# IMDB Movie Clustering and Recommender System

by M. Boyd-Vasiliou, A. Weber, Z. Zaidi

### Background

Our employer has acquired access to a full set of movies to stream to new customers. However, our customers have no watching history that we can build upon to provide recommendations, so we have to start with suggesting movies by similarity to each other. To do this we will use machine learning to cluster the movies along available details, including actors, genre, director and ratings. We will create a simple interface so customers can select a movie that they’re interested in, then be shown further recommendations to choose from.

### Proof-of-concept

Key Metrics and Functionality

Performance

* Search must be able to generate at least 10 appropriately related movies
* Appropriateness will be measured by application ‘stickiness’, that customers perform at least one click-through more than 50% of the time

Data Capture

* System must log its inputs and click-throughs, for performance analysis and building customer-specific data for modeling

Privacy

* Logged customer data must be stored and analyzed anonymously, and limited in corporate visibility to the scope of providing future lookups

## Data Overview

The IMDB dataset contains information on over 85,000 movies, from 1874 to the present, including cast information and rating scores. The data is obtained from here: <https://www.kaggle.com/stefanoleone992/imdb-extensive-dataset>.

The data is broken down into four files:

IMDb movies.csv

* All of the movies with summary data, including cast and ratings, one record per movie

IMDb title\_principals.csv

* Cast and primary crew, by movie, one row per person. Individuals are identified by a name key

IMDb names.csv

* Detail of individuals by name key, including full name, height, birthdate, short bio

[ \* First four names: Fred Astaire, Lauren Bacall, Bridget Bardot, John Belushi ]

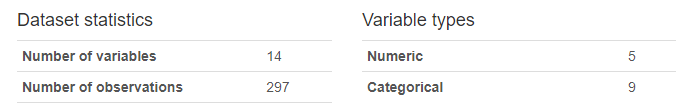
IMDb ratings.csv

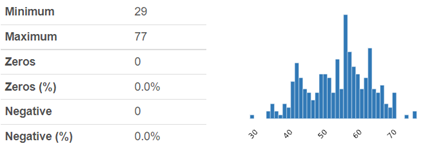
* Summary of customer movie ratings by movie, one row per movie, with ratings breakdown by demographics (gender, age)

We used the first two files, which we will refer to further as “Movies” and “Principals”.

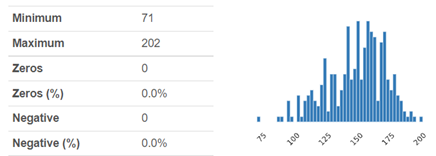
### Data Exploration

| Column Name | Short Description | Notes for our application |
| --- | --- | --- |
| imdb\_title\_id | Unique Movie ID |  |
| title | Name of Movie | Sometimes a translation, dropped |
| original\_title | Name of Movie | A few duplicates, to be expected with remakes |
| year of release | year movie released | One bad record with extra text, fixed. All converted to numeric |
| date of release | Actual release date | As above, but to day and month, dropped |
| genre | Classification into by category | 25 categories in all (Action, Adventure, Comedy, etc) Movies may be in more than one, for example: “ |
| duration |  |  |
| country |  |  |
| language |  |  |
| director |  |  |
| writer |  |  |
| production\_company |  |  |
| actors |  |  |
| description |  |  |
| avg\_vote |  |  |
| votes |  |  |
| budget |  |  |
| usa\_gross\_income |  |  |
| worlwide\_gross\_income |  |  |
| metascore |  |  |
| reviews\_from\_users |  |  |
| reviews\_from\_critics |  |  |



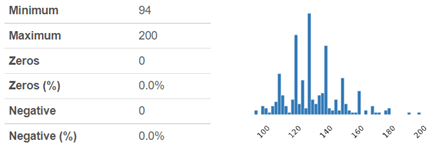
Numeric Data

Age

Age has a relatively normal distribution, with a mean of 55 years. Older than an average population, which makes sense as this group have all been examined in hospital for possible heart conditions. It is interesting to note the double-hump, with a secondary density of patients in their early forties.

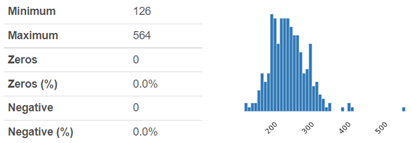
Max\_Heart\_Rate

A normal distribution, with a mean of 150. The min value of 71 looks like an outlier, this is supposed to be an exercise test and that figure suggests exercise was not performed, whatever the reason.

