Business Intelligence

Wine Review Dataset

Group:

Gabriel Fernandes	2018288117	gabrielf@student.dei.uc.pt
Miguel Rabuge	2018293728	rabuge@student.dei.uc.pt
Pedro Rodrigues	2018283166	pedror@student.dei.uc.pt

Requirements & Goals

- Our goal is to understand what makes reviewers endorse a certain wine bottle
  - The reviewers' opinions may have a very positive or very negative impact on the wine sales, hence making this analysis suitable, from a business standpoint.

- For that, we have enumerated a list of questions that we are looking forward to see answered:

  - **What makes a wine bottle have good or bad rating?**
    - ➢ How are the wineries related to the reviewer's rating?
    - ➢ How is the bottle price related to the rating?
    - ➢ How is the wine's category (red, white, etc) related to the rating?
    - ➢ How is the wine's alcohol level related to the rating?

  - **Other relevant questions:**
    - ➢ What is the rating count by wine category?
    - ➢ What is the average price by region?
    - ➢ What is the average rating by reviewer?

Dataset

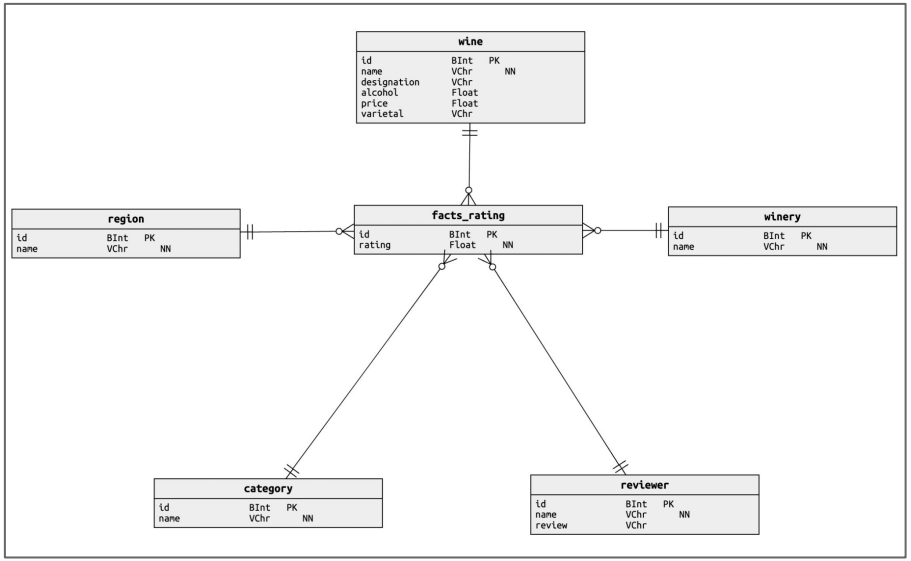
- The characteristics of the dataset chosen for this project is described below:
  - Entries: 320k
  - Features: 10
  - Scraped from www.winemag.com
  - Contains Metadata information about the data collection and cleansing process and code

- The features are mainly wine characteristics, geographical data and the reviewer rating and identification.

