Homework 2 submission

ECET 612 — Applied Machine Learning

Murat Isik

May 6, 2022

1 Submitted files

For this assignment, besides this report, the following archives were created:

1.1 SRC Folder

* *" LogisticRegression"*: This **function** estimates the parameters of a logistic model which are the coefficients in the linear combination.
* *" confusion_matrix"*: This **function** specifies a table layout that allows visualization of the performance of an algorithm, typically a supervised learning one.
* *" KNeighborsClassifier"*: This **function** classifies represents the k nearest neighbors, where k is an integer value specified by the user.
* *" RandomForestClassifier"*: This **function** estimates that fit a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting
* *" VotingClassifier"*: This **function** trains on an ensemble of numerous models and predicts an output (class) based on the highest probability of chosen class as the output.

* "+ ***"code2.ipyb"***: This **script** initiates all the necessary function calls, and also generates the graphs for housing prices. First, it calculates with the information provided by the X_training and Y_training files. Then it finds the plot of the graphs by using the *different python* functions. Afterward, it creates graph models.

2 Discussion

In this project, we construct an ensemble learner developed from any of the models covered except a neural network for classifying digits from a modified MNIST dataset. In contrast to a statistical ensemble in statistical mechanics, which is normally unlimited, a machine learning ensemble consists of just a specific finite number of different models, but typically allows for a far more flexible structure to exist among those alternatives. Evaluating an ensemble's prediction often needs more computing than evaluating a single model's forecast. In one sense, ensemble learning may be viewed as a method of compensating for bad learning algorithms by executing a large amount of extra computation. The alternative, on the other hand, is to undertake a lot more learning on a single non-ensemble system. An ensemble system may be more efficient in improving overall accuracy for the same increase in computation, storage, or communication resources by distributing that increase among two or more techniques than a single method would have been improved by increasing resource consumption. Fast algorithms, such as decision trees, are frequently employed in ensemble methods (for example, random forests), but slower algorithms can also benefit from ensemble approaches. As in previous works, we provide the training dataset while holding back the test set for evaluation of our model.

3. Results

3.1 Dataset and Classification

The given training dataset is comparable to the MNIST dataset, but with some noise applied to each instance, as illustrated in Figure 1's digit plot. As a result, each instance must be projected into a smaller dimension so that superfluous characteristics may be ignored. The model can obtain greater performance in a shorter runtime by applying PCA to each instance. As a result, within each individual learner of the model, the instances are projected into a reduced dimensional space using PCA, then scaled using StandardScale. Following that, the examples are categorized into classes, and the classifications are compared to the right labels to evaluate the classifiers' performance.

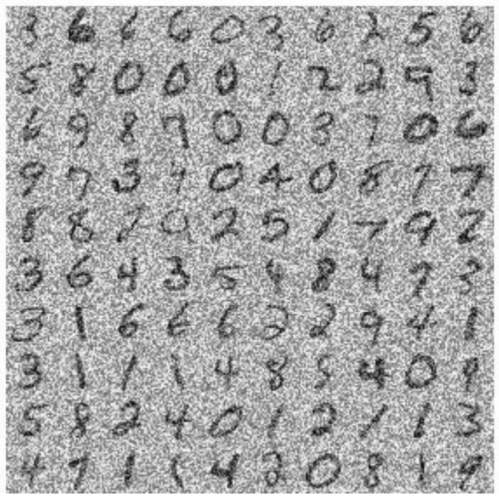

Figure 1: Plot of given digits in the training dataset.

3.2 Individual learners

i. Softmax Regression

Softmax Regression is the model's initial learner. It is generated by Scikit-Logistic Learn's Regression model, with the hyperparameter multi-class set to "multinomial." The intensity of the l2 regularization is controlled by the hyperparameter C, which is set at 20 to maximize the classifier.

```
logreg_clf = Pipeline([
    ("PCA", PCA(n_components=0.2)),
    ("scaler", StandardScaler()),
    ("logisticRegression",
    LogisticRegression(
            multi_class="multinomial",
            solver="lbfgs", C=20, random_state=42,
            max_iter=60000))
])
```

ii. KNN

The second learner of the model is K-Nearest Neighbors (KNN). With n neighbors set to 4 and weights set to 'distance,' the learner classifies one instance by examining its four nearest neighbors, with the weight of each neighbor being inversely proportionate to its distance from the classed one. Similar to Softmax Regression, a PCA and a

```
knn_clf = Pipeline([
    ("PCA",
    PCA(n_components=0.2)),
    ("scaler", StandardScaler()),
    ("knn", KNeighborsClassifier(weights="distance", n_neighbors =
    4)),
])
```

A PCA and a Standard Scaler are pipelined prior to classification, similar to Softmax Regression, with only 20% of the features evaluated for classifying.