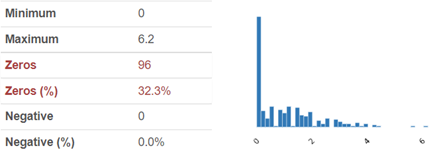


Blood\_Pressure

Normal distribution, mean of 132. Also the high level markers indicate some patients under considerable duress.

Cholesterol

Normal distribution, mean of 247. The max number of 564 looks off the scale, but figures above 500 are within the range of expected results, it’s simply considered a very high score



ST\_Depression

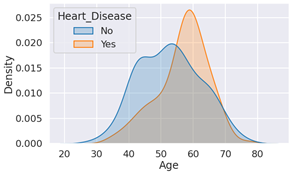
Lopsided distribution, and a third of the values are zero, which is a normal, healthy condition. The ST\_Depression is measured by exercise relative to rest, and the test is supposed to stop when the gap reaches 2mm. There are values in the data that go well beyond, up to 6.2mm. If we hard-cap the max value at 2, we’d in effect create a bin at 2, similar to the naturally occurring bin on the other side of the scale. What we did was create three categories for ST\_Depression, as follows, and it is analyzed further with the other categorical features.

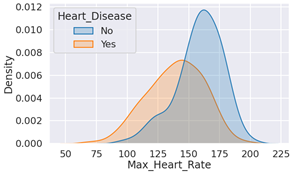
| Category | ST Depression Range | Qty |
| --- | --- | --- |
| None | 0 | 102 |
| Low | 0.1 - 1.5 | 121 |
| High | > 1.5 | 74 |

Correlation of Numeric features to Heart Disease

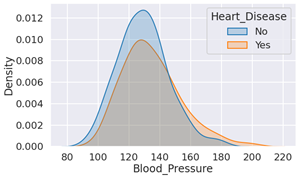
| Mean Value of Numeric features compared across presence of Heart Disease | | | | |
| --- | --- | --- | --- | --- |
| Heart\_Disease | Age | Blood\_Pressure | Cholesterol | Max\_Heart\_Rate |
| No | 53 | 129 | 243 | 159 |
| Yes | 57 | 135 | 252 | 139 |
| All | 55 | 132 | 247 | 150 |

This chart above shows the mean scores of patients grouped by whether they had heart disease. As expected, age, blood pressure and cholesterol all averaged higher in the disease group. Conversely, a higher value for Max Heart Rate indicates a healthier heart (able to beat faster), and so its correlation goes in the opposite direction, with a lower score associated with the disease group. It is also the strongest correlation of the four, as can be seen in the following graphs.

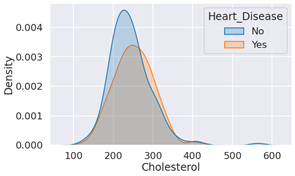
***Age vs Disease*** 

***Max\_Heart\_Rate vs Disease*** 

***Blood\_Pressure vs Disease***

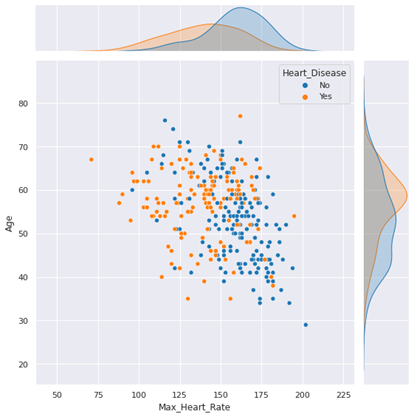


***Cholesterol vs Disease***



And as expected, Age and Max Heart Rate showed the most correlation with each other, with younger patients having the higher Max Heart Rate, so there’s a general downslope to the scatterplot, with a trend away from Heart Disease:

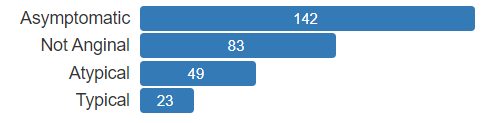
***Max \_Heart\_Rate vs Age, Disease***



Categorical Data

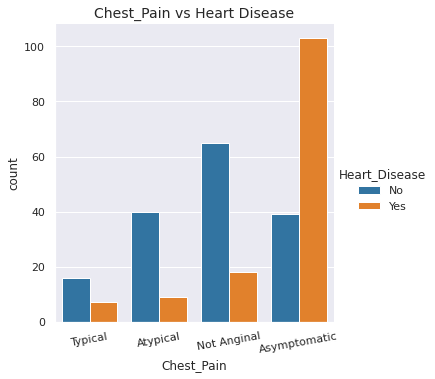
Heart\_Disease

46% of the patients in the dataset were identified as having a heart disease.