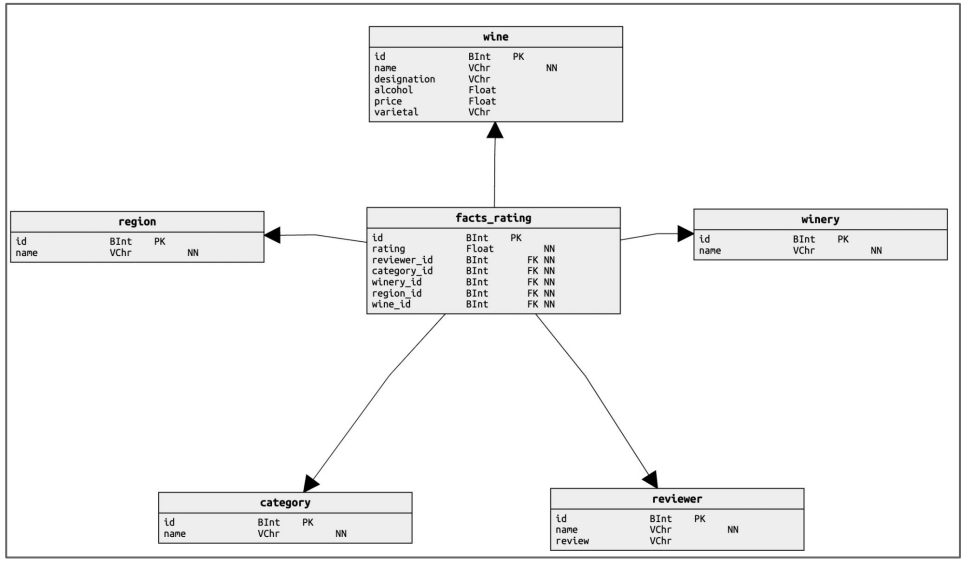
wine	winery	category	designation	varietal	appellation	alcohol	price	rating	reviewer
Las Positas 2011 Estate Barbera (Livermore Valley)	Las Positas	Red	Estate	Barbera	Livermore Valley, Central Coast, California, US	15.1%	$40	89	Virginie Boone

