Eurovision Song Contest

Prediktion av kvalificerade länder

Marcus Eklund

EC Utbildning

Projektarbete – Data Science

202410

Abstract

This project focuses on developing a machine learning model to predict whether a contestant in the Eurovision Song Contest (ESC) semi-final will advance to the final competition. Using a dataset from Kaggle and complementing data from Musicstax, we used cleaning and preprocessing methods to prepare the data for training. We built five models: Decision Tree, Decision Tree with fine-tuning, Logistic Regression, Random Forest, and Neural Network, each with and without random oversampling. Among these, the Random Forest model with random oversampling achieved the highest accuracy, correctly predicting outcomes 77% of the time.

Innehållsförteckning

1 Inledning

Eurovision Song Contest anordnades för första gången 1956 i Schweiz, då endast sju nationer deltog och de som inte kunde vara på plats följde tävlingen via radio. Idag, över 60 år senare, har Eurovision vuxit till ett globalt fenomen som lockar miljontals tittare världen över. Under ESC 2024, som hölls i Malmö, deltog hela 37 länder i tävlingen.

Eurovision har blivit en årlig tradition som många följer tillsammans med vänner och familj, där tävlingsbidragen ofta framkallar starka reaktioner — både positiva och negativa. Detta skapar ett naturligt engagemang och spekulationer om hur varje land kommer att placera sig. Betting på Eurovision är också populärt, och flera bettingsajter erbjuder olika sätt att satsa på tävlingen. De vanligaste insatserna rör slutplaceringar och chansen att ett bidrag går vidare till finalen. Många tar betting på stort allvar och lägger tid på att analysera de olika bidragen för att försöka förutsäga resultatet. (eurovisionworld)

Med hjälp av maskininlärningsmodeller kan vi nu ta ett systematiskt grepp om dessa förutsägelser. Genom att samla och mata in olika relevanta variabler kan modellen göra en prediktion baserat på dessa faktorer. Frågan som uppstår är: vilka faktorer är avgörande för att ett bidrag ska kvalificera sig till finalen? Och hur kan vi förbättra träffsäkerheten i våra förutsägelser?

Denna fråga är intressant inte bara för dem som planerar att satsa pengar, utan också för de som tävlingsinriktat vill briljera i vänkretsen under Eurovision-kvällen.

Syftet med denna rapport är att utveckla olika ML-modeller för att förutse om en låt i ESC kommer att gå vidare från semifinal till final. Följande frågeställning kommer att besvaras:

1. Går det att förutse vilka låtar som kvalificerar sig till ESC-finalen med hjälp av maskininlärning?

2 Teori

För att besvara frågeställningen och uppnå syftet med rapporten behöver vi transformera data och implementera olika modeller som kan hjälpa oss att prediktera ESC kvalificeringar. I följande avsnitt ges en överblick över de metoder och modeller som valts ut för ändamålet.

2.1 Transformeringsmetoder

De flesta dataseten som man stöter på eller bygger upp består av både numerisk och kategorisk data men när en maskininlärningsmodell ska tränas så kan dem flesta enbart hantera numerisk data. I hanteringen av kategorisk data så behöver en transformering ske som till exempel One Hot Encoding och Label Encoding.

2.1.1 Label Encoding

Label Encoding är en metod där kategoriska värden omvandlas till numeriska värden genom att varje kategori får ett unikt heltalsvärde. Det är särskilt användbart när kategorierna har en naturlig ordning, till exempel storlekar (liten, medium, stor) eller frekvenser. Det kan dock orsaka problem om ingen sådan ordning finns, eftersom vissa algoritmer kan tolka de tilldelade siffrorna som en rangordning. (scikit-learn)

2.1.2 One Hot Encoding

One Hot Encoding skapar en separat, binär kolumn för varje kategori. Om det till exempel finns tre kategorier: ”Röd”, ”Blå”, ”Grön” kommer One Hot Encoding att skapa tre kolumner, där varje rad får ett värde på 1 i kolumnen för sin specifika färg och 0 i de andra kolumnerna. Den här metoden används ofta för kategorier utan någon inbördes ordning, vilket minskar risken för att algoritmer misstolkar kategorierna som rangordnade. (scikit-learn)

2.2 Random Oversampling

Random Oversampling är en metod för att jämna ut en obalanserad variabel genom att slumpmässigt duplicera värden från minoritetsklassen så att den matchar antalet i majoritetsklassen. Denna metod kan leda till överanpassning då modellen kan börja komma ihåg dem duplicerade värdena. Den kan dock vara väldigt användbar när datasetet är för litet. (Wisam, E. (2023))

2.3 Klassificeringsmodeller

Klassificeringsmodeller är en grupp av modeller som används för att lösa problem där datapunkter ska delas in i specifika kategorier. Dessa modeller kan skilja sig markant i sin design och är olika lämpade för olika typer av klassificeringsproblem.