Chest\_Pain

There are three classes of angina (chest pain) according to these criteria:

* Chest pain occurs around the substernal portion of the body
* Pain is experienced after induction of emotional/physical stress
* The pain goes away after taking nitroglycerine and/or a rest

**Typical**: All criteria present

**Atypical**: Two of three criteria

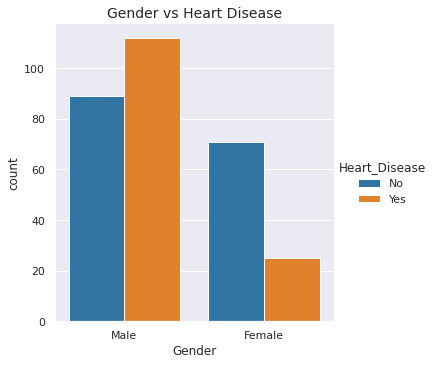
**Non-Anginal**: One criteria satisfied

**Asymptomatic**: None of the criteria are satisfied

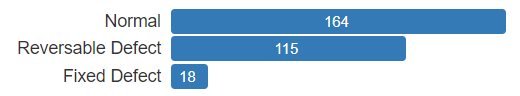
Our data has a very strong correlation between Chest\_Pain and Heart Disease, with the category Asymptomatic representing almost half of the records



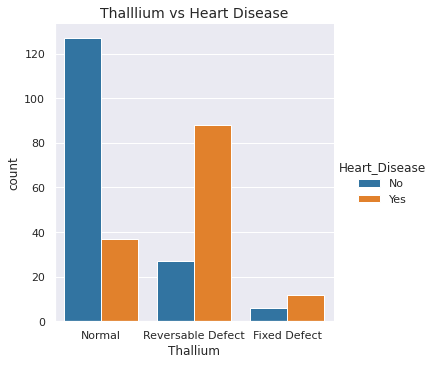
Gender



Men outnumber women in the sample cohort by a 2-to-1 margin, remember this is not a random sample, but patients admitted to hospital to be checked for heart disease. Even within this group however, there is a large discrepancy in the probability of having the disease, 55% for the men, and only 26% for the women.



Thallium



A thallium stress test is a nuclear imaging test that shows how well blood flows into your heart while you’re exercising or at rest.

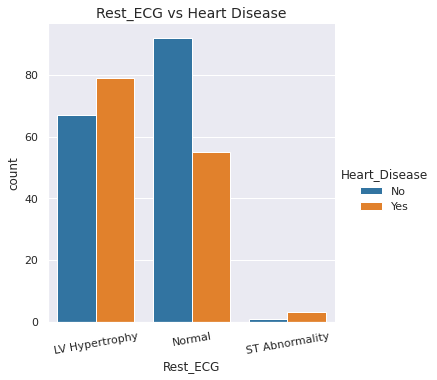
**Normal**: heart tissue is able to absorb thallium

**Reversible Defect**: heart tissue is unable to absorb thallium only under the exercise portion of the test

**Fixed Defect**: heart tissue can't absorb thallium both under stress and in rest



Rest\_ECG



Shape of Resting electrocardiogram wave:

**Normal**: normal

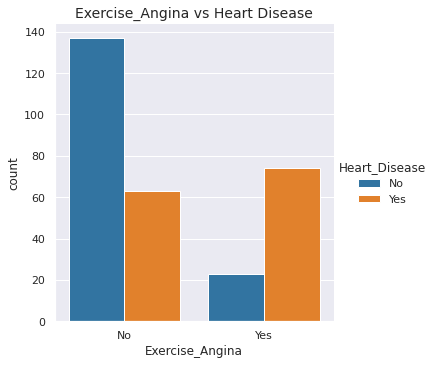
**ST Abnormality**: having ST-T wave abnormality such as T-wave inversions



**LV Hypertrophy**: showing possible left ventricular hypertrophy



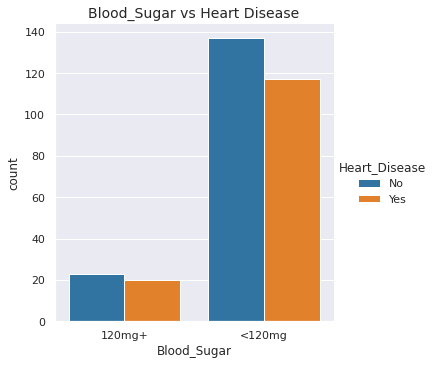
Exercise\_Angina



Patient experiences chest pain after exercise

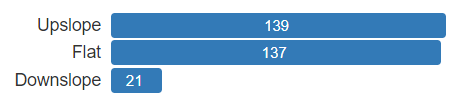
Correlation is very strong, 76% of patients with the symptom had heart disease, compared to only 32% of patients without