https://www.kaggle.com/samuelmcguire/wine-reviews-data

Data Model - Onda Database Modelling

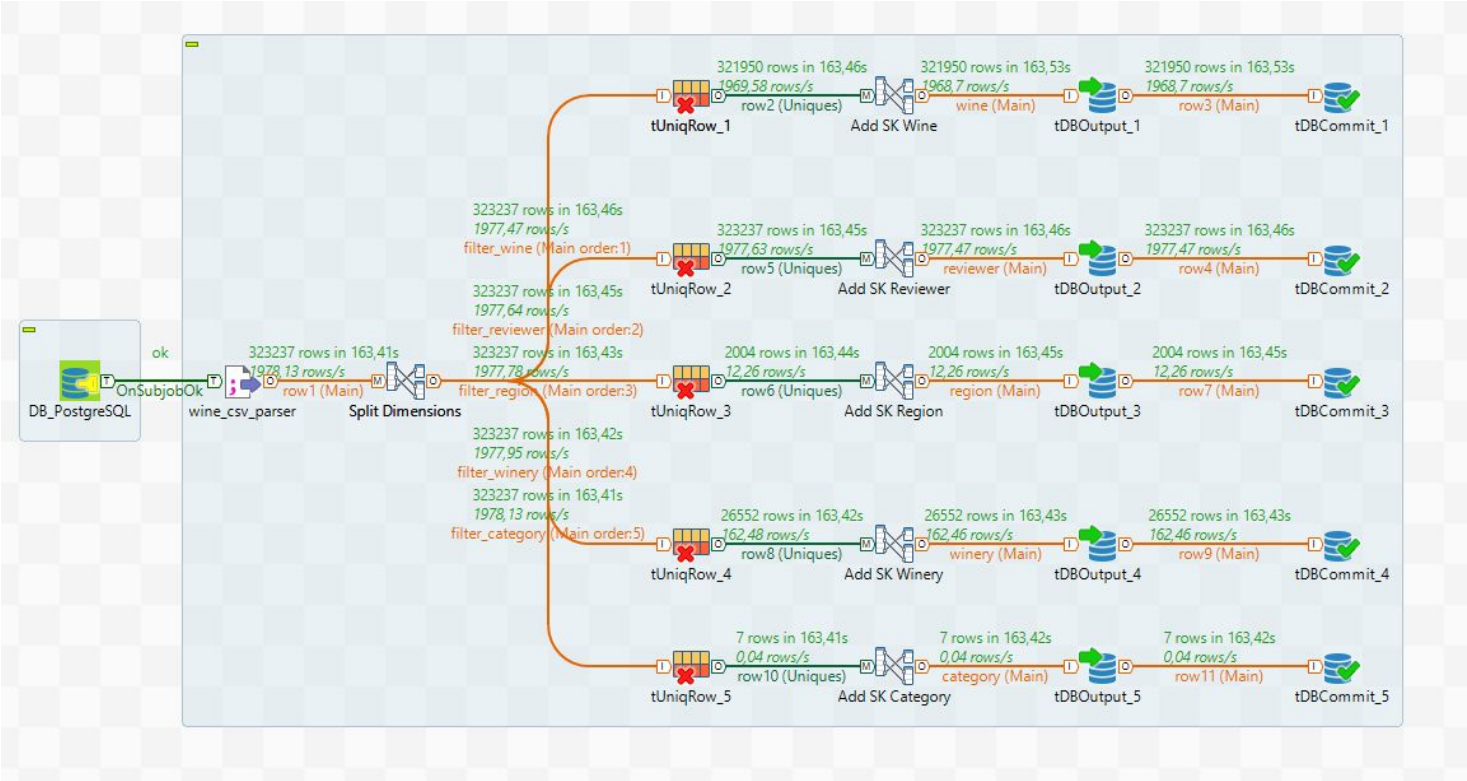


Conceptual Diagram