iii. SVM

The model's final learner is a Nonlinear SVM classifier. Due to time constraints, the GridSearchCV was not run on this model to determine the best hyperparameters for classifying this dataset. As a result, they are left as default settings.

```
svm_clf = Pipeline([
    ("PCA",
    PCA(n_components=0.2)),
    ("scaler", StandardScaler()),
    ("knn", SVC()),
])
```

3.3 Ensemble model

As three independent learners are assembled, they are combined into a single model, with each learner having a vote when categorizing an instance. The final categorization is determined by a majority of votes, which improves overall accuracy. The hyperparameter voting is set to 'hard,' such that votes are counted based on the labels predicted by each learner rather than the cumulative probabilities of each class.

```
voting_clf = VotingClassifier(
    estimators=[(‘lr’, logreg_clf), (‘svc’, svm_clf), (‘knn’,
    knn_clf)], voting=’hard’
)
```

3.4 Performance

iv. Cross-validationtion

Individual learners and the ensemble model are subjected to cross-validation, and learning curves are shown to visualize train and validation errors. Table 1 shows the accuracy gained throughout the cross-validation procedure, and the learning curves are provided below.

	1st Section	2nd Section	3rd Section	4th Section	5th Section
Softmax	0.8441	0.8367	0.8448	0.8325	0.8368
KNN	0.9284	0.9260	0.9263	0.9271	0.9312
SVM	0.9542	0.9489	0.9515	0.9504	0.9523
Ensemble model	0.9423	0.9371	0.9389	0.9369	0.9415

e 1: Cross Validation Accuracies

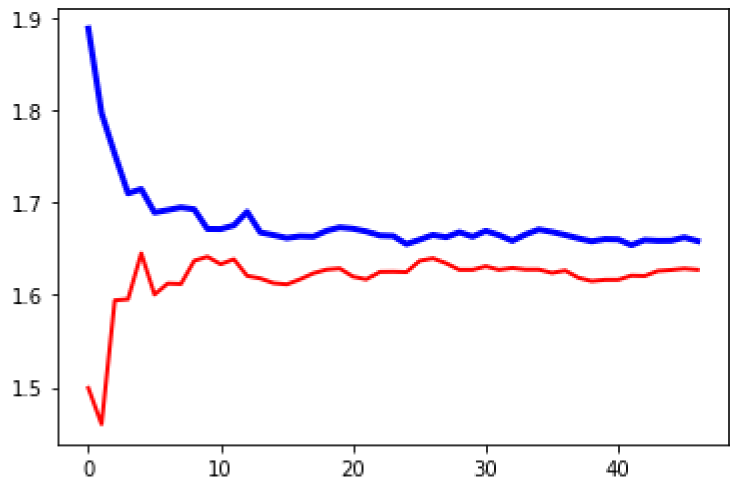

Figure 2: Softmax Learning Curve

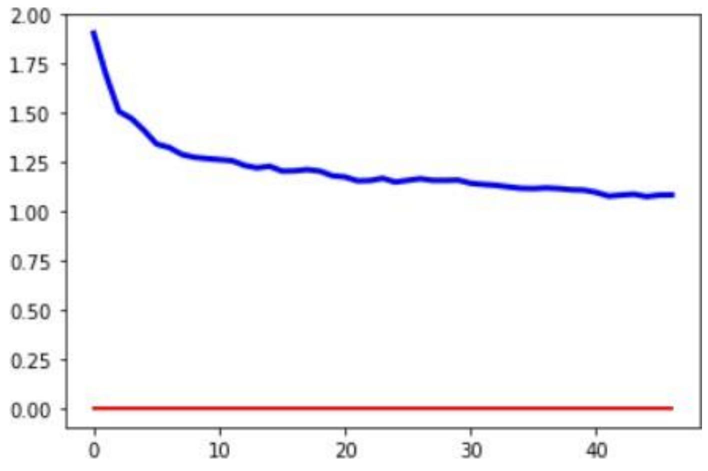

Figure 3: KNN Learning Curve

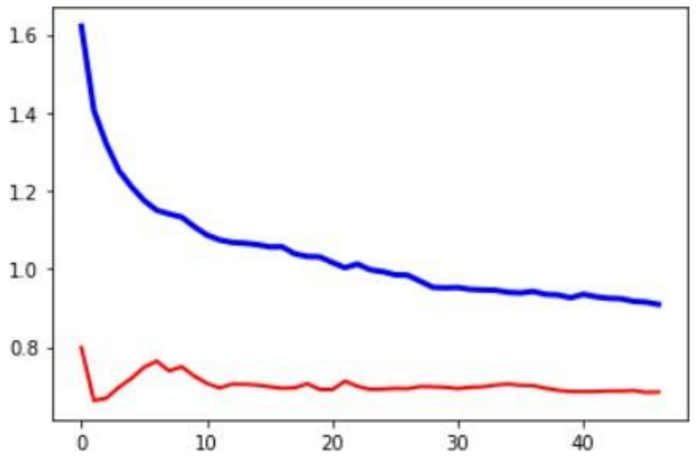

Figure 4: SVC Learning Curve

v. F1 Score

The model's performance is measured by the F1 score, which is the harmonic mean of Precision and Recall. With an F1 score of 0.9393, the model is demonstrated to be quite accurate, with Precision and Recall at roughly the same level of accuracy. Cross-validation accuracy for the ensemble model is also 0.94, demonstrating that the model is capable of classifying handwritten digits with high accuracy.