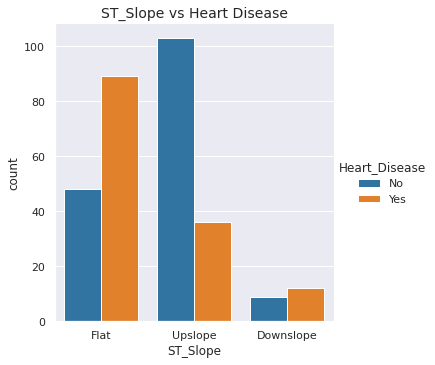
Blood\_Sugar

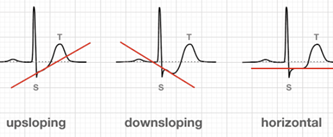
Fasting blood sugar over 120 mg/dL indicates possible diabetes.

Diabetes is associated with increased risk of heart disease, so one would expect a diabetes marker like this to correlate well with it. In our dataset this relationship turned out to be negligible.



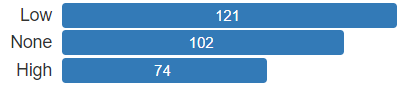
ST\_Slope

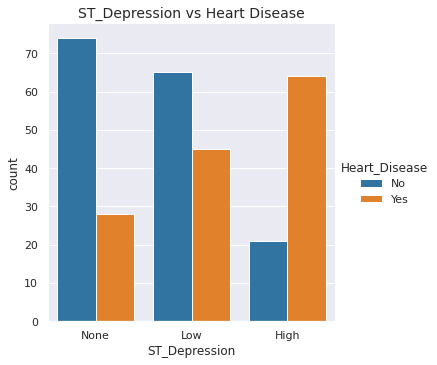
This is another ECG wave observation



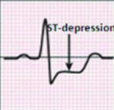
Upslope is the normal, healthy shape

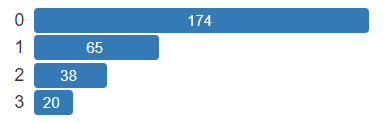
Our Downslope does not show as high a correlation as Flat, but this could be the result of our small dataset, and within that a tiny sample of Downslope records.

ST\_Depression

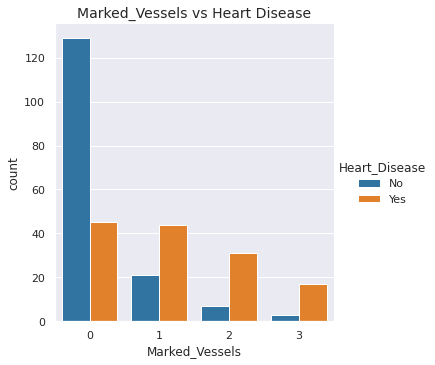
A second observation of the ST segment in the ECG wave.

ST depression induced by exercise, relative to rest. This was a numeric feature in the original data, we divided it into three ranges, as described above in the numeric data section.





Marked\_Vessels

Number of major vessels (0-3) colored by fluoroscopy. 

Radioactive dye is introduced to the body followed by x-ray imaging to detect any structural abnormalities present in the heart. The quantity of vessels colored is positively correlated with presence of heart disease.

### Data Preprocessing

The data exploration revealed a number of factors that could be adjusted for in the setup of the transformation pipeline to help tune the model.

#### Sample and Split

Train Test Split - The default model train/test split of 70% (0.7) did not perform well due to the smaller size of the data set. To provide more data for the model training, the train size was increased to 90% (0.9)

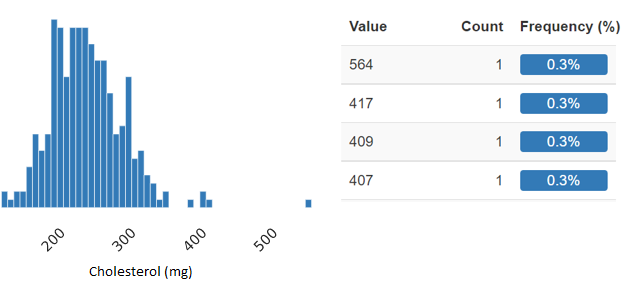


#### Data Preparation

Ordinal Encoding - By binning the ST\_Depression column into “None”, “Low”, “High” ordinal encoding was added during setup to retain their intrinsic order/ranking.



Outliers - During data exploration anomalous values were encountered in a number of records. For this reason remove\_outliers was set to True during the setup. For example the column cholesterol has a mean of 247, standard deviation of 52, and a few numbers that are 5+ standard deviations from the mean. This is not a long tail, but sporadic data in a sparse tail.