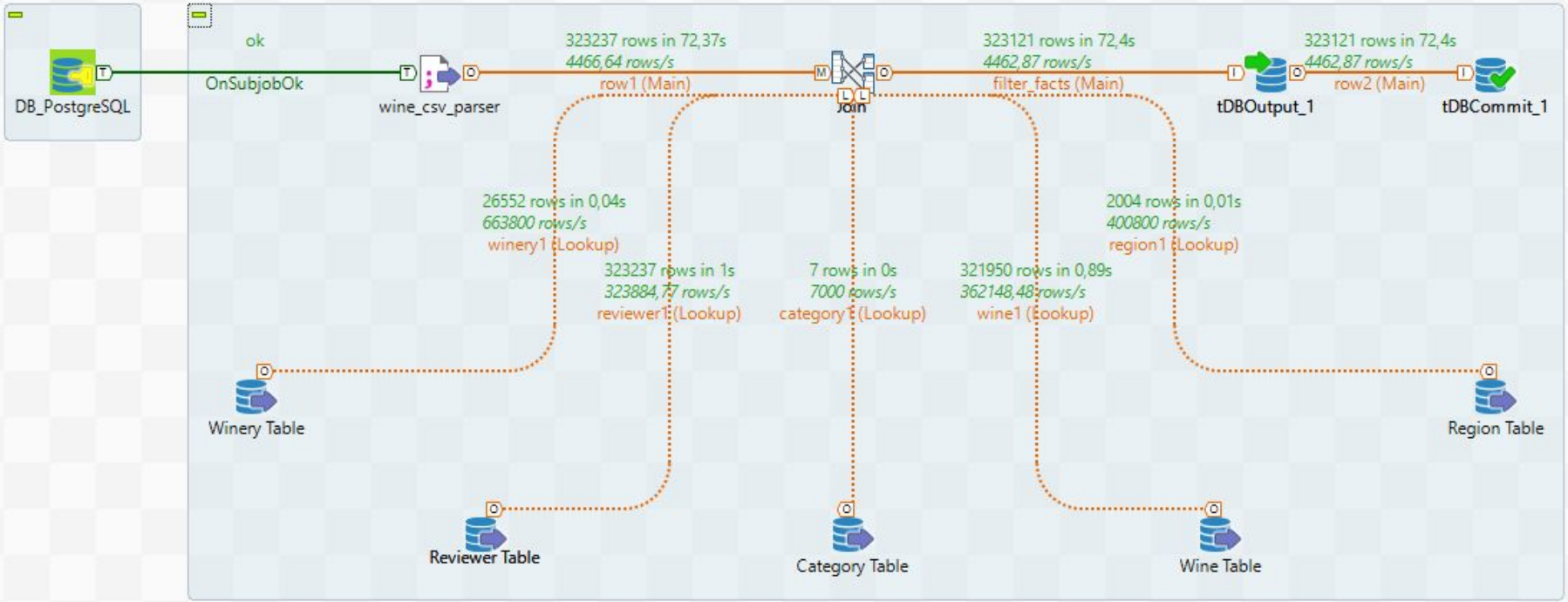


Physical Diagram

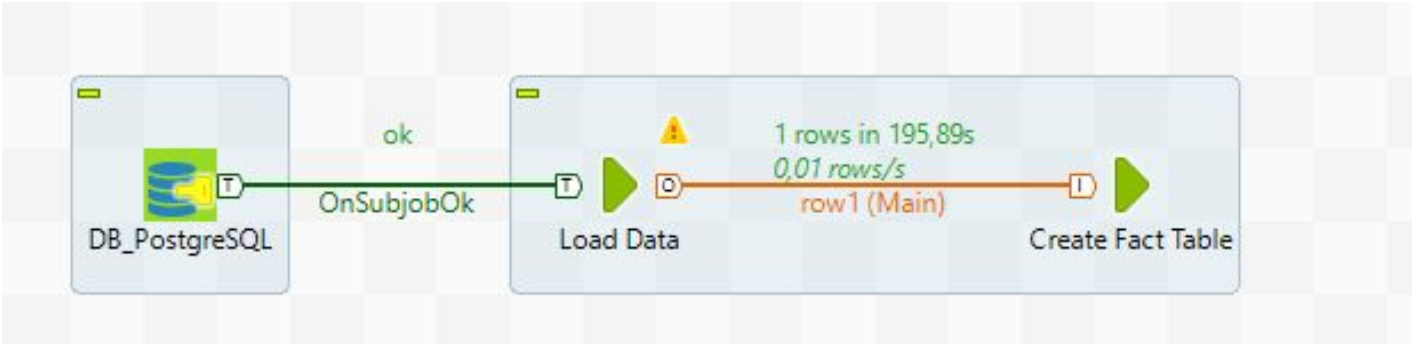
Data Integration Plan - TOS - Dimensions



