Assignment 1

Federico Meini
fmeini3@gatech.edu

Datasets

For the analysis, I chose two different datasets that can be used for classification tasks: Fashion-MNIST from Zalando Research [1] and the Adult Data Set from the UCI Machine Learning Repository [2].

Fashion-MNIST (balanced multi-class classification)

Fashion-MNIST is a dataset of 70000 clothing pictures taken from the Zalando e-commerce website and it is intended as a replacement of the original MNIST dataset of hand-written digits, from which it inherits the image format () and the split between training and test samples.

What makes it interesting

The original MNIST dataset has been used as a benchmark for machine learning models for many years, but some argue that with the increased availability of powerful computing resources and the advent of more advanced algorithms, a more challenging benchmark is necessary. Fashion-MNIST is advertised as a more challenging direct drop-in replacement for the original MNIST database.

Dataset structure

Each sample is a 28x28 pixel grayscale image and is associated with 1 of 10 classes (t-shirt, trouser, pullover, etc.). Each class is associated with an equal number of samples: the dataset perfectly balanced. The goal of the task is to correctly classify test samples. There are 60000 samples for training and 10000 for testing.

I worked with a subset of the dataset in order to obtain faster training times: I only kept samples associated with the first 5 classes (t-shirt, trouser, pullover, dress, coat) and then only kept 20% of the remaining samples (using stratified sampling to keep the classes balanced). That left me with a training dataset of 6000 samples equally distributed across 5 classes and a testing set of 1000 samples equally distributed across 5 classes.

Data cleaning

The data did not need any cleaning: each feature represents the color of one pixel of a 28x28 image and its value is an integer ranging between 0 and 255. Each label is an integer ranging between 0 and 4. There are no missing values in the dataset.

Performance measure

Since the dataset is perfectly balanced, I used *accuracy* [3] as the scoring function for this dataset. Accuracy was always computed as the mean accuracy over 5 folds of cross-validation to ensure that the trained model would generalize well.

Adult Census Data Set (unbalanced binary classification)

The Adult dataset from the UCI Machine Learning Repository, also known as the Census Income dataset, contains data extracted from the 1994 Census database together with a label indicating whether or not a person makes more than 50K dollars a year.

What makes it interesting

Differently from Fashion-MNIST, the Adult dataset contains both categorical and numerical features, each having a different scale. There are missing values and the dataset is *unbalanced*: only about 24% of the recorded persons earn more than 54K per year. Models trained on this kind of data can be used by banks to compute credit scores.

Dataset structure

The dataset, obtained from Kaggle, contains 48842 samples, each of them representing a person and having 14 attributes such as age, sex, working-class, education, etc. Each sample is associated with a label indicating whether or not that person's yearly income is over 50k dollars.

For the purpose of this analysis, I dropped all rows containing missing values to reduce the size of the dataset (and maximize training speed) and have data of better quality. I then divided the dataset (now containing 30162 records) into a training set of 24129 samples and a test set of 6033 samples (20% of the dataset) using stratified sampling to keep the same class balance across training and testing sets.

Data cleaning

I performed the following actions on the dataset:

1. I dropped all records containing missing values.
2. I dropped the redundant feature *education-num.*
3. I used *One Hot Encoding* [4] on all categorical features to turn them into multiple boolean features.
4. I encoded labels into boolean integer values (1 or 0).
5. I scaled all numerical features to the same range using a *min-max scaler* [5] to make sure they all have similar importance.

Performance measure

Since the classes are unbalanced (only 20% of the records have class 1), I decided to use *F1 score* [6] on the positive class (yearly income > 50k) as the performance metric of models trained on this dataset. The *F1 score* is the harmonic mean of *precision* and *recall* and consequently is a much better performance measure than accuracy when there is an uneven distribution (a large number of actual negatives) in the dataset.

The F1 score on the positive class was always computed as the mean F1 score over 5 folds of cross-validation to ensure that the trained model would generalize well.

Decision Trees

Two decision trees were trained respectively on the Fashion-MNIST and the Adult Census datasets. Both models are based on the *DecisionTreeClassifier* implementation from the *Scikit Learn* package.