With an initial set of 267 records, outlier detection removed 12 down to 255 records, with a Train/Test split of 228/27.



#### Scale and Transform

Normalization - The dataset has several numeric columns, each with a very broad range of possibilities. Normalization was applied to rescale the values of the numeric columns.



#### Feature Engineering

Bin Numeric Features - Due to the number and range of unique values , the continuous value of “Age” was binned during setup to minimize influence on the trained model



#### Feature Selection

Remove Multicollinearity - Data analysis shows that a number of columns possess a high correlation. To minimize this impact on the model “Remove Multicollinearity” was set to true.



Principle Component Analysis - Due to high level of related data PCA was set to True during setup to reduce the dimensionality of the data



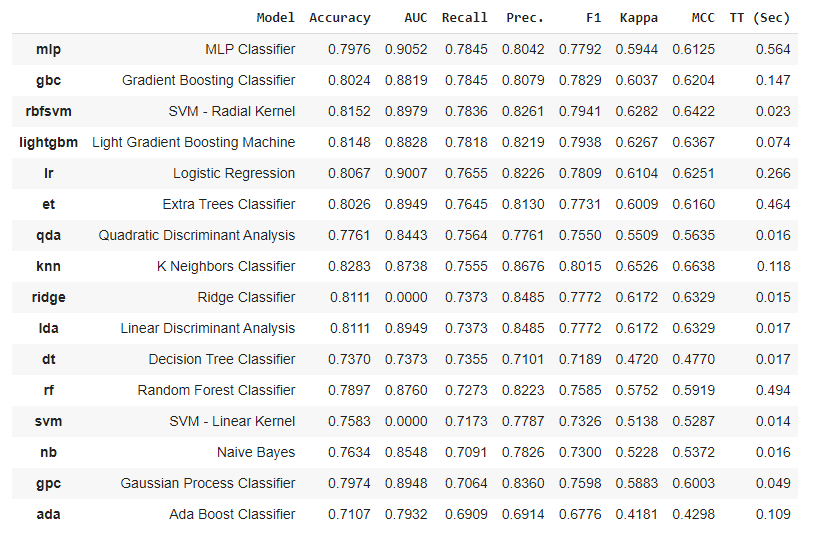
### Model Comparison

When generating the initial comparison of models several options were utilized:

**Number of Folds** - Number of folds was explored and left at the default value of 10. Due to the smaller size of the data set a reduction in the number of folds would reduce the size of the training component on the “train/test” split during model creation. Maximizing the amount of data in the train split provided a positive benefit to the model.

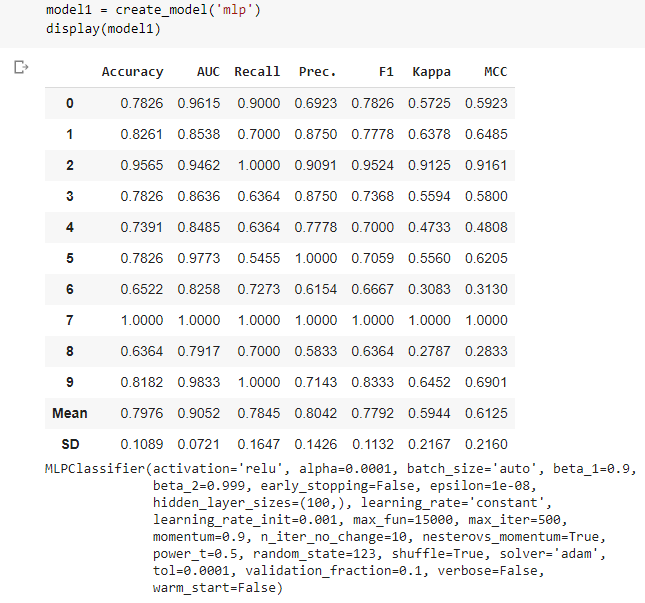
**Optimization** - There are three different directions to tune a classification model, depending on what you are trying to achieve and how it’s reflected in the Confusion Matrix. If the goal is to maximize True predictions, then Accuracy is the only metric needed, because all Trues are good. On the other hand, with False predictions we often prefer to err on one side or the other, depending on the relative cost of an incorrect Positive vs an incorrect Negative prediction. Precision is how we minimize false Positives, and Recall is minimizing false Negatives. In our situation we don’t want to miss an actual case of disease, so we tune our model to optimize its Recall performance.

**Additional Models** - The compare model feature of “Turbo” was set to false to enable the inclusion of additional models in our comparison. The additional models are more resource intensive in their creation and so are left out by default.