Data Integration Plan - TOS - Facts Table

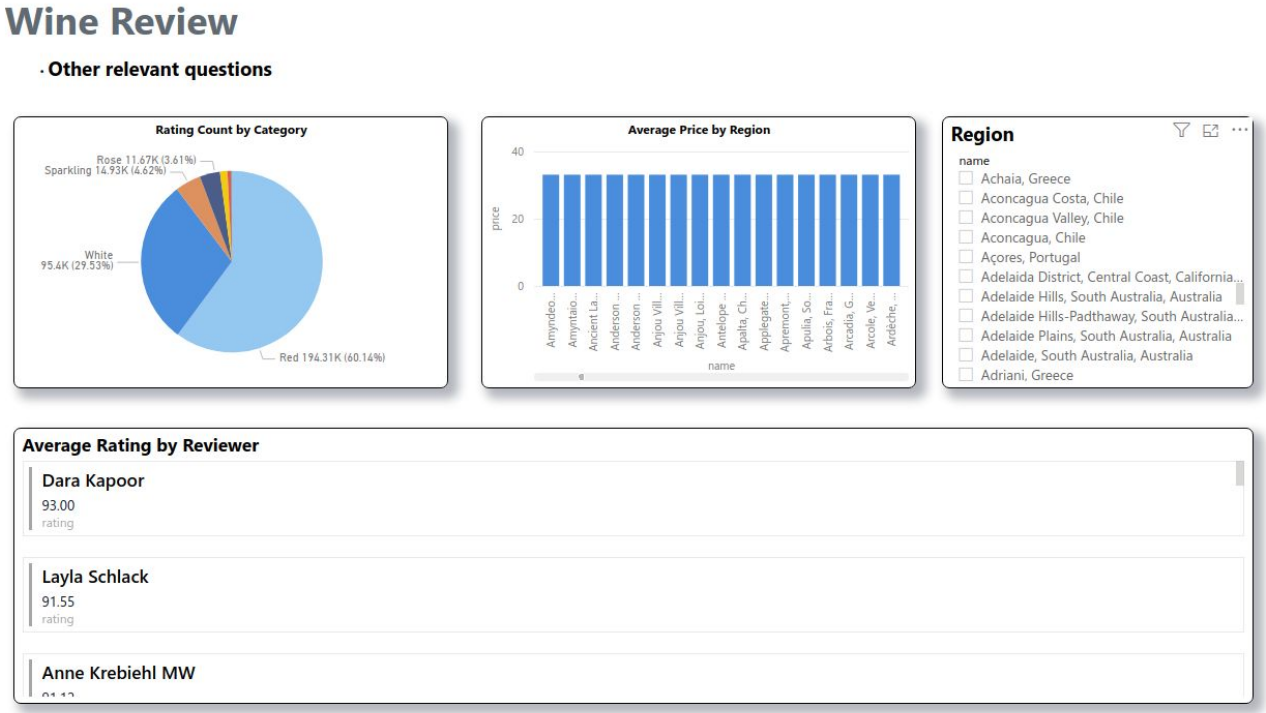


Data Integration Plan - TOS - Job Pipeline



Data Application - MS Power BI

Data Application - MS Power BI



Data Mining

Goals

Having answered some of our initial questions, now we are looking forward to dive deeper into the collected data. The main data mining questions that drove our analysis were:

- Can we predict the wine's price based on the other features?
- Can we predict the wine's rating based on the other features?

Having established the questions we want to see answered, they clearly map to a regression and a classification problem, respectively.

Data Preprocessing - Process

Firstly, in order to apply the data into the models, it needs to be processed. The transformations applied to each feature can be summarized below:

- **category**: One-Hot Encoding
- **region**: Mapped into latitude and longitude coordinates. If not able to find, discard the entry
- **wine_name**: The wine age will be extracted from the name string. If not possible, the age will be 0
- **designation**: Binning(30) and One-Hot Encoding of each bin
- **varietal**: Binning(30) and One-Hot Encoding of each bin
- **alcohol**: Unbounded values (less than 0 or greater than 100) will be bounded to [0.0, 100.0]
- **winery**: Binning(100) and One-Hot Encoding of each bin
- **reviewer**: One-hot Encoding

Binning(**N**) - The categories that have more than **N** entries, will have their own bin. Otherwise they will be mapped to NaN

Data Preprocessing - Results

After applying the previous mentioned transformations, the pandas dataframe was converted from ~320k entries and 10 attributes (**rating**, **category**, **region**, **wine_name**, **designation**, **varietal**, **alcohol**, **price**, **winery** and **reviewer**) into ~5k entries with 620 features.

	rating	alcohol	price	lat	lng	age	category_Dessert	category_Fortified	category_Port/Sherry	category_Red	...
6180	84	13.0	17	41.170042	-7.304750	10	0	0	0	0	...
6201	93	0.0	50	41.170042	-7.304750	8	0	0	0	1	...
6274	87	0.0	0	41.170042	-7.304750	15	0	0	0	1	...
6306	92	13.0	45	41.170042	-7.304750	7	0	0	0	0	...
6317	89	13.5	18	41.170042	-7.304750	14	0	0	0	1	...
...	...	...	...	...	...	...	...	...	...	...	...
320252	86	14.5	20	-35.113970	-71.279980	17	0	0	0	1	...
320253	82	13.0	18	-35.113970	-71.279980	18	0	0	0	0	...
320254	83	14.0	13	-35.113970	-71.279980	16	0	0	0	1	...
320665	91	14.0	25	-34.999037	-71.381712	9	0	0	0	1	...
320671	89	13.5	25	-34.999037	-71.381712	8	0	0	0	1	...

Data Preprocessing - Final Notes

In the preprocessing stage, some decisions were made. We present them below:

- We were time constrained due to the models' processing time for the big amount of data available in the dataset. Initially, we had more forgiving sized bins, i.e. we allowed a bin to be created with a lower count, but eventually it would lead to more features (due to the One-Hot encoding). When we tried to apply the data to the models, they simply would not run. That is the reason why we only binned categories with a count of 30 or 100, depending on the feature. This reduced the features number from around ~3.5k to the 620 that we presented in the previous slide.

- The previous reduction was not enough, as the models still did not run, so we decided to remove every entry that had a NaN feature. This decision decreased the number of entries from ~320k to about ~5k.

These two decisions allowed us to run the models smoothly, without any relevant time constraints.

