# Comparison

SVC is slow – runs in O(n^2) time so not appropriate for the training set.

List three of the supervised learning models above that are appropriate for this problem that you will test on the census data. For each model chosen

* Describe one real-world application in industry where the model can be applied.
* What are the strengths of the model; when does it perform well?
* What are the weaknesses of the model; when does it perform poorly?
* What makes this model a good candidate for the problem, given what you know about the data?

\*\* HINT: \*\*

Structure your answer in the same format as above^, with 4 parts for each of the three models you pick. Please include references with your answer.

## AdaBoost

*Describe one real-world application in industry where the model can be applied.*

There are plenty of examples where AdaBoost would be an appropriate choice for a classifier. A few of them would be:

* Predicting whether a patient would get a heart decease
* Predicting customer churn
* Classifying and email as spam or not

*What are the strengths of the model; when does it perform well?*

AdaBoost is a powerful classification algorithm that has enjoyed practical success with applications in a wide variety of fields, such as biology, computer vision, and speech processing. Unlike other powerful classifiers, such as [SVM](https://www.nickgillian.com/wiki/pmwiki.php/GRT/SVM), AdaBoost can achieve similar classification results with much less tweaking of parameters or settings (unless of course you choose to use SVM with AdaBoost). The user only needs to choose: (1) which weak classifier might work best to solve their given classification problem; (2) the number of boosting rounds that should be used during the training phase.

*What are the weaknesses of the model; when does it perform poorly?*

AdaBoost can be sensitive to noisy data and outliers. In some problems, however, it can be less susceptible to the overfitting problem than most learning algorithms.

*What makes this model a good candidate for the problem, given what you know about the data?*

During our data preprocessing step, we've normalized the features, which reduces the impact of the outliers and skewed features, which can potentially harm the performance of AdaBoost. Also, as AdaBoost uses a sequence of weak classifiers, it's less expensive computationally compared to other powerful algorithms like SVM. Out training data has almost 40 000 records, which already means we need a more performant algorithm. This is again where AdaBoost fits better.

*Resources:*

<https://www.nickgillian.com/wiki/pmwiki.php/GRT/AdaBoost>

<https://stats.stackexchange.com/questions/8930/when-would-one-want-to-use-adaboost>

<https://www.youtube.com/watch?v=LsK-xG1cLYA>

<https://hackernoon.com/under-the-hood-of-adaboost-8eb499d78eab>

## Decision Tree

*Describe one real-world application in industry where the model can be applied.*

The real-world applications are too many. Some of them are:

1. Predicting high occupancy dates for hotels
2. Identifying factors leading to better gross margins on a retail chain
3. Identifying correlates to high average checks for a global quick-service restaurant chain

*What are the strengths of the model; when does it perform well?*

1. Compared to other algorithms decision trees requires less effort for data preparation during pre-processing.
2. A decision tree does not require normalization of data.
3. A decision tree does not require scaling of data as well.
4. Missing values in the data also do NOT affect the process of building a decision tree to any considerable extent.
5. A Decision trees model is very intuitive and easy to explain to technical teams as well as stakeholders.

*What are the weaknesses of the model; when does it perform poorly?*

1. A small change in the data can cause a large change in the structure of the decision tree causing instability.
2. For a Decision tree sometimes calculation can go far more complex compared to other algorithms.
3. Decision tree often involves higher time to train the model.
4. Decision tree training is relatively expensive as complexity and time taken is more.
5. Decision Tree algorithm is inadequate for applying regression and predicting continuous values.

*What makes this model a good candidate for the problem, given what you know about the data?*

During the comparison of the algorithms against our training data, the Decision Tree classifier performed almost as good as the AdaBoost classifier. It didn't show any signs of overfitting with correctly chosen hyperparameters.

*Resources:*

<https://www.quora.com/What-are-the-disadvantages-of-using-a-decision-tree-for-classification>

## Random Forest

*Describe one real-world application in industry where the model can be applied.*

**Banking Sector:**The banking sector consists of most users. There are many loyal customers and also fraud customers. To determine whether the customer is a loyal or fraud, Random forest analysis comes in. With the help of a random forest algorithm in machine learning, we can easily determine whether the customer is fraud or loyal. A system uses a set of a random algorithm which identifies the fraud transactions by a series of the pattern.

*What are the strengths of the model; when does it perform well?*

Random forest algorithm can be used for both classifications and regression task.

It provides higher accuracy.

Random forest classifier will handle the missing values and maintain the accuracy of a large proportion of data.

If there are more trees, it won't allow overfitting trees in the model.

It has the power to handle a large data set with higher dimensionality

*What are the weaknesses of the model; when does it perform poorly?*

Random Forest classifier is similar to Decision Tree. The main disadvantage is probably that it is a lot harder to analyze and explain the model.

*What makes this model a good candidate for the problem, given what you know about the data?*

It is a right candidate due to the same reasons are the Decision Tree classifier. During the testing, it even slightly outperforms the Decision Tree, so the Random Forest classifier is one of the strongest candidates to give the best results in the present data set.

*Resources:*

<https://www.newgenapps.com/blog/random-forest-analysis-in-ml-and-when-to-use-it>

<https://www.researchgate.net/publication/259543750_Mining_data_with_random_forests_Current_options_for_real-world_applications>

<https://stats.stackexchange.com/questions/285834/difference-between-random-forests-and-decision-tree>

# Winner

The most performant algorithm is AdaBoost. Here is an Accuracy and F-Score comparison between the algorithms:

|  |  |  |
| --- | --- | --- |
|  | Accuracy | F-score |
| AdaBoost | 0.8576 | 0.7246 |
| Decision Trees | 0.8286 | 0.6579 |
| Random Forest | 0.8415 | 0.6805 |

The three algorithms' performance is very close. But even after tunning the hyperparameters, AdaBoost has a small advantage. AdaBoost seems to be quite a bit slower to train the training set compared to the other algorithms, but with the training set size we have, this is acceptable.

The optimizer score is slightly better than the unoptimized. The optimized model outperforms the naïve classifier significantly.

1. educational-level
2. capital-gain
3. hours-per-week
4. age
5. native-country

The most significant difference compared to my prediction is the capital-loss feature that I didn't even include in my top 5 list.