Initial performance

Learning curves for the two tree classifiers when using the default parameters (and hyper-parameters) showed a very clear variance between learning and cross-validation score. Without performing any tree pruning, in fact, the trees were classifying training data perfectly (overfitting) but were not able to generalize well (thus the low cross-validation score).

Parameter tuning

A few empirical tests showed that pruning the tree by setting a *maximum depth* solved the initial overfit problem and greatly improved cross-validation scores on both datasets. Good values for the most important model parameters and hyper-parameters were found using *Grid Search* and are reported in *Table 1*. Cost-complexity post-pruning was also tested but only revealed useful on the Fashion-MNIST dataset.

(hyper-) parameter	**Fashion-MNIST**	**Adult Census**
criterion	gini	gini
max_depth	20	10
min_samples_split	2	5
min_samples_leaf	5	20
ccp_alpha (post-pruning)	0.001	0

Table 1. Parameters and hyper-parameters values for decision tree classifiers on Fashion-MNIST and Adult Census datasets.

Learning curves

After having tuned parameters and hyper-parameters, learning curves were plotted for both datasets; they are shown in *Figure 1*. On Fashion-MNIST accuracy (both on training and cross-validation sets) is plotted versus the number of training samples, while on Adult Census the F1 score is plotted against the number of training samples.

The model scores a cross-validation accuracy value of 0.82 on Fashion-MNIST and a cross-validation F1 value of 0.67 on the Adult Census dataset.

Bias and variance on Fashion-MNIST

For Fashion-MNIST, the learning curves don't show any particular bias: the training and validation curves are slowly converging while maintaining some margin. There is some evident variance (the high margin between the two curves) but the trend seems to suggest that it would decrease by adding more training samples. I decided not to add more samples to keep training times reasonable.

Bias and variance on Adult Census

For Adult Census, there is no sign of high variance, but both curves tend to plateau at a constant value, which is a sign of high bias. Similar curves were obtained trying different parameter values, which seems to suggest that the model is not complex enough to map the underlying function.

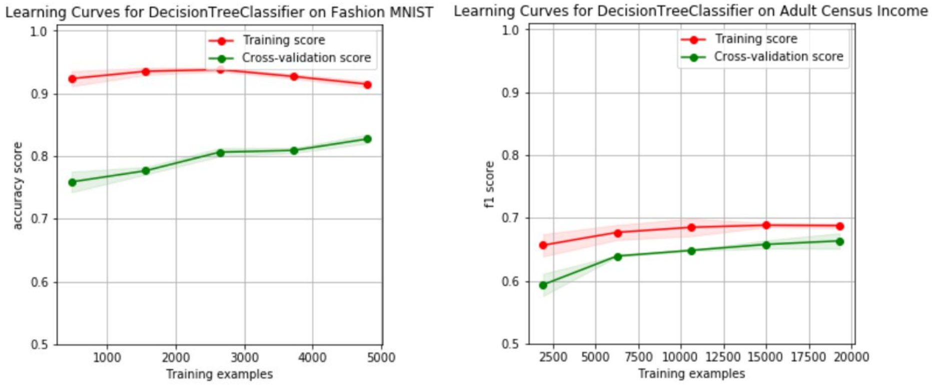

Figure 1. Learning curves for decision tree classifiers (after parameter tuning) on Fashion-MNIST and Adult Census datasets.

Model complexity

Validation curves were plotted to study the correlation between some important model parameters and the score obtained by the model (*accuracy* or *F1*, depending on the dataset), the plots are shown in *Figure 2.* The analysis confirmed the parameter values found by *Grid Search* but, unfortunately, did not provide better ones.