2.3.1 Decision Tree

En Decision Tree-modell får sitt namn från dess trädliknande struktur, där varje grening motsvarar en fråga relaterad till en variabel i datan. Vid varje förgrening delas datapunkterna upp i undergrupper beroende på svaren på dessa frågor, vilket fortsätter tills en slutpunkt eller "blad" i trädet nås. Dessa blad representerar slutgiltiga beslut eller kategorier. Decision Tree är både kraftfullt och lätt att tolka, särskilt inom klassificerings- och regressionsproblem, men riskerar att överanpassa data om trädet blir för komplext och inte beskärs. (Géron, A. (2019))

2.3.2 Logistisk Regression

Trots namnet är logistisk regression en klassificeringsmodell och är särskilt användbar för problem med binära utfall, exempelvis att förutsäga om ett objekt hör till en viss kategori eller inte. Modellen utnyttjar en logistisk funktion som beräknar sannolikheten att en observation tillhör en av två klasser, där utfallen oftast representeras som 0 eller 1. Genom sigmoid-funktionen omvandlas en linjär kombination av indata till ett sannolikhetsestimat mellan 0 och 1, vilket är grundläggande i modellen. Logistisk regression passar särskilt bra när ett linjärt samband finns mellan de oberoende variablerna och log-odds för utfallet. (Géron, A. (2019))

2.3.3 Random Forest

Random Forest är en ensemble-modell som bygger på en samling av decision trees. Genom att kombinera prediktionerna från många träd ökar modellen noggrannheten, där majoriteten av trädens prediktioner avgör den slutliga klassificeringen. Trots sin enkelhet är random forest en av de mest kraftfulla modellerna för maskininlärning idag och presterar ofta mycket väl på olika typer av klassificeringsproblem. (Géron, A. (2019))

2.4 Neurala Nätverk

Neurala nätverk är inspirerade av hur hjärnan fungerar och består av sammankopplade enheter som kallas neuroner, organiserade i lager. Dessa nätverk är extremt kraftfulla och utmärker sig vid att identifiera komplexa mönster i stora datamängder.

2.4.1 Feedforward Neural Network

Ett Feedforward Neural Network (FNN) är en typ av neuralt nätverk där information endast flödar framåt, från inmatningslagret genom ett eller flera dolda lager till utmatningslagret, utan återkopplingar eller slingor. Alla neuroner i ett lager är kopplade till neuronerna i nästa lager, vilket gör FNN till ett "fullständigt anslutet" nätverk. Modellen används ofta för klassificerings- och regressionsuppgifter och tränas genom att justera vikterna mellan neuronerna för att minimera felmarginalen i dess förutsägelser. (Géron, A. (2019))

2.5 Bedömningsmetriker

För att bedöma prestandan hos maskininlärningsmodeller behövs pålitliga och effektiva mätvärden som kan ge en snabb och precis bild av modellens resultat. Två av dem mest användbara är Accuracy score och F1-score.

2.5.1 Accuracy score

Accuracy score anger hur stor andel av modellens prediktioner som är korrekta, där värdet varierar mellan 0 och 1. Om modellen till exempel gör 100 prediktioner och 80 av dessa är rätt, får den en accuracy score på 0,8. Detta mätvärde är användbart för att snabbt uppskatta modellens prestanda. Dock är det viktigt att vara uppmärksam på risken för missvisande höga accuracy-värden när data är obalanserad, det vill säga när ett utfall dominerar. I sådana fall kan modellen uppnå hög accuracy genom att i stor utsträckning förutsäga det vanligaste utfallet, även om den missar andra kategorier. (Géron, A. (2019))

2.5.2 F1-score

För att komplettera accuracy, och för att säkerställa att modellen även presterar bra på mindre vanliga utfall, används ofta F1-score. F1-score balanserar modellens recall (förmåga att identifiera alla faktiska positiva fall) och precision (exakthet i att endast identifiera positiva fall). Den matematiska beräkningen av F1-score kombinerar precision och recall till ett harmoniskt medelvärde, vilket gör den särskilt användbar vid obalanserade dataset. (Géron, A. (2019))

3 Metod

3.1 Datainsamling

Vi laddade ner ett dataset från Kaggle som innehåller information om Eurovision för åren 2009–2023. Under den explorativa dataanalysen (EDA) upptäckte vi att ca 80 låtar saknade vissa värden. Av dessa kunde vi manuellt komplettera data för nästan 60 låtar via Musicstax.com, medan de resterande togs bort från datasetet.

