PROJECT SPECIFICATION

Finding Donors for CharityML

Exploring the Data

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS

Data Exploration	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
		This can be observed in output of cell 3 in the notebook of the html file. But here is the result:
		`Total number of records: 45222`
		`Individuals making more than $50,000: 11208`
		`Individuals making at most $50,000: 34014`
		`Percentage of individuals making more than $50,000: 24.78%`

Preparing the Data

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Data Preprocessing	Student correctly implements one-hot encoding for the feature and income data.	One-hot encoding is done in cell 7 of the ipython notebook. A total of 103 features are created after one-hot encoding.

Evaluating Model Performance

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Question 1: Naive Predictor Performance	Student correctly calculates the benchmark score of the naive predictor for both accuracy and F1 scores.	The naive predictor scores can be seen from cell 10 output in the ipython notebook. `Naive Predictor: [Accuracy score: 0.2478, F-score: 0.2917]`
Question 2: Model Application	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 answer to question 2 can be seen after cell 10 in the notebook. But here is the answer again. The three models I have chosen are •Logistic Regression •Random Forest Classifier •Gradient Boosting classifier I wanted to test and compare how the basic logistic regresion fares along with a couple of ensemble methods which are supposed to be far suprior and also compare the ensemble methods themselves. **1. Logistic Regression** **Applications:** [1] Logistic regression can be used in many fields such as medical, engineering, marketing etc.. Specific examples include 1. to find a process is effective or not, 2. whether a person defaults on a homeloan or not, 3. a particulr marketing campain would be successful or not.

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		Strengths: [2] Logistic regression is easy to understand and have nice probabilistic interpretation. Regularization can be applied to reduce overfitting. **Weaknesses:** [2] Logistic regression underperforms when there are multiple or non-linear boundaries. Its not a natural fit when the task at hand is complex. **Suitability:** In this project we just need to find whether an individual is making 50K or less. This is a straight forward classification task and hence chose logistic regression to check on how it performs. ## 2. Random Forest Classifier **Applications:** Random forests can be used in a veriety of applications. I'll mention a couple of applications here. 1.Multi-class object detection. [3] 2.Medical diagnosis. [4] **Strengths:** •RF are much easier to tune than GBM. There are typically two parameters in RF: number of trees and number of features to be selected at each node. [5] •RF are harder to overfit than GBM. [5] **Weaknesses:** •Slow to apply to real-time tasks if large number of trees are involved. [6]

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		•Sometimes hard for humans to interpret compared to decision trees. [6] •If the data contain groups of correlated features of similar relevance for the output, then smaller groups are favored over larger groups [6] **Suitability:** Random forest classifier is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier. It also works very well for complex tasks. So, I choose this to test. ## 3. Gradient Boosting Classifier **Applications:** One of the applications where gradient boosting is used is in web ranking algorithms. [7] **Strengths:** Benchmark results have shown GBDT are better learners than Random Forests. (comparing to random forests) [8] **Weaknesses:** Training generally takes longer because of the fact that trees are built sequentially. (comparing to random forests) [8] **Suitability:** Gradient boosting is one of the best methods out there and there is saying even to start off with gradient boosting without considering any other technique. [9] **References:** [1] https://en.wikipedia.org/wiki/

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		Logistic_regression#Applications [2] https://elitedatascience.com/machine-learning-algorithms [3] https://pdfs.semanticscholar.org/9035/e87ce49b67b751838c7346d36fe481260217.pdf [4] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2648734/ [5] https://medium.com/@aravanshad/gradient-boosting-versus-random-forest-cfa3fa8f0d80 [6] http://www.liquisearch.com/random_forest/disadvantages [7] https://en.wikipedia.org/wiki/Gradient_boosting#Usage [8] https://www.quora.com/What-are-the-advantages-disadvantages-of-using-Gradient-Boosting-over-Random-Forests [9] https://machinelearningmastery.com/start-with-gradient-boosting/
Creating a Training and Predicting Pipeline	Student successfully implements a pipeline in code that will train and predict on the supervised learning algorithm given.	Implemented in cell 11 of the notebook
Initial Model Evaluation	Student correctly implements three supervised learning models and produces a performance visualization.	Implemented in cell 12 of the notebook.

Improving Results

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Question 3: Choosing the Best Model	Justification is provided for which model appears to be the best to use given computational cost, model performance, and the characteristics of the data.	**Metrics:** From the graphs above the **GradientBoostingClassifier** is performing better on accuracy score and F score on testing data. So, I choose **GradientBoostingClassifier** for improving the model. **Prediction/Training time:** In general **GradientBoostingClassifier** takes longer than other methods and that was evident during training. **Final Algorithm:** I choose **GradientBoostingClassifier** for final training and improvement. I selected this due to two reasons. 1. The model has better scores from tests above 2. To get more intuition on the model works.
Question 4: Describing the Model in Layman's Terms	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.	Gradient boosting model used the decition treses and boosting techniques. 1.First it creates small and weak decision trees. 2.Checks how many outcomes are misclassified.

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		3.It next builds another tree to correct the errors made by previous trees by alternating parameters. 4.The process is continued until we get the error threshold to an acceptable level.
Model Tuning	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.	Model tuning is done in cell 76, 77, 78, 79, 80, 83 and 85. In cells 76, 77, and 78, I tried understand the learning_rate parameter. In cell 79, I added the max_depth parameter. In cells, 76-79 the n_estimators always edged towards the highest value (130). So varied that in cell 80. Once I get the better value for n_estimators, I added the min_sample_split parameter in cell 83.

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
		Finally in cell 85, I reduce the number of parameters to reduce the training time. The training time from cell 83 to cell 85 has reduced significantly.

Question 5:
Final Model Evaluation
	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.	`Unoptimized model`
		
		`------`
		`Accuracy score on testing data: 0.8630`
		`F-score on testing data: 0.7395`
		
		`Optimized Model`
		`------`
		`Final accuracy score on the testing data: 0.8703`
		`Final F-score on the testing data: 0.7518`
		`Took  46.30412173271179 seconds.`

Feature Importance

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Question 6: Feature Relevance Observation	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.	**Answer:** **Education:** The better educated, the better the job and hence the salaries. **Hours-per-week:** The number of hours worked is directly related to the amount of salary gets. So, i think this is an important parameter that will have impact. **Age:** The higher the age, the more experience and better salary **Capital-gain:** If the investments are paying off, people will have more money if not, less money. So capital gains have direct relation on how much an individial earns. **Capital-loss:** Simlar to capital gain, capital loss also has direct impact on people's wealth. So, i believe these 5 are the most important factors that will have an impact.
Question 7: Extracting Feature Importances	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.	**Answer:** My answers were close except for the marital status. Probably the people with working spouses have more money and have better disposable income, which makes sense.

CRITERIA	MEETS SPECIFICATIONS	STUDENT COMMENTS
Question 8: Effects of Feature Selection	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**.	**Answer:** The following are the scores with full and reduced data. **Final Model trained on full data** Accuracy on testing data: 0.8703 F-score on testing data: 0.7518 **Final Model trained on reduced data** Accuracy on testing data: 0.8595 F-score on testing data: 0.7290 The scores on reduced data are lower but not significantly. So, if training time was a factor, it's worth using the reduced data. But in some cases we need to squeeze every percentage point that is possible. In such circumstances, we need to use the model with full data.