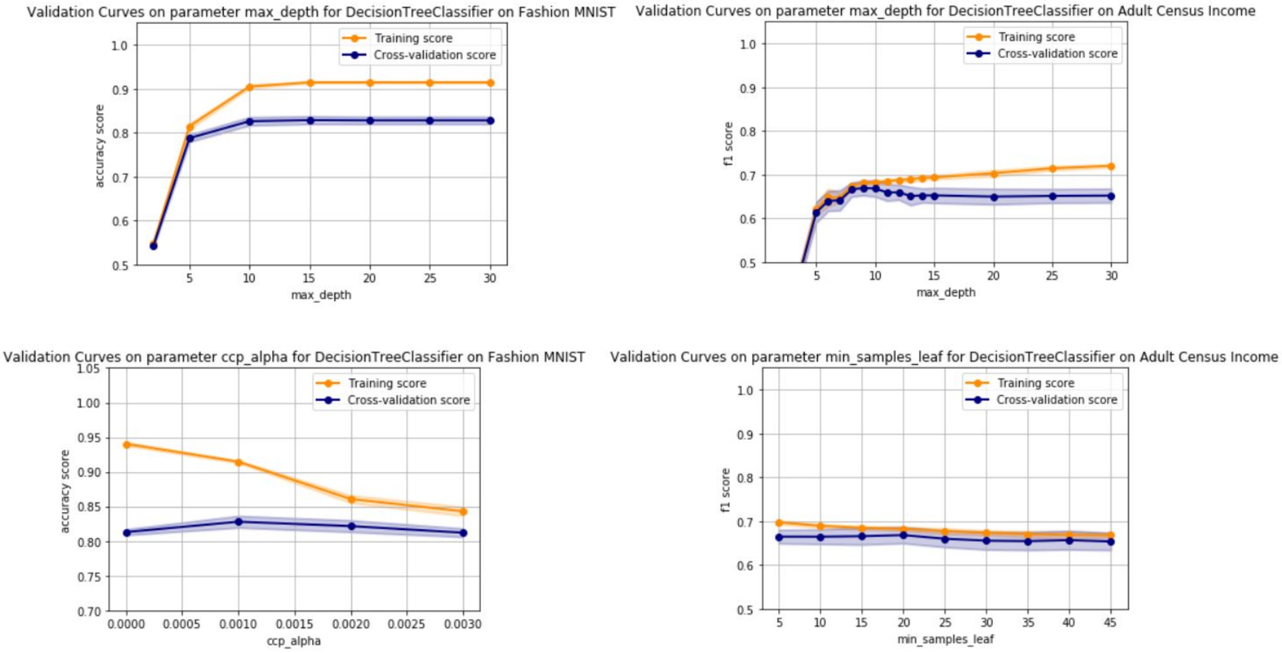

Figure 2. On the left, decision tree validation curves for parameters *max_depth* (tree pruning) and *ccp_alpha* (post pruning) on Fashion-MNIST. On the right, decision tree validation curves for parameters *max_depth* and *min_samples_leaf* on Adult Census.

Performance on the test set

On Fashion-MNIST, the classifier scored an *accuracy* value of 0.81 (versus a cross-validation accuracy value of 0.82), confirming good generalization capabilities of the model.

On Adult Census, the classifier scored an F1 value of 0.66 for the positive class (versus a cross-validation F1 value of 0.67), confirming good generalization capabilities for this model as well.

k-Nearest Neighbors

Two kNN classifiers were trained respectively on the Fashion-MNIST and the Adult Census datasets. Both models are based on the *KNeighborsClassifier* implementation from the *Scikit Learn* package.

Initial performance

Learning curves for the two tree classifiers when using the default parameters (and hyper-parameters) showed some bias and variance (the margin between learning and cross-validation curves) that would remain constant despite adding samples.

Parameter tuning

Good values for the number of neighbors to consider when making predictions were found using *Grid Search* and are reported in *Table 2*.

(hyper-) parameter	Fashion-MNIST	Adult Census
n_neighbors	10	5

Table 2. Parameter values for kNN classifiers on Fashion-MNIST and Adult Census datasets.

Learning curves

After having tuned the *n_neighbors* parameter, learning curves were plotted for both datasets; they are shown in *Figure 3*.

The model scores a cross-validation accuracy value of 0.86 on Fashion-MNIST and a cross-validation F1 value of 0.62 on the Adult Census dataset.

Bias and variance on Fashion-MNIST

On Fashion-MNIST, the learning and cross-validation curves are really close to each other (low variance) but start diverging after 3800 samples. More samples should be added to study the continuation of the curves.

Bias and variance on Adult Census

For Adult Census, both learning and cross-validation curves reach a plateau after 6000 samples and always keep a constant margin (variance) between them. I believe this is an indication that the model is not complex enough to map the underlying function.

