gav projektet en tydligt perspektiv på användningen av neurala nätverk för att lösa en sådant problem. När det gäller andra frågeställningen så har vi inte uppnått ett f1 utvärdering på mer än 50% och med det kan vi säga att, med tekniken som användes, så är det inte möjligt.

### 5.1 Möjlighet till vidareutveckling av projektet

Detta projekt kan vara en hård grund till vidareutveckling av en programvara som kan ge insikter på en ny generation studenter och se till att de bemöts med korrekta resurser och stöd för att optimera deras lärande och välbefinnande.

Det kan vara mycket användbar för skolor och pedagoger genom att ge dem verktyg för att identifiera elever som kan behöva extra stöd och intervention för att förhindra underprestation och främja deras akademiska framgång. I stället för att identifiera enskilda elever som kan behöva stöd så kan det vara lika bra att få en statistisk rapport såsom en dashboard.

## Källförteckning

Geoffrey E. Hinton, Nitish Srivastava, Alex Krizhevsky, Ilya Sutskever, Ruslan R. Salakhutdinov, (2012). *Improving Neural Networks by Preventing Co-Adaptation of Feature Detectors*, Hämtat från: <https://arxiv.org/abs/1207.0580>

Géron, A. (2019). *“Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.”*

Joiv, joiv.org, (2016). *Classification of Alcohol Consumption among Secondary School Students*. Hämtat från: <https://joiv.org/index.php/joiv/article/view/64>

Kaggle, (2019). *Student Alcohol Consumption*. Hämtat från: <https://www.kaggle.com/datasets/uciml/student-alcohol-consumption/discussion/72551>

Tensorflow, (Senast uppdaterad april 2024). *SparseCategoricalCrossentropy*. Hämtat från: [https://www.tensorflow.org/api\\_docs/python/tf/keras/losses/SparseCategoricalCrossentropy#get\\_config](https://www.tensorflow.org/api_docs/python/tf/keras/losses/SparseCategoricalCrossentropy#get_config)

Skolinspektionen, (2021). *Arbetet med individuella stödinsatser i gymnasieskolan* (p.4, sammanfattning), Hämtat från: <https://www.skolinspektionen.se/beslut-rapporter-statistik/publikationer/kvalitetsgranskning/2021/individuella-stodinsatser-i-gymnasieskolan/>

Skolverket, (Senast uppdaterad januari 2024). *Rätt att få hjälp i skolan*. Webbida skolverket.se. Hämtat från: <https://www.skolverket.se/for-dig-som-.../elev-eller-foralder/smanavigation-elevers-rattigheter/ratt-att-fa-hjalp-i-skolan>

Tensorflow, (2024), *Introduction to the Keras Tuner*, Hämtat från: [https://www.tensorflow.org/tutorials/keras/keras\\_tuner](https://www.tensorflow.org/tutorials/keras/keras_tuner)

Yugesh Verma, (Senast uppdaterad april 2024). *Dense Layers*. Hämtat från: <https://analyticsindiamag.com/topics/what-is-dense-layer-in-neural-network/>