3.2 EDA

För att få en smidig och heltäckande överblick av vårt dataset använde vi Python-paketet ydata_profiling. Paketet innehåller funktionen ProfileReport, som genomför en EDA på en pandas DataFrame. EDAn visade att datasetet omfattade totalt 565 låtar och 44 variabler, med 2,448 celler där värden saknades.

3.3 Rensning och Transformering

Vi inledde med att välja ut vilka variabler som vi ansåg var relevanta baserat på vår ProfileReport. Vi exkluderade variabler som saknade värden för alla låtar samt sådana som endast var kopplade till finalen, eftersom de inte var relevanta för vår analys av semifinalerna. Dessutom togs vissa variabler bort på grund av en extremt ojämn fördelning av värden. Efter selektionen återstod totalt 17 kolumner och 438 låtar. Flera av dessa kolumner representerade kategoriska värden som behövde omvandlas med hjälp av One Hot Encoding eller Label Encoding. Vi valde att separera data för låtarna från 2023 för att använda den i våra prediktioner med de färdiga modellerna. För att få en överblick över datan efter selektionen och transformationen använde vi återigen ProfileReport, som nu visade att vi hade 62 variabler och 438 låtar.

3.4 Modellträning

Vi delade upp datasetet i tränings- och test set. Vid analys av vår målvariabel upptäckte vi en viss obalans. För att undersöka om detta skulle påverka modellernas prestanda skapade vi två versioner av träningsdatan: en obalanserad och en balanserad med Random Oversampling. Vi testade flera olika modeller som vi ansåg lämpade för vårt syfte:

- Decision Tree Classifier
- Logistic Regression
- Random Forest Classifier
- Neural Network

För att smidigt kunna köra och jämföra resultaten från de olika modellerna utvecklade vi funktioner som konsekvent visar modellernas prestanda. Vi skapade även en funktion för att tillämpa modellerna på 2023 års bidrag och få fram prediktioner.

Vi gjorde även ett stapeldiagram för att visa träningsvariablerna och rangordna dem enligt hur betydande dem var på modellens prestanda.

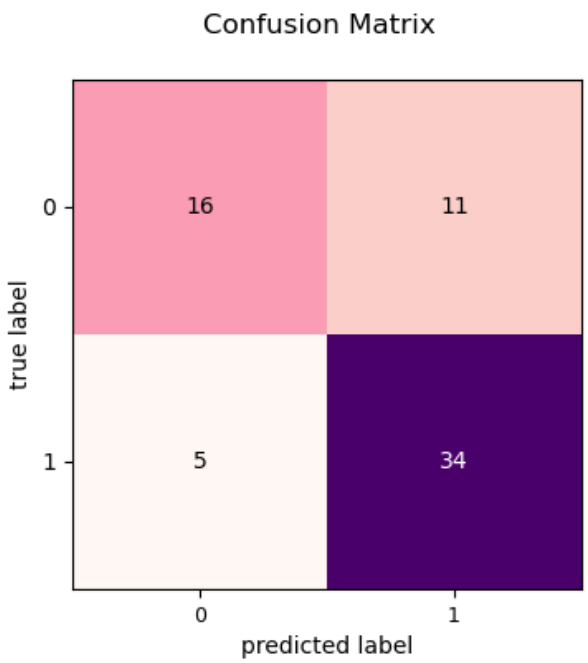
4 Resultat och Diskussion

4.1 Resultat

I denna sektion presenteras resultaten från de fem maskininlärningsmodellerna (Decision Tree, Decision Tree med Fine-Tuning, Logistic Regression, Random Forest, och Neural Network) på både det balanserade och obalanserade datasetet. Vi analyserar också vilka variabler som påverkade prestandan i den mest effektiva modellen.

