**Requires Changes**

**7 specifications require changes**

Dear student,

Great first submission

I would recommend you to spend some more time understanding each of supervised models and try to explain them in simple terms. It will help you understand the model very well and you can easily explain it to others as well.

Please make sure you read the instructions clearly before implementing the code. I have provided suggestions for each of required sections. I am sure the required changes won't take much time and it is worth your time.

Keep up the good work! I look forward to next submission.

**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

The percentage must be in floating point with two decimal vlaues. You could write like :

greater\_percent = (n\_greater\_50k \* 100.0)/n\_records

**Preparing the Data**

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

There are a couple other ways that we can encode the labels here.  
One way is to use boolean indexing.

income = (income\_raw == ">50K").astype(np.uint8)

We can also use the LabelEncoder class provided by sklearn. This class is especially useful when we have lots of possible output labels. We can use it for this problem as follows:

encoder = LabelEncoder()  
income = encoder.fit\_transform(income\_raw)

**Evaluating Model Performance**

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

You correctly calculated both accuracy and f-score. Good work!

You could check this [link](https://www.quora.com/What-is-the-best-way-to-understand-the-terms-precision-and-recall) for further understanding precision and recall.

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.

The reason of selection is not accurate for both svm and logistic regression. You need to answer this question in both dataset properties as well as in model strengths.

Here is a sample reason of selection for svm :

I expected the data to not be linear, given that there are so many features. Hence, using an appropriate kernel like rbf, I would be able to effectively tune the classifier for high score

Please check below videos on svm, decision trees and ensemble methods to futher understand the concepts.

[svm](https://matterhorn.dce.harvard.edu/engage/player/watch.html?id=92e3adbf-2212-4cff-b1a9-b1bfe72d93bf)  
[Decision trees](https://matterhorn.dce.harvard.edu/engage/player/watch.html?id=c22cbde8-94dd-42ad-86ef-091448ad02e4)  
[Ensemble methods](https://matterhorn.dce.harvard.edu/engage/player/watch.html?id=6f374ba5-6e54-432c-9916-fb61fa2327ef)

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

Nice implementation of pipeline!

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

**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.

Nice explanation and I agree with your reason of selection of final model.

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.

You forgot to explain how svm will classify non linear points. You can also include how svm will use on given census data for training and prediction.

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.

Set a random\_state if one is available to the same state you set before.

Make sure you set the random\_state for your final model if provided in sklearn library for reproducible results.

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.

Please update this section after run your final model with random\_state.

**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.

Great selection of features but please explain why these features are relevant for finding individual who make more $50k?

Here are simple explanation of some example features :

* age : Usually older people earn more compared to younger people
* education-num : Income is directly proportional with education\_num(13 bachelors, 14 masters and 16 Doctorate). Higher the num then higher the income
* hours-per-week : If we considered two persons with same hourly rate then the person have higher hours-per-week will earn

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.

An alternative feature selection approach consists in leveraging the power of Recursive Feature Selection to automate the selection process and find a good indication of the number of relevant features (it is not suitable for this problem because that is not what is required by the project rubric, though it is generally a very good approach).  
<http://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html>

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**.

I would recommend you to re-run this section acutally we are using the same final model here on both full and reduced data and I guess the results will change after you run your final model with random\_state.

Please update the description as per results.