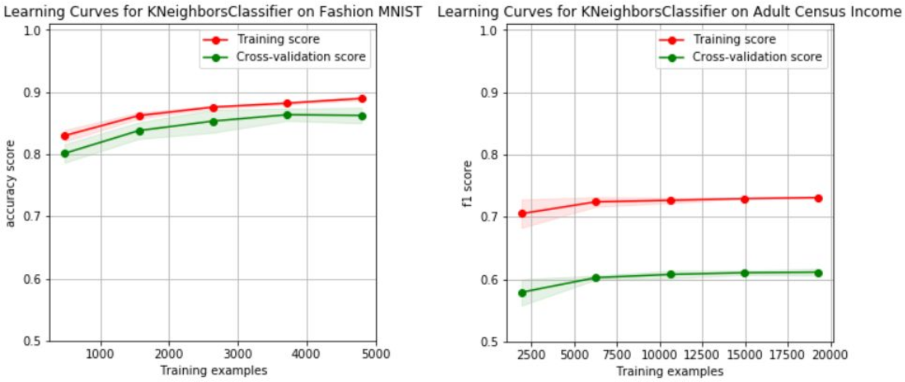

Figure 3. Learning curves for kNN classifiers (after parameter tuning) on Fashion-MNIST and Adult Census datasets.

Model complexity

Validation curves were plotted to study the correlation between the number of nearest neighbors considered and the score obtained by the model (*accuracy* on Fashion-MINST and *F1* on Adult Census), the plots are shown in *Figure 4.* The analysis confirmed the values found with *Grid Search* (*n_neighbors=10* on Fashion-MNIST, and *n_neighbors=5* on Adult Census) but did not provide better ones.

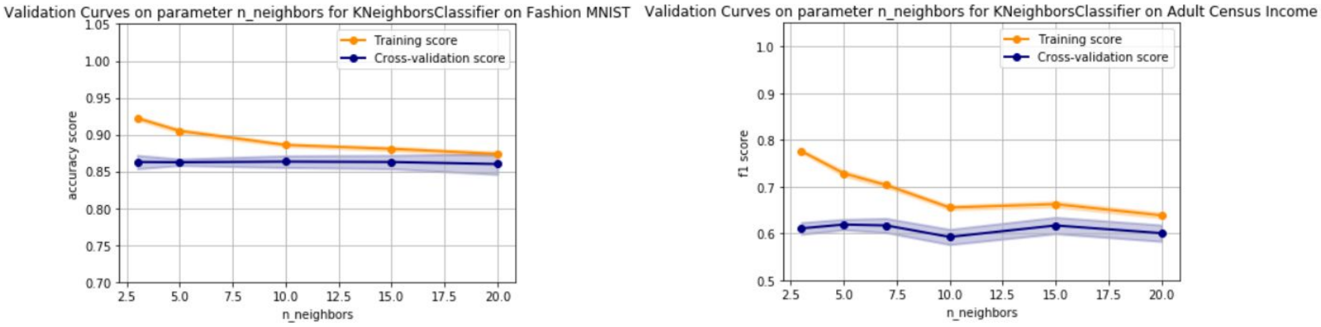

Figure 4. kNN validation curves for parameter *n_neighbors* on Fashion-MNIST and Adult Census.

Performance on the test set

On Fashion-MNIST, the classifier scored an *accuracy* value of 0.84 (versus a cross-validation accuracy value of 0.86), confirming good generalization capabilities of the model.

On Adult Census, the classifier scored an F1 value of 0.63 for the positive class (versus a cross-validation F1 value of 0.62), confirming good generalization capabilities for this model as well.

Support Vector Machines

Two SVM classifiers were trained respectively on the Fashion-MNIST and the Adult Census datasets. Both models are based on the *SVC* implementation from the *Scikit Learn* package.

Initial performance

With the model default parameters learning curves on Fashion-MNIST looked normal, while accuracy on the Adult Census dataset was extremely low. This is because support vector machines are very sensitive to features with different scales: initially, this is what prompted me to normalize all features on the Adult dataset.