## Model Analysis

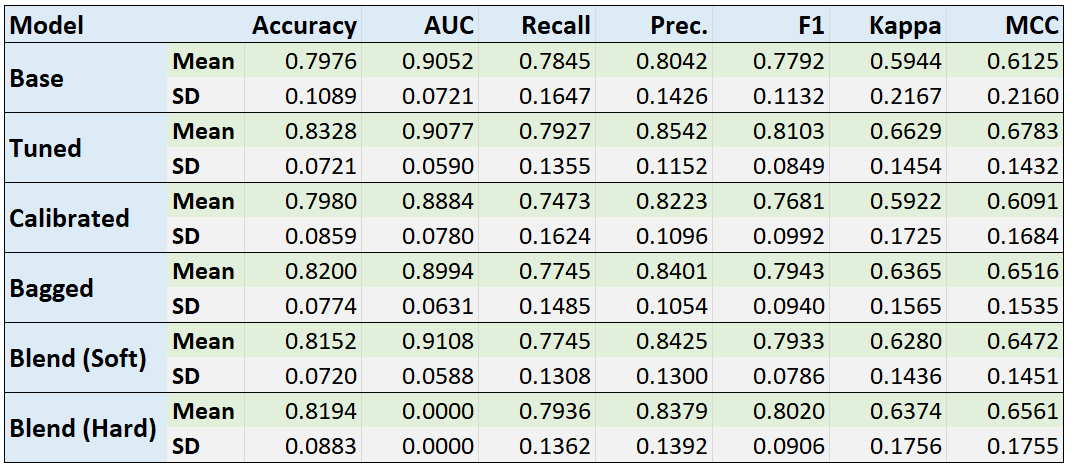
Based on the pipeline setup parameters and the options in the compare model, the top performing model while optimizing for Recall is the MLP Classifier (mlp) model.

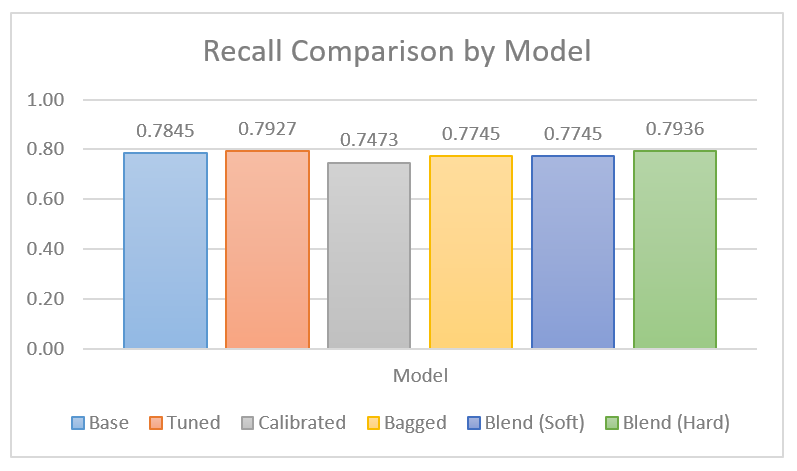


Using the MLP Classifier as our base model, analysis and model exploration was completed. Model options were created for:

* Base model
* Tuned model
* Calibrated model
* Bagged model
* Blended model (soft) - (MLP Classifier, Gradient Boosting Classifier, SVM - Radial Kernel)
* Blended model (hard) - (MLP Classifier, Gradient Boosting Classifier, SVM - Radial Kernel)