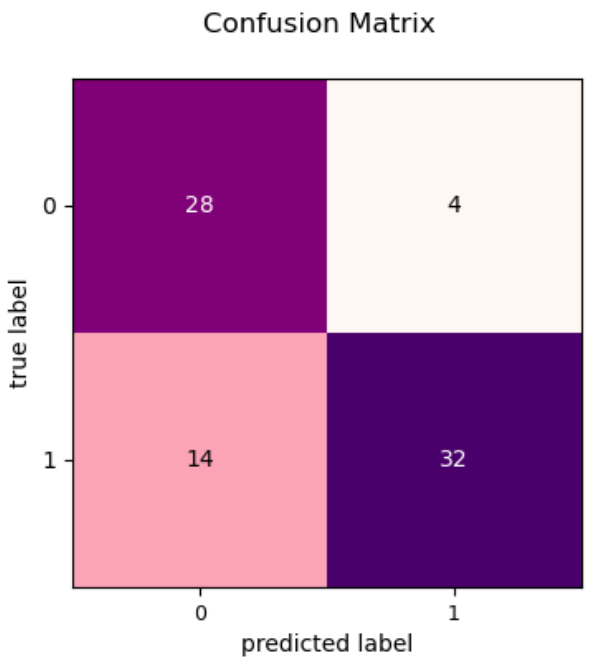
4.1.1 Decision Tree

Decision Tree				
Dataset	**Accuracy**	**Precision**	**Recall**	**F1-score**
Obalanserad	0,757576	0,755556	0,871795	0,809524
Random Oversampling	0,769231	0,888889	0,695652	0,780488



Figur 1: Confusion Matrix för Decision Tree på obalanserad data

14 av 20 korrekt för 2023 prediktionen.

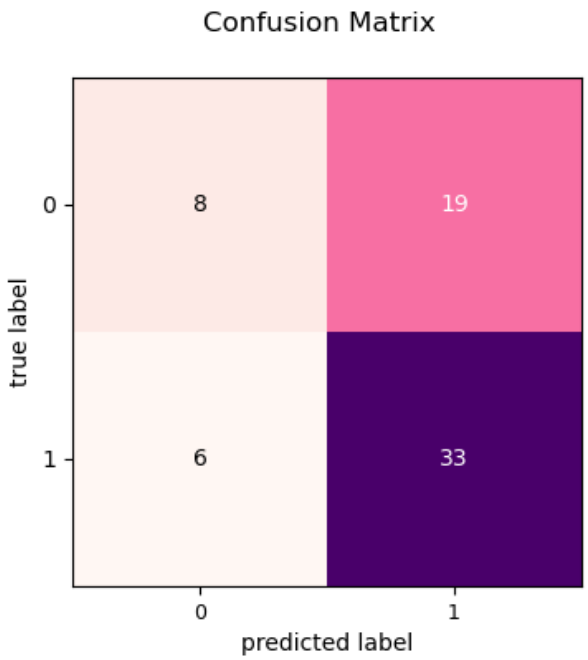


Figur 2: Confusion Matrix för Decision Tree med Random Oversampling

14 av 20 korrekt för 2023 prediktionen.

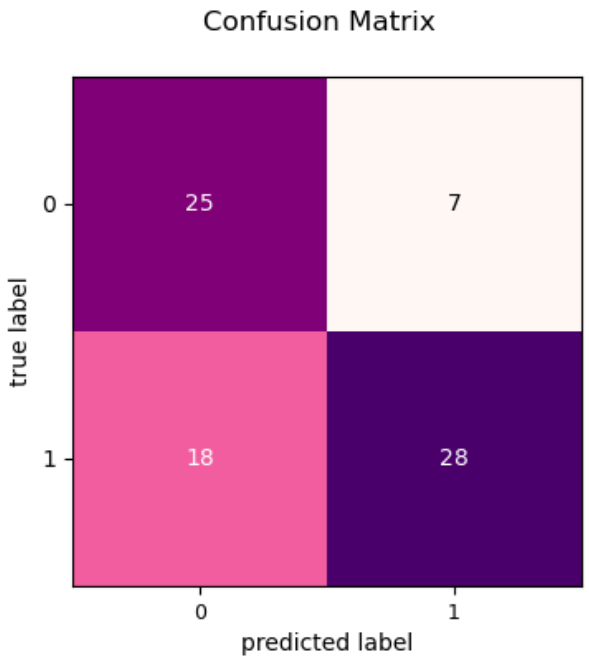
4.1.2 Decision Tree med Fine Tuning

Decision Tree med Fine Tuning				
Dataset	**Accuracy**	**Precision**	**Recall**	**F1-score**
Obalanserad	0,621212	0,634615	0,846154	0,725275
Random Oversampling	0,679487	0,800000	0,608696	0,691358



Figur 3: Confusion Matrix för Decision Tree med fine tuning på obalanserad data

13 av 20 korrekt för 2023 prediktionen.