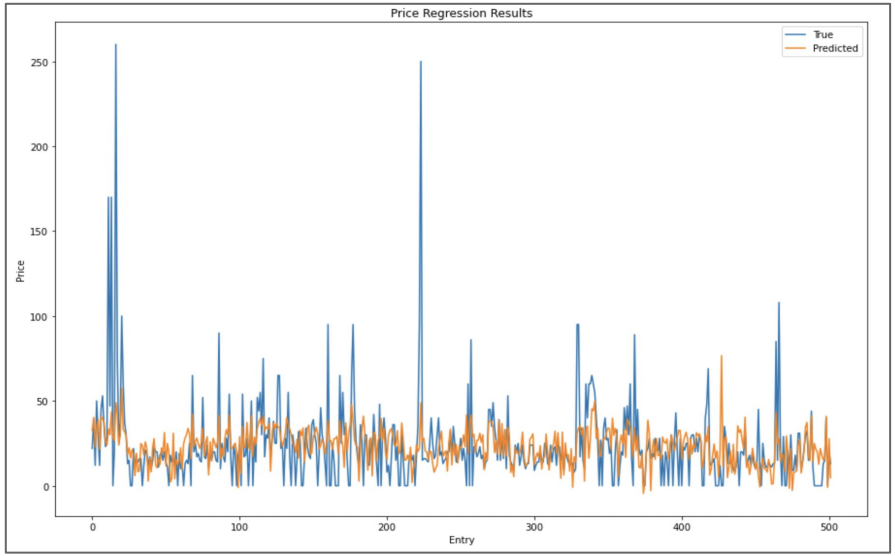
Price Regression

For the price regression, we applied several classifiers to the dataset, like the ones presented to us in the practical classes. The results are shown below:

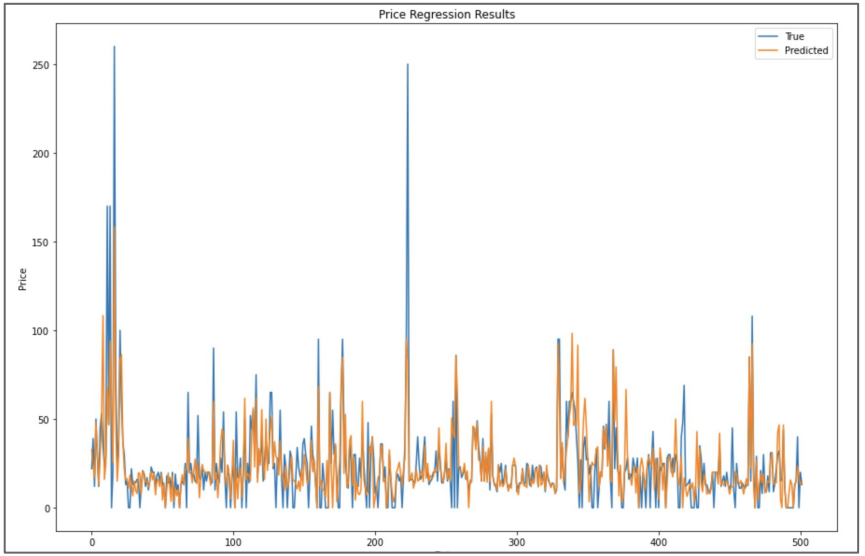
- The classifiers were tested by using kfold with 10 folds.
- The metric selected was the Mean Absolute Error (MAE).
- The models were able to achieve around 10 mean MAE with low (< 1) standard deviation

Model	SVM	KNN	Linear	Logreg	DTree	Lasso	ElasticNet
Mean	12.565	8.598	5.4e8	10.204	7.812	12.403	12.537
STD	0.715	0.426	7.2e8	0.787	0.677	0.434	0.420