Parameter tuning

The following kernels were tried on both datasets: *linear, polynomial, sigmoid* and *RBF (radial basis function).* The best kernel for each dataset and the best values for the *C* regularization parameter were chosen using *Grid Search* and are reported in *Table 3.*

(hyper-) parameter	**Fashion-MNIST**	**Adult Census**
kernel	RBF	poly
gamma	scale	scale

C	3	3

Table 3. Parameter values for SVM classifiers on Fashion-MNIST and Adult Census datasets.

Learning curves

After having chosen the best kernel for each dataset and found a good value for the *C* parameter, learning curves were plotted for both datasets; they are shown in *Figure 5.*

The model scores a cross-validation accuracy value of 0.89 on Fashion-MNIST and a cross-validation F1 value of 0.66 on the Adult Census dataset.

Bias and variance

Learning curves are almost ideal on both datasets: variance between learning and cross-validation curves is not high and diminishes when adding samples; there is no sign of high bias, which could indicate that both models are able to map the underlying function of the datasets.

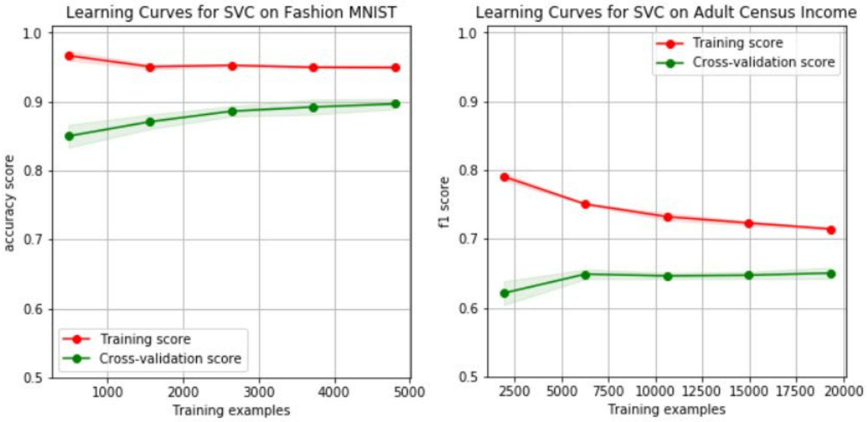

Figure 5. Learning curves for SVM classifiers (after parameter tuning) on Fashion-MNIST and Adult Census datasets.

Model complexity

Validation curves were plotted to study the correlation between the regularization parameter *C* and the score obtained by the model (*accuracy* on Fashion-MINST and *F1* on Adult Census). Since the model trained on the Adult Census uses a polynomial kernel, a validation curve on the grade of the polynomial kernel was also plotted; the plots are shown in *Figure 6.*

The analysis confirmed the *C* parameter value found with *Grid Search* (*C=3*) and provided a good value (*degree=2*) for the grade of the polynomial kernel used for the Adult Census dataset.