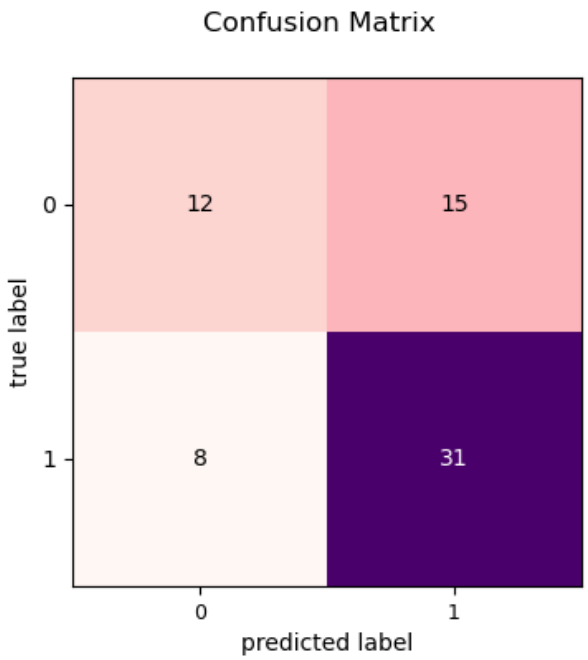


Figur 4: Confusion Matrix för Decision Tree med fine tuning med Random Oversampling

12 av 20 korrekt för 2023 prediktionen.

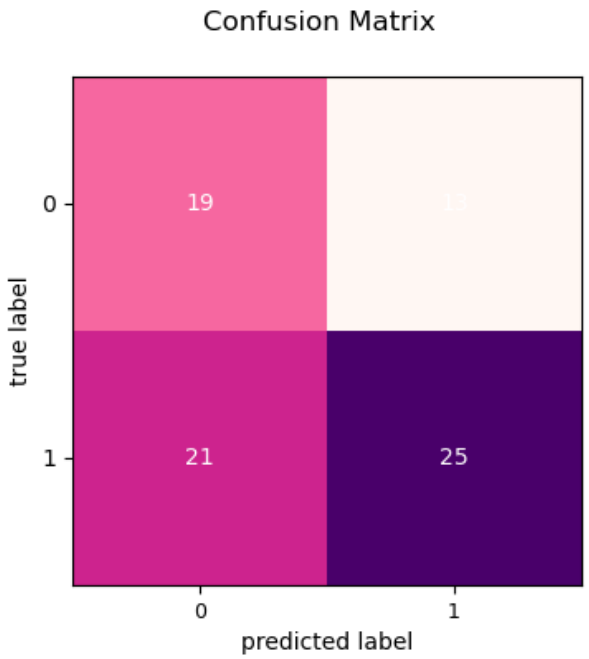
4.1.3 Logistic Regression

Logistic Regression				
Dataset	**Accuracy**	**Precision**	**Recall**	**F1-score**
Obalanserad	0,651515	0,673913	0,794872	0,729412
Random Oversampling	0,564103	0,657895	0,543478	0,595238



Figur 5: Confusion Matrix för Logistic Regression på obalanserad data

16 av 20 korrekt för 2023 prediktionen.