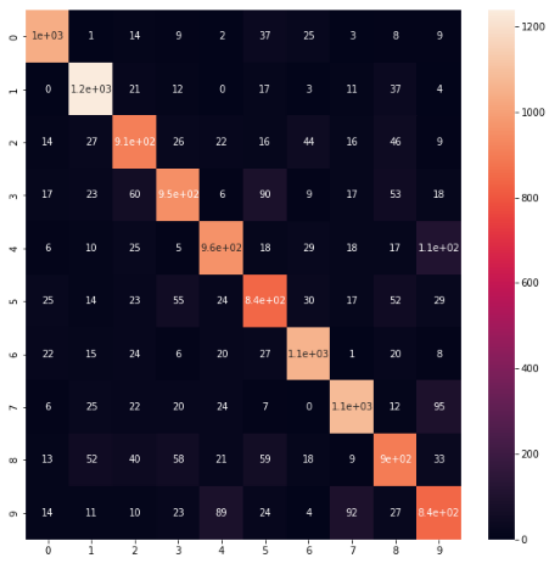

Figure Confusion Matrix

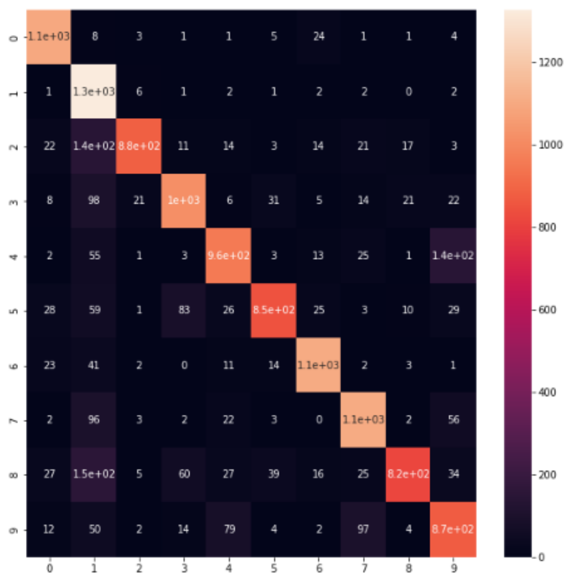

Figure KNeighborsClassifier

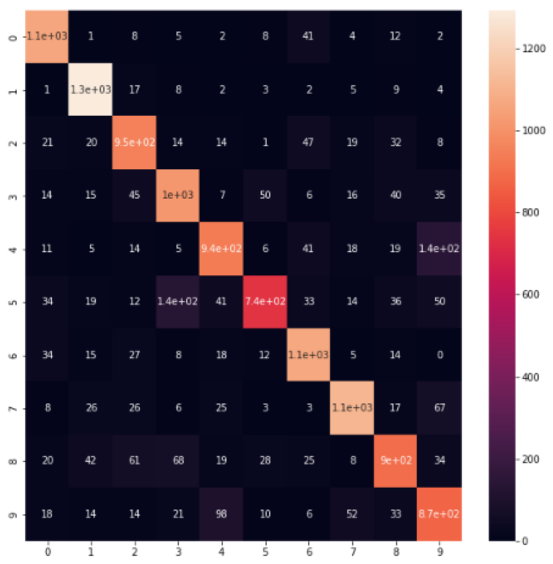

Figure RandomForestClassifier

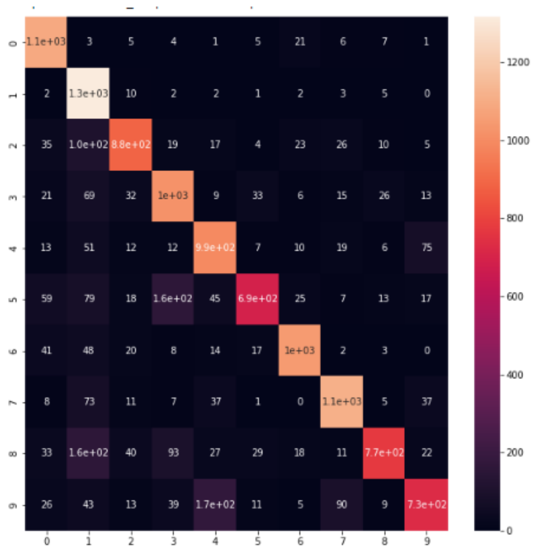

Figure VotingClassifier