← Go Back to Ensemble Techniques

Course Content

Hands-On Quiz

Type	:	Practice Quiz
Attempts	:	1/1
Questions	:	10
Time	:	2h
Your Marks	:	18/20

Instructions

Attempt History

Attempt #1
Jan 08, 9:55 PM

Marks: 18

Q No: 1 Correct Answer

Marks: 2/2

Load the dataset and identify the variables that have more than 0.7 correlation with the dependent variable Opt_Service

A) WorkExp
B) Salary
C) MBA
D) Age

- Only A
- Both A and B — You Selected
- Both A and D
- A, B, C, and D

```
plt.figure(figsize=(10,7))
sns.heatmap(df.corr()[df.corr()>0.7],annot = True,vmax = 1, vmin = -1, cmap=
```

Q No: 2 — Correct Answer

Marks: 2/2

Create dummies for the Gender column. Split the data into a 70:30 ratio. What is the percentage of 0 and 1 classes in the test data (y_test)?

Note - Do not use stratify on the dataset.

0: In a range of 0.8 to 0.95

1: In a range of 0 to 0.1

You Selected

0: In a range of 0.7 to 0.8

1 In a range of 0.2 to 0.3

0: In a range of 0.5 to 0.6

1: In a range of 0.3 to 0.6

0: In a range of 0.1 to 0.2

1: In a range of 0.3 to 0.5

```
# Creating X and Y variables
X = df.drop(['Opt_service'],axis=1)
y = df['Opt_service']
```

```
# creating dummies for Gender
X= pd.get_dummies(X, columns = ['Gender'],drop_first = True)
```

```
# Splitting data into training and test set:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, ra
print(X_train.shape, X_test.shape)
```

```
#checking percentage of classes in y_test
print(y_test.value_counts(1))
```

Q No: 3

Incorrect Answer

Marks: 0/2

Build a bagging classifier with default parameters. How many employees who would take the service are correctly identified by the model from the training data?

in a range of 10 - 20

in a range of 20 - 30 Correct Option

in a range of 30 - 40 You Selected

in a range of 40 - 50

```
# fitting the baggin classifier on training set
bag = BaggingClassifier(random_state=1)
bag.fit(X_train, y_train)

# predicting on training set
y_train_pred = bag.predict(X_train)
```

```
# creating the confusion matrix
sns.heatmap(confusion_matrix(y_train,y_train_pred),annot=True,fmt='.1f')
plt.xlabel('Predicted values')
plt.ylabel('Actual values')
plt.show()
```

Q No: 4 Correct Answer

Marks: 2/2

Build a random forest classifier with default parameters and a bagging classifier with logistic regression as the base estimator.

Select which of the following statements is/are true?

A) Random forest is giving the same recall as bagging classifier on the train data
B) Random forest is giving a higher recall than bagging classifier on the train data
C) Random forest is giving a higher f1_score than bagging classifier on the train data
D) Random forest is giving a lower f1_score than bagging classifier on the train data

A and D

C and D

A and C

B and C You Selected

```
# Random Forest
rf = RandomForestClassifier(random_state=1)
rf.fit(X_train, y_train)
```

```
# bagging classifier with base estimator  =Logistic Regression
bag_lr = BaggingClassifier(random_state=1, base_estimator = LogisticRegress
bag_lr.fit(X_train, y_train)
```

```
# predicting on the training set
rf_predict = rf.predict(X_train)
bag_predict = bag_lr.predict(X_train)
```

```
# Performance of Random Forest
```

```
print('Random Forest')
```

```
print(recall_score(y_train, rf_predict))
print(f1_score(y_train, rf_predict))
```

```
# Performance of Bagging classifier
```

```
print('Bagging')
```

```
print(recall_score(y_train, bag_predict))
print(f1_score(y_train, bag_predict))
```

Q No: 5 Correct Answer

Marks: 2/2

Build a bagging classifier model with the base estimator as a decision tree.

Vary the depth of the base estimator/Decision tree from depth 1 to 5 (both values included) and compare their performance.

Select which of the following statements is true?

A) At depth = 1, the bagging classifier gives the highest f1 score among all the models on the training set.
B) At depth = 2, the bagging classifier gives the highest f1 score among all the models on the training set.
C) At depth = 4, the bagging classifier gives the highest f1 score among all the models on the training set.
D) At depth = 5, the bagging classifier gives the highest f1 score among all the models on the training set.

- ○ Only A
- ○ Only B
- ○ Only C
- ◉ Only D — You Selected

```
scores = []
for i in range(1,6):
 bag = BaggingClassifier(base_estimator=DecisionTreeClassifier(random_state=
 bag.fit(X_train, y_train)
 pred = bag.predict(X_train)
 case = {'Depth':i,'F1 Score':f1_score(y_train,pred)}
 scores.append(case)
```

```
print(scores)
```

Q No: 6 Correct Answer

Marks: 2/2

Which of the boosting models (AdaBoost, GradientBoost, XGBoost) with default parameters has the lowest f1-score on the training set?