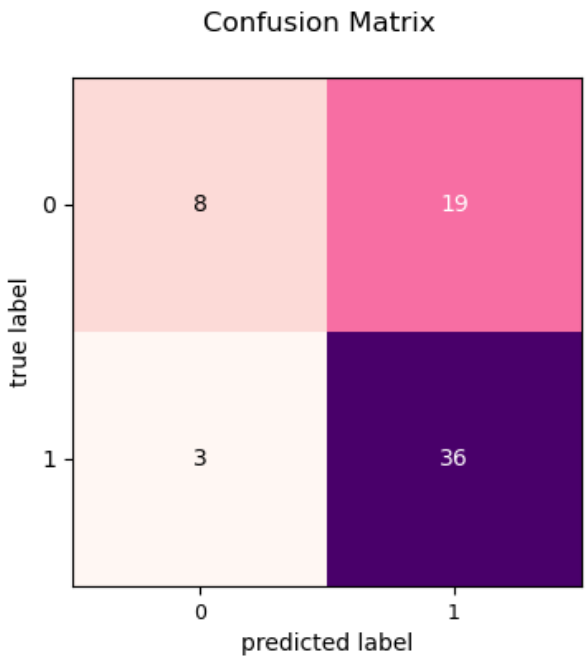


Figur 6: Confusion Matrix för Logistic Regression med Random Oversampling

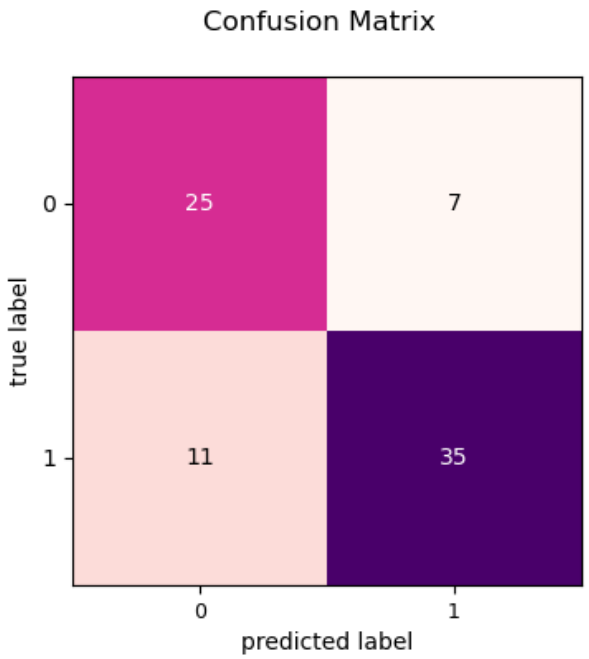
14 av 20 korrekt för 2023 prediktionen.

4.1.4 Random Forest

Random Forest				
Dataset	**Accuracy**	**Precision**	**Recall**	**F1-score**
Obalanserad	0,666667	0,654545	0,923077	0,765957
Random Oversampling	0,769231	0,833333	0,760870	0,795455



Figur 7: Confusion Matrix för Random Forest på obalanserad data



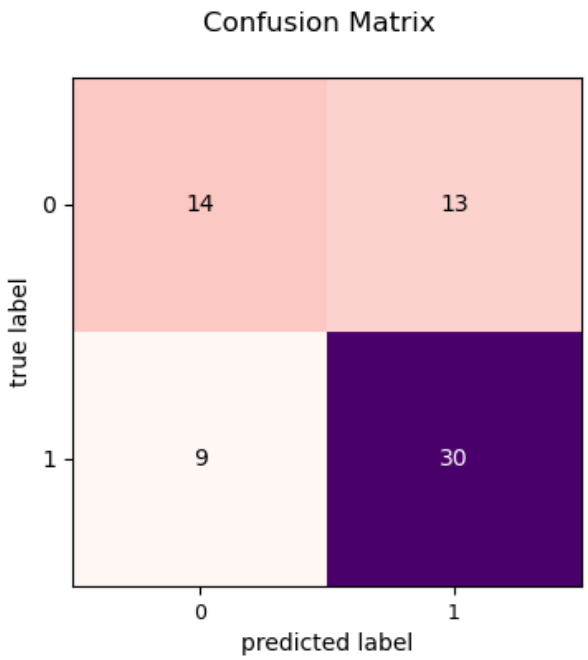
Figur 8: Confusion Matrix för Random Forest med Random Oversampling

12 av 20 korrekt för 2023 prediktionen.

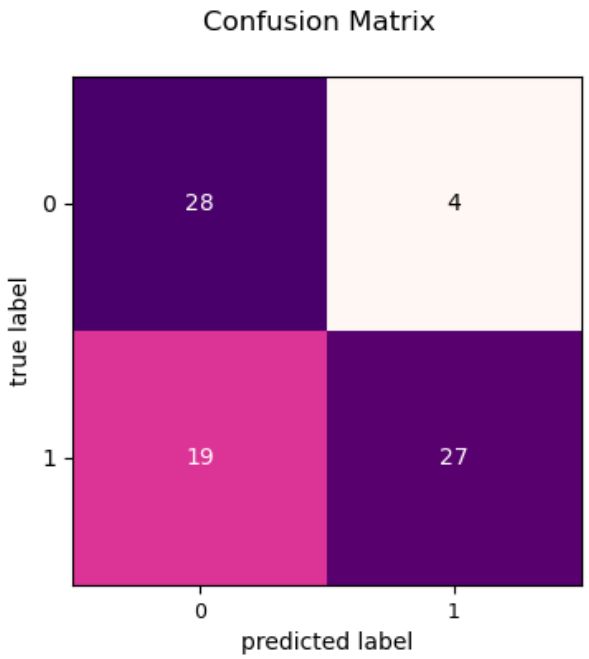
15 av 20 korrekt för 2023 prediktionen.