Price Regression - Plots



ElasticNet



KNN

Rating Classification

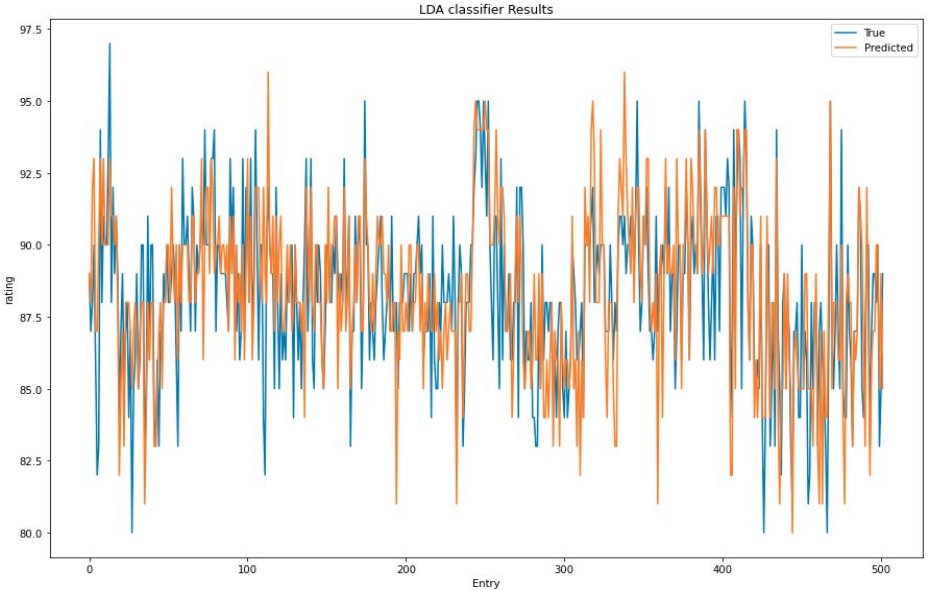
For the rating classification, we tried to apply several classifiers to the dataset, like the ones presented to us in the practical classes, but without much success. None was able to properly classify the entries.

- The classifiers were tested by using kfold with 10 folds.
- The better ones could only get around 20% accuracy in each fold.

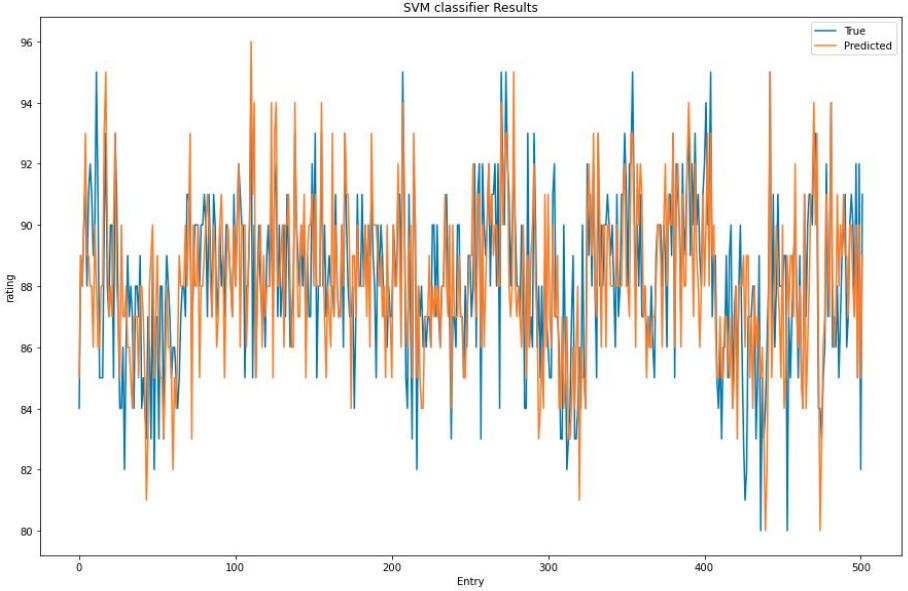
Clearly, these results are not good, giving us the impression that the rating does not depend on the features that we have, or at least not only on these features. Therefore, we conclude that more and better data is needed in order to face this classification problem.

Model	**KNN**	**Decision Tree**	**SVM**	**LDA**	**GaussianNB**
Accuracy (%)	18.745	19.900	20.976	21.036	7.590
STD (%)	1.606	0.794	1.195	1.8430	1.136

Classification results



A fold from LDA.



A fold from SVM.

The End