**Green = Passed**

**Blue = Edited for this change**

**Orange = Edited but I’m not sure that the new answer is correct**

Requires Changes

**10 SPECIFICATIONS REQUIRE CHANGES**

Very strong first submission! Well done! Go through the comments carefully and amend where necessary. You're almost there!

I'm not a native english speaker. So some sentences or words may cause you confused.  
But please feel free to comment me. : )

* You're doing really well considering, keep it up

**Exploring the Data**

**Student's implementation correctly calculates the following:**

* **Number of records**
* **Number of individuals with income >$50,000**
* **Number of individuals with income <=$50,000**
* **Percentage of individuals with income > $50,000**

**Note**

* Great job! It is generally a good idea when faced with new data to collate some summary statistics about the data, this can really help in some cases. Here we have a binary target variable and we get to see how many belong to each class, what this tells us right away is whether there's a class imbalance as this can affect in what direction our analysis will go further on. Two things generally done when there's a class imbalance is to ensure when splitting into train/test sets, they are split as evenly as possible so there isn't too many of one class in a particular set. Another use here is it tells us what kind of evaluation metric is appropriate, or at least isn't. The accuracy metric for example, is considered a bad metric when there's a class imbalance. For ideas on other [summary information](https://en.wikipedia.org/wiki/Summary_statistics) one could view about a dataset.

**Preparing the Data**

**Student correctly implements one-hot encoding for the feature and income data.**

Required

income = [(lambda price: price=='<=50K')(price) for price in income\_raw]

* As it is, individuals making at most 50K are the ones being set to 1, this should be the other way round.

**Evaluating Model Performance**

**Student correctly calculates the benchmark score of the naive predictor for both accuracy and F1 scores.**

Required

Naive Predictor: [Accuracy score: 0.7522, F-score: 0.7914]

* This is incorrect, consider for a second that accuracy should represent the score if we assumed everyone made over 50K, but since we know that in this project as well as in real life, most people don't make over 50K. The accuracy can not be 75%,

**The pros and cons or application for each model is provided with reasonable justification why each model was chosen to be explored.**

**Please list all the references you use while listing out your pros and cons.**

Required

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

* The above question hasn't been sufficiently answered, what is expected here is to include why the models were chosen given what is known about the problem. Basically what this part is asking is to justify picking the models given this particular problem. To answer this, simply consider the characteristics of both the data and the models and why they might be appropriate for each other.

**Note**

Choosing a model for a particular problem is a very interesting aspect of analytics that does require some careful consideration and that is what makes this particular question important, one approach that one may always use involves simply trying out a bunch of models and seeing which performs best, another approach involves considering the characteristics of the models and of the problem being worked on, as well as a number of different factors that may be broken down into different areas -

**Time**

* It is important to consider how long it would take during training or prediction, a model like Gaussian Naive Bayes would generally perform quickly while others like SVM would not.

**Accuracy**

* Some models tend to perform better than others, this obviously depends a lot on the kind of problem being worked on. An example of this is the model XGB (Extreme Gradient Boosting) this model is quite popular in the Kaggle community as it's known to perform quite well and win competitions.

**Interpretability**

* Decision tree based algorithms are generally known to be great here as they help see quite clearly how different features contribute to the predictive capability of the model. An artificial neural network for example, is a black box model and wouldn't be easily interpret-able.

**Size of Data**

* This I would consider a very important factor as based on the nature of the model, performance may vary based on the kind of data being fed in, if the data is large enough, too large, low/high number of features, low/high number of observations.

**No of Parameters**

* This could go either way to be honest, one may opt for a model with few parameters to tune for simplicity's sake, on the other hand, a model with a lot of parameters to tune provides lots of opportunities to increase the predictive power of the model.

This is not an exhaustive list and is simply a number of things to consider, for more information on this, see the link included -

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-choice>

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-algorithm-cheat-sheet>  
<http://scikit-learn.org/stable/tutorial/machine_learning_map>

**Student successfully implements a pipeline in code that will train and predict on the supervised learning algorithm given.**

Required

# **TODO**: Compute accuracy on the first 300 training samples which is y\_train[:300]

results['acc\_train'] = predictions\_train

* The accuracy should be computed here, and this should only be based on a slice of the first 300 samples.

**Student correctly implements three supervised learning models and produces a performance visualization.**

Required

* The visualization does not look quite right, the accuracy score of the models except Decision Trees should not have a perfect score. This could be related to the objection from the previous section,

**Improving Results**

**Justification is provided for which model appears to be the best to use given computational cost, model performance, and the characteristics of the data.**

Required

* This section should be revisited once the previous section is addressed.

**Student is able to clearly and concisely describe how the optimal model works in layman's terms to someone who is not familiar with machine learning nor has a technical background.**

Required

* Well done on the discussion provided, a more intuitive explanation is required here. As it is, it isn't quite clear how the model's processes works, for decision trees, consider the game 20 questions, this could help in explaining it intuitively. For the Random Forest explanation, it was mentioned that the trees are combined using bagging, note that the kind of person we're explaining to would not know what this is. Also another important concept here is weak learners, this should be mentioned and a discussion included explaining what it means.

<https://en.wikipedia.org/wiki/Twenty_Questions>

**The final model chosen is correctly tuned using grid search with at least one parameter using at least three settings. If the model does not need any parameter tuning it is explicitly stated with reasonable justification.**

Awesome

* Multiple parameters tuned, each with at least 3 settings.

**Student reports the accuracy and F1 score of the optimized, unoptimized, models correctly in the table provided. Student compares the final model results to previous results obtained.**

Required

* This might seem like a nitpick, accuracy and fscores returned should be between 0 and 1. While it's fine to use percentage, this should be added. As it is, the scores look incorrect.

**Feature Importance**

**Student ranks five features which they believe to be the most relevant for predicting an individual's’ income. Discussion is provided for why these features were chosen.**

Required

* It is required to also include a discussion here on each feature, why are each of these believed to be important?

**Student correctly implements a supervised learning model that makes use of the feature\_importances\_ attribute. Additionally, student discusses the differences or similarities between the features they considered relevant and the reported relevant features.**

Required

* For the features that you missed, why do you think they turned out to be important? Note there are 3 questions here, address each of them.

How do these five features compare to the five features you discussed in Question 6?  
If you were close to the same answer, how does this visualization confirm your thoughts?  
If you were not close, why do you think these features are more relevant?

**Student analyzes the final model's performance when only the top 5 features are used and compares this performance to the optimized model from Question 5.**

* Great job! other than the time factor, another reason we might prefer to use fewer features also is that it makes for a more stable model that would generalize better.