## Model Comparison



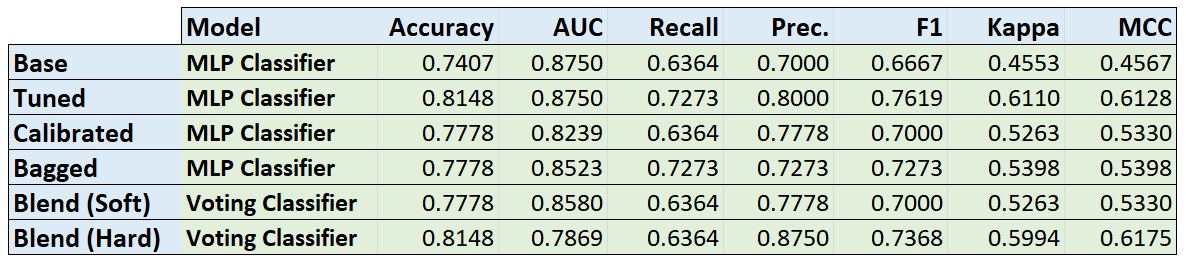


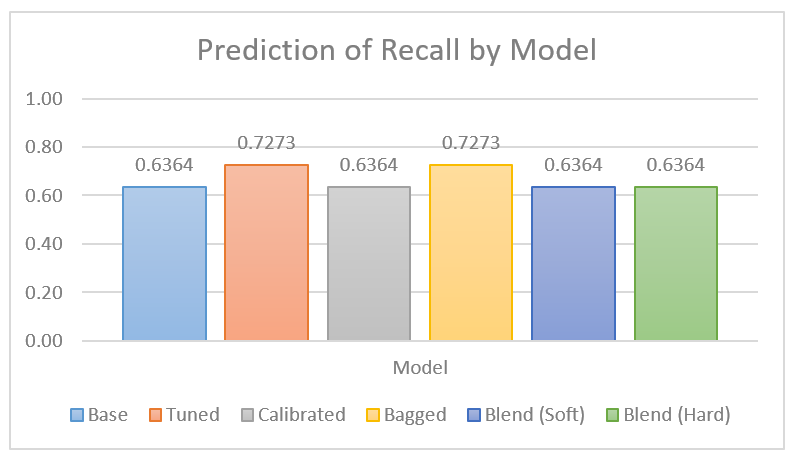
As we are optimizing for Recall, from the chart it’s clear that the “Blended model (hard)” at 0.7396 outperforms the next best model, “Tuned” at 0.7927. It is however an extremely marginal increase, and if you include the lower standard deviation of the Tuned model results the difference is negligible.

### Model Prediction Comparison

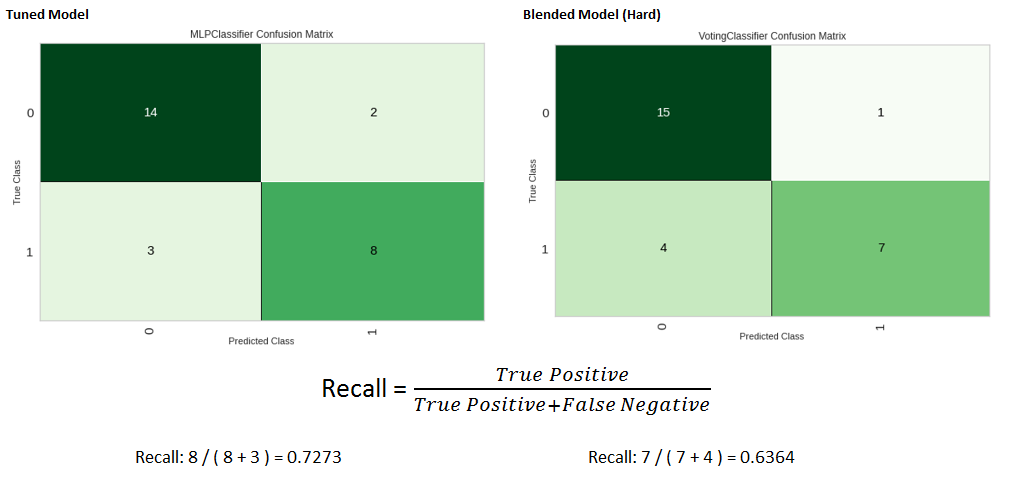
Each model was run through “predict\_model” for a further comparison. e.g for tuned\_model:







We notice that the Recall results fluctuate between two distinct values 0.6364 and 0.7273. While this difference may seem significant it’s important to look at the origin of those numbers. If we examine the confusion matrix for the “Tuned” and the “Blend (Hard)” models:

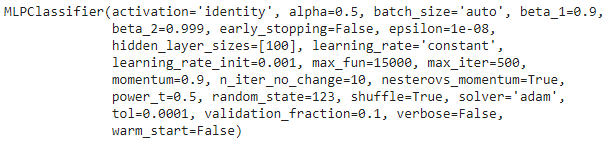


Due to the smaller data set size, the recall range on test data is shifted by .0909 from a single observation changing it’s prediction category. While the prediction is effective, it’s clear that both the model training and the model choice would benefit greatly from a much larger dataset.