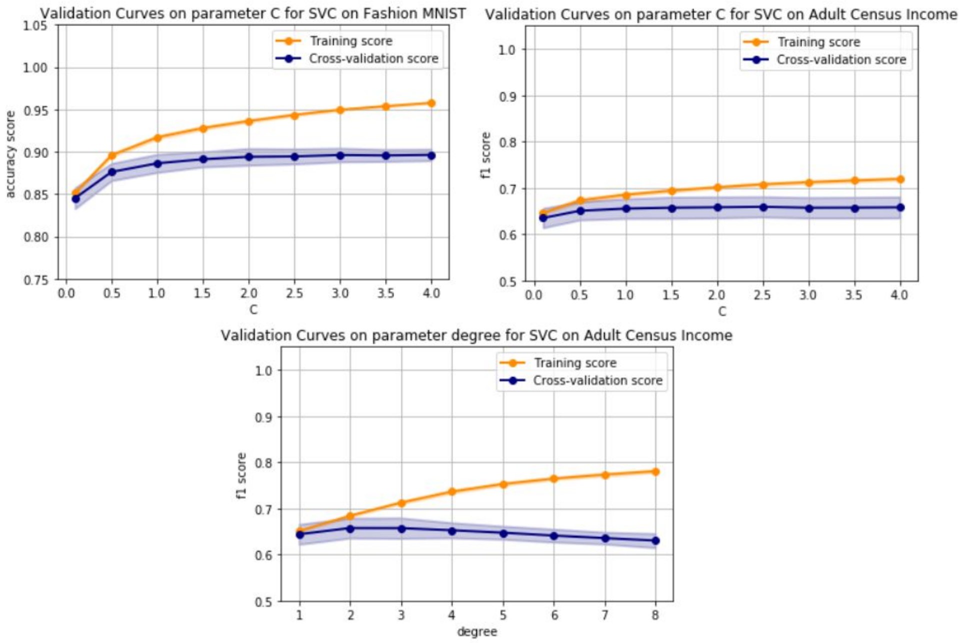

Figure 6. SVM validation curves for parameter *C* on Fashion-MNIST and Adult Census, and for parameter *degree* (polynomial degree) on Adult Census.

Performance on the test set

On Fashion-MNIST, the classifier scored an *accuracy* value of 0.89 (same as the cross-validation accuracy value), confirming good generalization capabilities of the model.

On Adult Census, the classifier scored an F1 value of 0.66 for the positive class (same as the cross-validation F1 value), confirming good generalization capabilities for this model as well.

Boosting

Two Adaboost classifiers were trained respectively on the Fashion-MNIST and the Adult Census datasets. Both models are based on the *AdaboostClassifier* implementation of the Adaptive Boosting algorithm [7] from the *Scikit Learn* package.

Initial performance

I initially chose *weak learners* that were overfitting and that caused the whole Adaboost classifier to overfit the training data (very high variance between the learning curve and the cross-validation curve). The solution to the overfitting problem was to choose a very shallow tree as the *week learner*.

Parameter tuning

Empirical tests and *Grid Search* were used to determine the best *weak learner* to use, how many *weak learners* to use and the best value for the algorithm's *learning rate*. The values are reported in *Table 4.*

(hyper-) parameter	**Fashion-MNIST**	**Adult Census**
base_estimator	Decision tree classifier with *max_depth=3*	Decision tree classifier with *max_depth=3*

n_estimators	20	30
learning_rate	0.3	1.0

Table 4. Parameter values for Adaboost classifiers on Fashion-MNIST and Adult Census datasets.

Learning curves

After having chosen the best parameter values, learning curves were plotted for both datasets; they are shown in *Figure 7.*

The model scores a cross-validation accuracy value of 0.82 on Fashion-MNIST and a cross-validation F1 value of 0.70 on the Adult Census dataset.

Bias and variance

Learning curves look extremely good on both datasets: variance between learning and cross-validation curves is not high and diminishes when adding samples; there is no sign of high bias, which should indicate that both models are able to map the underlying function of the datasets.

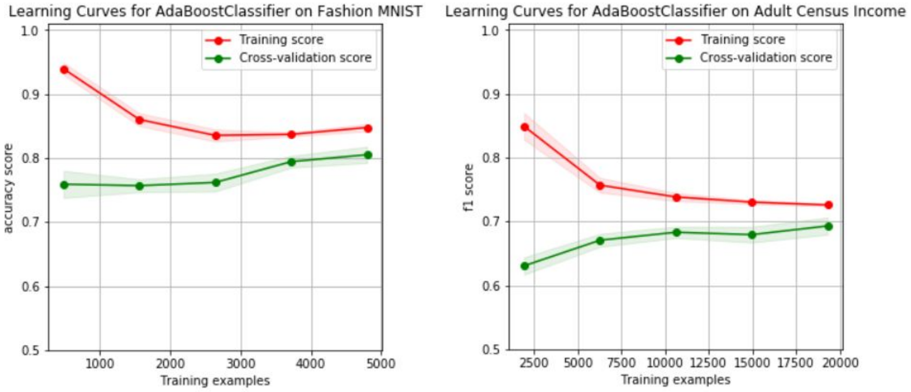

Figure 7. Learning curves for Adaboost classifiers (after parameter tuning) on Fashion-MNIST and Adult Census datasets.