Note: Set the eval_metric = 'logloss' for XGBoostClassifier

- AdaBoost
- Gradient boosting
- Xgboost
- All three models have the same f1 score — You Selected

```
# Adaboost classifier
abc = AdaBoostClassifier(random_state=1)
abc.fit(X_train, y_train)
```

```
# GradientBoost classifier
gbm = GradientBoostingClassifier(random_state=1)
gbm.fit(X_train, y_train)
```

```
#XGBoost classifier
xgb = XGBClassifier(random_state=1, eval_metric = 'logloss')
xgb.fit(X_train, y_train)
```

```
# predicting on training set
abc_predict = abc.predict(X_train)
gbm_predict = gbm.predict(X_train)
xgb_predict = xgb.predict(X_train)
```

```
# Checking model performance
print('ABC')
print(f1_score(y_train, abc_predict))
```

```
print('GBM')
print(f1_score(y_train, gbm_predict))
```

```
print('XGB')
print(f1_score(y_train, xgb_predict))
```

Q No: 7 Correct Answer

Marks: 2/2

Use the gradient boosting classifier trained in Q6 and plot the feature importance of the variables. Which are the 2 most important variables respectively?

- ○ MBA and Engineer
- ◉ Salary and Distance — You Selected
- ○ Age and WorkExp
- ○ License and Gender_Male

```
# Gradient boosting model
gbm = GradientBoostingClassifier(random_state = 1)
gbm.fit(X_train, y_train)
```

```
# visualizing feature importance
feature_names = X_train.columns
importances = gbm.feature_importances_
indices = np.argsort(importances)
plt.figure(figsize=(12, 12))
plt.title("Feature Importances")
plt.barh(range(len(indices)), importances[indices], color="violet", align="
plt.yticks(range(len(indices)), [feature_names[i] for i in indices])
plt.xlabel("Relative Importance")
plt.show()
```

Q No: 8 Correct Answer

Marks: 2/2

Train three models:

1. Model1 = Gradient Boosting classifier with n_estimator = 50 and learning rate = 0.01
2. Model2 = Gradient Boosting classifier with n_estimator = 100 and learning rate = 0.01
3. Model3 = Gradient Boosting classifier with n_estimator = 400 and learning rate = 0.01

f1_score1, **f1_score2**, **f1_score3** are f1_scores of the three models respectively.

What is the order of f1_score on the training set?

f1_score1 > f1_score2 > f1_score3

f1_score2 > f1_score1 > f1_score3

f1_score3 > f1_score2 > f1_score1 You Selected

f1_score3 > f1_score1 > f1_score2

```
# Training first model
gbm1 = GradientBoostingClassifier(random_state=1, n_estimators=50, learning_
gbm1.fit(X_train,y_train)

# Training second model
gbm2 = GradientBoostingClassifier(random_state=1, n_estimators=100, learnin
gbm2.fit(X_train,y_train)
```

```
# Training third model
gbm3 = GradientBoostingClassifier(random_state=1, n_estimators=400, learnin
gbm3.fit(X_train,y_train)
```

```
# Checking model performance
print('First Model Result')
y_pred = gbm1.predict(X_train)
print(f1_score(y_train,y_pred))
```

```
print('Second Model Result')
y_pred = gbm2.predict(X_train)
print(f1_score(y_train,y_pred))
```

```
print('Third Model Result')
y_pred = gbm3.predict(X_train)
print(f1_score(y_train,y_pred))
```

Q No: 9 Correct Answer

Marks: 2/2

Build a stacking classifier using two models - Decision Tree, Bagging Classifier as base estimators and use Random Forest as the final estimator. Which of the following is/are true?

Select which of the following statements is/are true?

A) Stacking classifier is giving f1 score in the range of 0.7 - 0.9 on the training set

B) Stacking classifier is giving recall score in the range of 0.8 - 1 on the training set

C) Stacking classifier is giving f1 score in the range of 0.5 - 0.7 on the training set

D) Stacking classifier is giving recall score in the range of 0.65 - 0.75 on the training set

- (●) A and B — You Selected
- ( ) B and C
- ( ) C and D
- ( ) A and D

```
# creating list of estimators
estimators = [('DT',DecisionTreeClassifier(random_state=1)),
 ('BAG',BaggingClassifier(random_state=1))]
```

```
# Training stacking classifier
clf = StackingClassifier(estimators = estimators, final_estimator = RandomF
clf.fit(X_train, y_train)
```

```
# Predicting on training set
y_pred = clf.predict(X_train)
```

```
# Checking model performance
print('F1 score')
f1_score(y_train,y_pred)
```

```
print('Recall score')
recall_score(y_train,y_pred)
```

Previous

Next