4.1.5 Neural Network

Neural Network				
Dataset	**Accuracy**	**Precision**	**Recall**	**F1-score**
Obalanserad	0,666667	0,697674	0,769231	0,731707
Random Oversampling	0,705128	0,870968	0,586957	0,701299



Figur 9: Confusion Matrix för Neural Network på obalanserad data

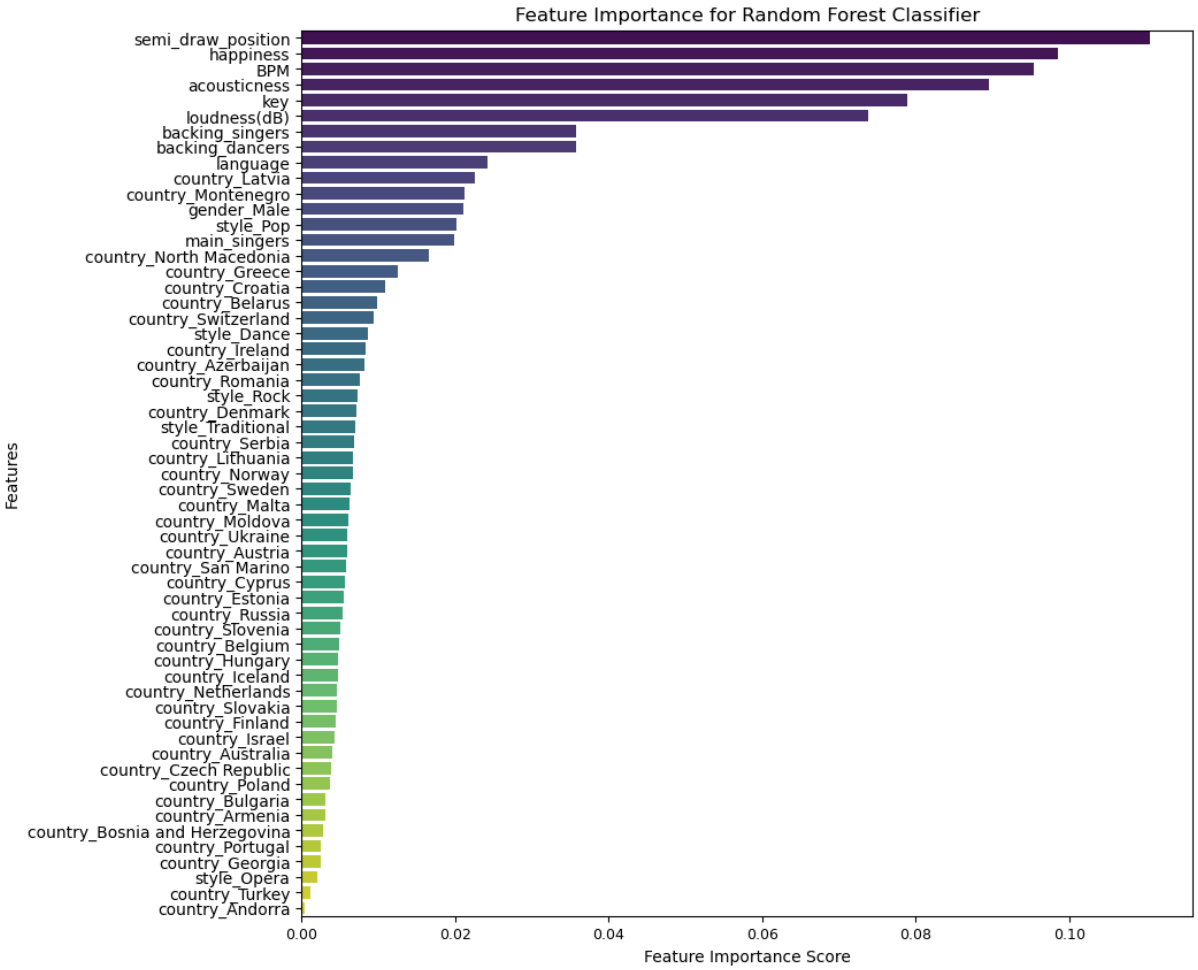


Figur 10: Confusion Matrix för Neural Network med Random Oversampling

14 av 20 korrekt för 2023 prediktionen.