Model complexity

Validation curves were plotted to study the correlation between the number of *weak learners* used (*n_estimators* parameter) and the score obtained by the model (*accuracy* on Fashion-MINST and *F1* on Adult Census). The plots are shown in *Figure 8.*

On Fashion-MNIST, the analysis confirmed the parameter value found with *Grid Search* (*n_estimators=20*), while on the Adult Census dataset it highlighted a better value (*n_estimators=40*) than the one found with *Grid Search.*

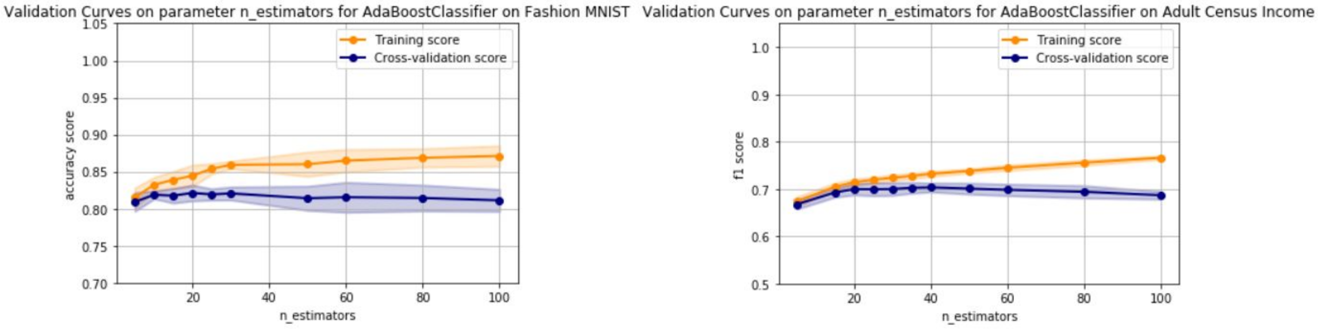

Figure 8. Adaboost validation curves for parameter *n_estimators* on Fashion-MNIST and Adult Census.

Performance on the test set

On Fashion-MNIST, the classifier scored an *accuracy* value of 0.80 (versus a cross-validation accuracy value of 0.82), confirming good generalization capabilities of the model.

On Adult Census, the classifier scored an F1 value of 0.69 for the positive class (versus a cross-validation F1 value of 0.70), confirming good generalization capabilities for this model as well.

Neural Networks

Two Adaboost classifiers were trained respectively on the Fashion-MNIST and the Adult Census datasets. Both models are based on the *AdaboostClassifier* implementation of the Adaptive Boosting algorithm [7] from the *Scikit Learn* package.

Initial performance

I initially chose *weak learners* that were overfitting and that caused the whole Adaboost classifier to overfit the training data (very high variance between the learning curve and the cross-validation curve). The solution to the overfitting problem was to choose a very shallow tree as the *week learner*.

Parameter tuning

Empirical tests and *Grid Search* were used to determine the best *weak learner* to use, how many *weak learners* to use and the best value for the algorithm's *learning rate*. The values are reported in *Table 4*.

(hyper-) parameter	**Fashion-MNIST**	**Adult Census**
base_estimator	Decision tree classifier with *max_depth=3*	Decision tree classifier with *max_depth=3*
n_estimators	20	30
learning_rate	0.3	1.0

Table 4. Parameter values for Adaboost classifiers on Fashion-MNIST and Adult Census datasets.

Learning curves

After having chosen the best parameter values, learning curves were plotted for both datasets; they are shown in *Figure 7*.

The model scores a cross-validation accuracy value of 0.82 on Fashion-MNIST and a cross-validation F1 value of 0.70 on the Adult Census dataset.

Bias and variance

Learning curves look extremely good on both datasets: variance between learning and cross-validation curves is not high and diminishes when adding samples; there is no sign of high bias, which should indicate that both models are able to map the underlying function of the datasets.

