Practice Exam - Applied Machine Learning COMS W4995

Date:

Name:

UNI:

1 True/False

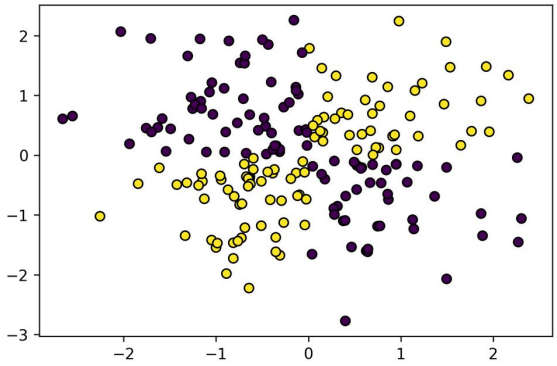
	True	False
A fast-forward git merge creates a new commit.		
np.random.uniform() == np.random.uniform() evaluates to True.		
Using cross-validation avoids overfitting when tuning parameters.		
The sign of a particular coefficients in ridge regression will be the same, no matter what the regularization parameter.		
Stochastic gradient descent is suitable for datasets with a very high number of samples.		
It is good practice to standardize sparse dataset so that each feature has zero mean.		
A node in a decision tree always contains exactly half the samples of its parent.		
Support vector machines don't scale well to large datasets.		
Decision Trees are very sensitive to the scaling of the data.		
For a perfectly calibrated classifier, 80% of the data for which p(y=1) =.8 belong to class 1.		

2 Multiple choice

2.1 Which of the following are non-parametric models?

- ❏ Random Forest
- ❏ Linear Regression
- ❏ Logistic Regression
- ❏ Nearest Neighbors
- ❏ Nearest Shrunken Centroid

2.2 Given a two-class classification dataset with the two features shown below and additional non-informative features, which of the following feature selection methods would be able to identify the these two features as informative?



- ❏ SelectPercentile(f_classif)
- ❏ SelectKBest(mutual_info)
- ❏ SelectFromModel(DecisionTreeClassifier())
- ❏ SequentialFeatureSelector(SVC(kernel='rbf'))
- ❏ RFE(LogisticRegression())

2.3 Which of the following algorithms provide feature importances or coefficients and can be used with SelectFromModel?

- ❏ SVC(kernel="rbf")
- ❏ GradientBoostingRegressor
- ❏ KNeighborsClassifier
- ❏ LinearSVC

2.4 Which of the following transformations allow linear classifiers to learn non-linear decision boundaries?

- ❏ RobustScaler()
- ❏ PolynomialFeatures(degree=2)
- ❏ RBFSampler()
- ❏ SelectFromModel(DecisionTreeClassifier())

3 Debugging

For each code snippet, find and explain all errors given the task.

3.1 Task: Use cross-validation to assess how well feature selection and Random forest will do on the test set.

```
select = SelectPercentile(percentile=50).fit(X_train, y_train)
X_train_selected = select.transform(X_train)
scores = cross_val_score(RandomForestClassifier(n_estimators=100),
                                    X_train_selected, y_train, cv=10)
```

3.2 Task: Use the Box-Cox transform to preprocess data and learn a Ridge model, and visualize the coefficients. Assume that BoxCox is a scikit-learn implementation of the Box-Cox transform.

```
pipe = make_pipeline(StandardScaler(), BoxCox(), Ridge())
scores = cross_val_score(pipe, X_train, y_train, n_folds=10)
plt.barh(range(X_train.shape[0]), pipe.coef_)
```

4 Coding

Provide code to build a `LogisticRegression` model and evaluate its performance on a separate test set, given a classification dataset as numpy arrays X and y.

Provide code to implement grid-searching the parameters C and gamma of an SVC in a pipeline with a StandardScaler, and evaluating the best parameter setting on a separate test set, given data as numpy arrays X and y.

5 Concepts

Answer each question with a short (2-5 sentences) explanation.

5.1 How are the “jet” and “viridis” colormaps different and why does it matter?

5.2 Explain the difference between Logistic regression and linear SVMs.

5.3 Explain the basic idea of the RANSAC algorithm.

5.4 When should you use the Box-Cox transformation?