14 av 20 korrekt för 2023 prediktionen.

4.1.6 Variabelbetydelse



Figur 11: Variabelbetydelse för Random Forest rangordnat från störst till minst betydelse

4.2 Diskussion

4.2.1 Påverkan av obalans i målvariabeln

Resultaten visar att balanseringen av målvariabeln hade en begränsad påverkan. I vissa fall förbättrades resultaten, medan de i andra fall försämrades något. Detta var förväntat, då vår målvariabel inte var särskilt obalanserad. Hade skillnaden varit större, skulle balansering sannolikt ha haft en mer positiv effekt.

4.2.2 Överanpassning hos Decision Tree

I Decision Tree-modellen uppnås relativt höga poäng under träningen, men vid prediktion ger modellen enbart sannolikheterna 1 eller 0. Detta indikerar att modellen är överanpassad och har svårt att generalisera till ny data. För att hantera detta testade vi fine-tuning med GridSearchCV, vilket resulterade i mer representativa träningsresultat. Samtidigt visade det att Decision Tree inte var den mest lämpliga modellen för vårt syfte.

4.2.3 Variabelbetydelse

Vi kan se att ingen av egenskaperna överstiger 0,12, vilket gör det imponerande att modellen kan göra så pass bra förutsägelser. Semi_draw_position har den största påverkan, men skiljer sig inte så mycket från de andra egenskaperna.

4.2.4 Varför Random Forest presterar bäst

Random Forest överträffar de andra modellerna tack vare sin ensemble-struktur, där flera beslutsstödträd kombineras för att skapa en robust och träffsäker modell. Genom att använda slumpmässiga delmängder av både data och funktioner minskar Random Forest risken för överanpassning och är mindre känslig för brus och extrema värden. På balanserade dataset uppnår modellen dessutom en bättre balans mellan precision och recall, vilket ger högre F1-score.

I kontrast lider Decision Tree av överanpassning, medan Logistic Regression förutsätter linjära samband, vilket begränsar dess förmåga att fånga komplexa mönster. Neural Network kan hantera komplexa samband men kräver stora datamängder för effektiv träning, vilket gör den mindre lämplig för vårt lilla dataset.

5 Slutsatser

Vår ursprungliga frågeställning var om vi med hjälp av maskininlärning kunde förutsäga vilka låtar i Eurovision som skulle gå vidare från semifinalerna till finalen. Vi var medvetna om hur komplex den frågan är, där faktorer som publikens subjektiva åsikter om både låtar och länder spelar stor roll. Trots detta ville vi undersöka om förutsägelser kunde göras utan att ta hänsyn till opinionsmätningar eller geopolitiska faktorer.

Efter att ha testat flera modeller blev vi positivt överraskade av resultatet från **Random Forest Classifier med Random Oversampling** som uppnådde en accuracy på **0,769** och en F1-score på **0,795**, och korrekt förutspådde 15 av de 20 länder som kvalificerade sig till finalen 2023. Våra tester visade att balansering av målvariabeln med Random Oversampling hade liten inverkan på modellernas prestanda; vissa resultat förbättrades marginellt, medan andra blev något sämre, men de totala prediktionerna var i stort sett likvärdiga. Även om den ursprungliga Decision Tree-modellen gav relativt höga poäng, visade den tydliga tecken på överanpassning. Logistic Regression utan balansering förutspådde korrekt 16 av 20 länder, men tenderade också att klassificera flera icke-kvalificerande låtar som kvalificerade, vilket gav den en lägre accuracy totalt sett jämfört med Random Forest.

Framtida arbete

Detta projekt kan utvecklas vidare för att förutsäga framtida års tävlingar och förbättra modellerna i takt med att mer data blir tillgänglig för träning. Våra nästa mål är att testa modellerna på 2024 års tävling och sedan på 2025 när de nya bidragen släpps.

Källförteckning

Eurovisionworld https://eurovisionworld.com/odds/eurovision-semi-final-1

Géron, A. (2019). Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow. (2. uppl.) Sebastopol: O'Reilly Media, Inc.

Kaggle https://www.kaggle.com/datasets/diamondsnake/eurovision-song-contest-data/data

Musicstax https://musicstax.com/

scikit-learn https://scikit-learn.org/dev/modules/preprocessing.html#preprocessing-categorical-features

Wisam, E. (2023). *Class Imbalance and Resampling: A Formal Introduction* https://towardsdatascience.com/class-imbalance-and-oversampling-a-formal-introduction-c77b918e586d