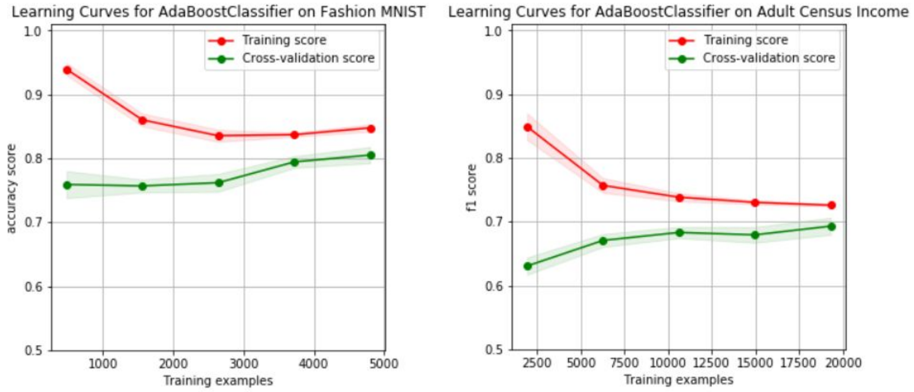

Figure 7. Learning curves for Adaboost classifiers (after parameter tuning) on Fashion-MNIST and Adult Census datasets.

Model complexity

Validation curves were plotted to study the correlation between the number of *weak learners* used (*n_estimators* parameter) and the score obtained by the model (*accuracy* on Fashion-MINST and *F1* on Adult Census). The plots are shown in *Figure 8*.

On Fashion-MNIST, the analysis confirmed the parameter value found with *Grid Search* (*n_estimators=20*), while on the Adult Census dataset it highlighted a better value (*n_estimators=40*) than the one found with *Grid Search*.

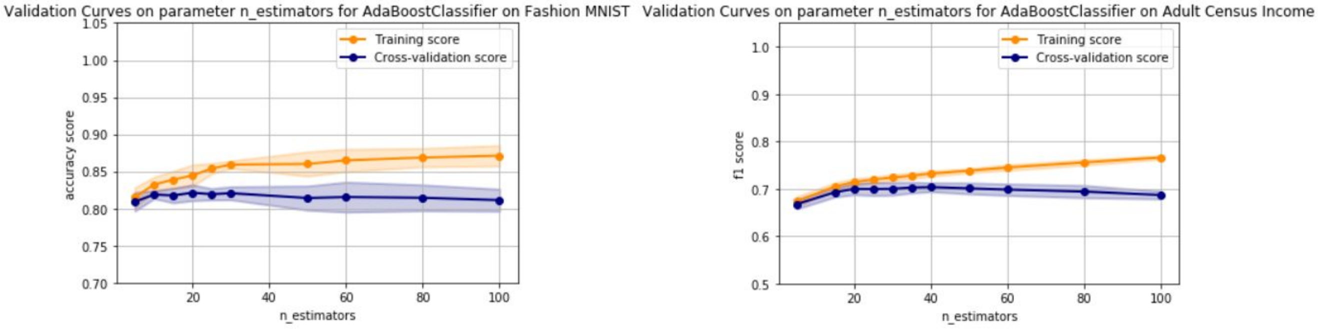

Figure 8. Adaboost validation curves for parameter *n_estimators* on Fashion-MNIST and Adult Census.

Performance on the test set

On Fashion-MNIST, the classifier scored an *accuracy* value of 0.80 (versus a cross-validation accuracy value of 0.82), confirming good generalization capabilities of the model.

On Adult Census, the classifier scored an F1 value of 0.69 for the positive class (versus a cross-validation F1 value of 0.70), confirming good generalization capabilities for this model as well.

References

1. Xiao, et al. “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms.” *ArXiv.org*, 15 Sept. 2017, arxiv.org/abs/1708.07747.
2. Kohavi. “Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid.” Scaling up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid (Conference) | OSTI.GOV, AAAI Press, Menlo Park, CA (United States), 31 Dec. 1996, www.osti.gov/biblio/421279-scaling-up-accuracy-naive-bayes-classifiers-decision-tree-hybrid.
3. “Classification: Accuracy | Machine Learning Crash Course.” *Google Machine Learning Crash Course*, Google, developers.google.com/machine-learning/crash-course/classification/accuracy.
4. “Sklearn.preprocessing.OneHotEncoder.” *Scikit Learn*, scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html.
5. “Sklearn.preprocessing.MinMaxScaler.” *Scikit*, scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html.
6. “F1 Score.” *Wikipedia*, Wikimedia Foundation, 30 Jan. 2020, en.wikipedia.org/wiki/F1_score.
7. “AdaBoost.” *Wikipedia*, Wikimedia Foundation, 4 Jan. 2020, en.wikipedia.org/wiki/AdaBoost.