### Best Model

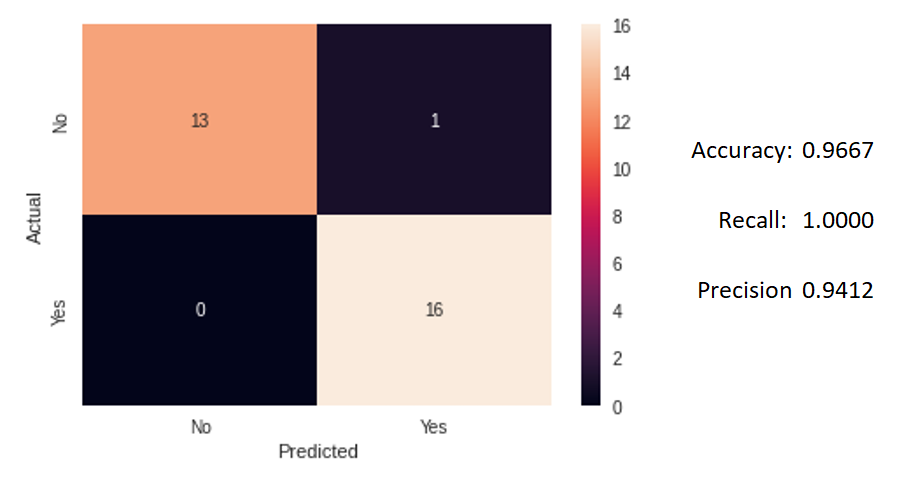
While the model with the best performing “Recall” is the “Blended model (hard)”, the model only outperforms the “Tuned” model by only 0.0009. Additionally the standard deviation in the “Tuned” model is lower by 0.0007.

The more important analysis factor is the smaller size of the data. Since these two models are virtually identical in “Recall” metrics they can be viewed as approximate equivalents. However, since the data size is relatively small, choosing the single “Tuned” MLP model, rather than a more complicated blending of models is a simpler and more applicable approach.



A prediction of the finalized model against the unseen data was completed with promising results, and no false negatives.





## Conclusion

The prediction modeling provides a satisfactory method for detecting heart disease based on our test results. The current model set is smaller than would be ideal but the model creation, predictions, and testing proves that this would be an effective endeavor. Implementation should include a continuous retraining and update of the model as new information is available. There are a number of opportunities to utilize this model or modified models to improve client care, save money, and improve resource planning.

An area of opportunity would be to conduct a business flow analysis, to evaluate the order in which tests should be completed, and results returned. This could allow opportunities to improve patient triage by scheduling appointments for those who have the higher risk of heart disease. Improvements to client care, budgeting, as well as space and resource planning could be forecast if some modeling predictions can be obtained at earlier decision points.

The model meets our performance criteria with a false-negative rate < 5% and a false-positive rate < 30%, strong model documentation to ensure reliability and repeatability, and a web interface provides simple cross platform usability.

Integrating this model into the business workflow in a more automated fashion would be far more beneficial than the web application. Ideally, as patient test results are entered into an electronic medical system (EMS) it would trigger an API call to the model and flag the patient status. This “flag” could be aggregated across all clients to remove private information, and shared with other stakeholders to improve their business forecasting capabilities to have an idea of the number of patients requiring future support and resources.

The implementation and utilization of this model could have a significant benefit to a number of stakeholders whether they be the patient, doctor, or medical administration. There are two main areas of concern with implementing a model such as this: predicting the client has no heart disease but actually does (false negative), and the transmission/storage/usage of personal information.

To alleviate these concerns two things should occur. First, the business process should be investigated and adjusted to ensure that when the model is integrated into the workflow, adequate rigour is taken to ensure patients do not go undiagnosed. Secondly, the model does not require any personally identifying information, so any API implementation should restrict the collection, submission, transmission of any personal information. Any automated submission and matching of information should ensure that all personal data is obfuscated, and encrypted during transmission as well as at rest.

## Bibliography

Health Care Services—Nunavut. 2017 March Report of the Auditor General of Canada. URL: <https://www.oag-bvg.gc.ca/internet/english/nun_201703_e_41998.html>

Nillsf. Confusion Matrix, accuracy, recall, precision, false positive rate and F-scores explained. URL: <https://blog.nillsf.com/index.php/2020/05/23/confusion-matrix-accuracy-recall-precision-false-positive-rate-and-f-scores-explained/>

Heart Disease UCI. Kaggle. URL: <https://www.kaggle.com/cherngs/heart-disease-cleveland-uci>

O. Pelivan. HeartDisease.URL:<https://www.kaggle.com/onatto/predicting-heart-disease-a-detailed-guide>

Jesse Charis. JCharis Tech. How to Split Dataset into Training and Testing Dataset for Machine Learning. URL:<https://blog.jcharistech.com/2020/09/23/how-to-split-dataset-into-training-and-testing-dataset-for-machine-learning/>

Pycaret. Pycaret - Preprocessing. URL: <https://pycaret.org/preprocessing/>

Towards Data Science. Moez Ali. Build and deploy your first machine learning web app. URL: <https://towardsdatascience.com/build-and-deploy-your-first-machine-learning-web-app-e020db344a99>