PROJECT E12: ESC

Eurovision - odds vs reality

Lauri Lüüsi, Marek Murumäe

https://github.com/MarekMurumae/Eurovision-Betting-Odds

Business understanding

Identifying our business goals

Background

We are avid viewers of the Eurovision Song Contest (ESC) with an interest in betting who want to be able to win more consistently with our bets. It can be observed that the results of ESC are predictable through some features and by applying knowledge learned in this course, we can make more consistent and profitable bets. We believe the main feature that helps predict the performance of a song is the country of origin. Therefore we can make more profitable bets by finding which countries perform well or badly.

Business goals

- **Goal 1:** Attempt to roughly predict the 2023 and future contest's results. (e.g. winner, who will qualify, etc.)
- **Goal 2:** Find if and what features (e.g. language, running order, etc.) affect the final position of a song positively or negatively, based on which it would be possible to predict how well a country will perform (so that we could make more confident and safe bets)
- **Goal 3:** Find which countries perform consistently well/badly in odds and well/badly in reality.

Business success criteria

We will consider our project a success:

- if our model's predictions are more accurate than random guessing
- if we successfully identify features (e.g. running order, language) that determine the final position of a song consistently
- if we find a countries that consistently perform well or badly in actual results and/or betting odds

Assessing your situation

Inventory of resources:

- Us as in the duo working on this project as novice data scientists
- Jupyter notebook for manipulating and mining the data
- Github for organizing the project
- Our datasets (can be found here)
- Our computers provided to us by Tartu University

Requirements, assumptions, and constraints

- The project deadline is Monday, 12th of December, 2022, at noon (12:00).
- The developed model must be able to use the features of a given entry to predict the results of the contest for said entry.
- The dataset must not contain any wrong data.

Risks and contingencies

- Our internet connection
- Our team members time management skills
- Perhaps lacking enough specific knowledge in data science / Python or its libraries we will be using
- Any other difficulties that might arise unexpectedly

Terminology

- ESC - Eurovision Song Contest
- Entry - a song entered into ESC by a country.
- Bet - a wager made on a particular prediction.
- Running order - the order in which countries perform in the contest

Costs and benefits

Our costs are our time spent working on the project and the money lost making potential unprofitable bets. Our benefits will be the winnings made from our placed bets in the upcoming ESCs.

Defining our data-mining goals

Data-mining goals

Making accurate models to predict the results of an entry along with an accuracy report for said model.

Data-mining success criteria

Our model correctly predicts on average 7 out of the top 10 songs in ESC and accurately predicts the winner at least 51% of the times.

Data understanding

Gathering Data

Data requirements:

Each entry before 2023 should contain at least the following features:

- Country they are performing for
- Title of the song
- Language that the song was performed in

- Place and points in the final
- Did the entry qualify for the final or not
- Running order in the semi-final and the final

Data availability

The required data both exists and is available for usage, although some songs (lyrics, music, etc) might be copyrighted and can't be made publicly available.

Selection Criteria

- Information about countries positions, points, running order, etc. gathered manually from eurovisionworld.com
- Data about ESC songs collected privately by lab supervisor.

Describing data

The main dataset for the project is gathered from eurovisionworld.com which contains the following features: **year**, **country**, **place**, **pts** (points), **tele** (i.e. points from the televote), **jury** (i.e. points from the jury), **qualified** (binary feature that shows if the country qualified to the final or not), **odds_final** (odds in the final), **odds_qualify** (odds that the entry would qualify to the final), **semi_final** (which semi-final the country performed in, first or second), **final_ro** (running order in final), **semi_ro** (running order in semi final), **artist**, **song**, **language**. We will most likely extract more features from features that are not numeric (e.g. artist, song, language) by performing feature engineering.

The second dataset is provided by lab supervisor. The features we use from there are **love_mentions**, **country_population**, **lyrics_compression** and any others we might find useful.

Exploring data

There are no visible errors in the data, for example no contestant has gotten a position that is worse than the number of participants and nobody has gotten a 0 or a negative number as their place. The missing info from 2020 is also not an error but due to the fact that in 2020 the competition was cancelled. The general amount of points from 2016 onwards has seemingly doubled from prior contests, because the contest changed the voting system of the contest.

Also, if the country didn't qualify for the final, their final running order is 0.

Data quality

The data we have should be sufficient in amount for our goals however we can not be sure of this before we have completed our project goals. As mentioned previously there is data missing for 2020 however this should not be a problem for building our models. Perhaps the biggest problem is the small amount of data (e.g. no betting odds before 2015 and the song information not yet released for 2023) which might result in faulty predictions which we can't really do much about.

Project plan

Task	**Hours**	**Team member**
Data preparation Merge the datasets together, find useful features, feature engineer categorical data into numeric, etc.	**8**	**Lauri**
Split data into training, test and validation	**4**	**Marek**
Train machine learning models with different algorithms For example: Random Forest, KNN, SVM, etc.	**5**	**Lauri, Marek**
Test the accuracy of trained models	**4**	**Lauri, Marek**
Test the most accurate model on the validation data	**4**	**Marek, Lauri**
Make a poster for the project	**4**	**Marek, Lauri**

List of methods

For the project we will use machine learning methods to train our models using Python and its libraries (numpy, pandas, scikit-learn and many more).
Machine learning methods: KNN, SVM, Random Forest and potentially others.