Assignment3 (Score: 4.0 / 5.0)

1. Test cell (Score: 1.0 / 1.0)
2. Test cell (Score: 1.0 / 1.0)
3. Test cell (Score: 1.0 / 1.0)
4. Test cell (Score: 1.0 / 1.0)
5. Test cell (Score: 0.0 / 1.0)
6. Test cell (Score: 0.0 / 0.0)

*You are currently looking at **version 0.1** of this notebook. To download notebooks and datafiles, as well as get help on Jupyter notebooks in the Coursera platform, visit the Jupyter Notebook FAQ course resource.*

In [1]:

```
import numpy as np
import pandas as pd
```

Question 1¶

Import the data from `assets/fraud_data.csv` . What percentage of the observations in the dataset are instances of fr

This function should return a float between 0 and 1.

In [2]:

Student's answer

```
def answer_one():

    # Your code here
    df = pd.read_csv('assets/fraud_data.csv')

    return df['Class'].sum()/len(df['Class'])
answer_one()
```

Out[2]:

```
0.016410823768035772
```

In [3]:

Grade cell: `cell-09b987c4d8138e24` Score: 1.0 / 1.0

In [ ]:

In [4]:

```
# Use X_train, X_test, y_train, y_test for all of the following questions
from sklearn.model_selection import train_test_split

df = pd.read_csv('assets/fraud_data.csv')

X = df.iloc[:,:-1]
y = df.iloc[:,-1]

X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
```

In [ ]:

```

```

Question 2¶

Using `X_train`, `X_test`, `y_train`, and `y_test` (as defined above), train a dummy classifier that classifies everything the majority class of the training data. What is the accuracy of this classifier? What is the recall?

This function should a return a tuple with two floats, i.e. `(accuracy score, recall score)`.

In [5]:

Student's answer

```
def answer_two():
    from sklearn.dummy import DummyClassifier
    from sklearn.metrics import recall_score

    # Your code here
    clf = DummyClassifier()
    clf.fit(X_train, y_train)
    predictions = clf.predict(X_test)
    score = clf.score(X_test, y_test)
    recall_score = recall_score(y_test, predictions)

    return score, recall_score # Return your answer
answer_two()
```

Out[5]:

```
(0.9852507374631269, 0.0)
```

In [6]:

Grade cell: `cell-a901c7f5cfea1a8c` Score: 1.0 / 1.0

```

```

Question 3¶

Using X_train, X_test, y_train, y_test (as defined above), train a SVC classifer using the default parameters. What is the accuracy, recall, and precision of this classifier?

This function should a return a tuple with three floats, i.e. `(accuracy score, recall score, precision score)`.

In [7]:

Student's answer

```
def answer_three():
    from sklearn.metrics import recall_score, precision_score, accuracy_score
    from sklearn.svm import SVC
    model = SVC().fit(X_train,y_train)
    y_pred = model.predict(X_test)
    accuracy = accuracy_score(y_test,y_pred)
    recall = recall_score(y_test,y_pred)
    precision = precision_score(y_test,y_pred)
    # Your code here

    return (accuracy,recall,precision) # Return your answer
answer_three()
```

Out[7]:

```
(0.9900442477876106, 0.35, 0.9333333333333333)
```

In [8]:

Grade cell: `cell-30a8c78257c28475` Score: 1.0 / 1.0

Question 4¶

Using the SVC classifier with parameters `{'C': 1e9, 'gamma': 1e-07}`, what is the confusion matrix when using a threshold of -220 on the decision function. Use X_test and y_test.

This function should return a confusion matrix, a 2x2 numpy array with 4 integers.

In [9]:

Student's answer

```
def answer_four():
    from sklearn.metrics import confusion_matrix
    from sklearn.svm import SVC

    # Your code here
    clf = SVC(C=1e9, gamma=1e-07)
    clf.fit(X_train, y_train)
    y_scores = clf.decision_function(X_test) > -220
    confusion_matrix = confusion_matrix(y_test, y_scores)

    return confusion_matrix # Return your answer



answer_four()
```

Out[9]:

```
array([[5320,   24],
       [  14,   66]])
```

In [10]:

Grade cell: `cell-d10afc8717f94586` Score: 1.0 / 1.0

Question 5¶

Train a logisitic regression classifier with default parameters using X_train and y_train.

For the logisitic regression classifier, create a precision recall curve and a roc curve using y_test and the probability estimates for X_test (probability it is fraud).

Looking at the precision recall curve, what is the recall when the precision is `0.75`?

Looking at the roc curve, what is the true positive rate when the false positive rate is `0.16`?

This function should return a tuple with two floats, i.e. `(recall, true positive rate)`.

In [11]:

Student's answer

```
def answer_five():
    import numpy as np
    from sklearn.linear_model import LogisticRegression
    from sklearn.metrics import precision_recall_curve, roc_curve

    # Train Logistic Regression with the solver 'liblinear'
    lr = LogisticRegression(solver='liblinear').fit(X_train, y_train)

    # Get probability estimates for X_test (using predict_proba for positive class probabil
    y_scores = lr.predict_proba(X_test)[:, 1]

    # Precision-Recall Curve
    precision, recall, _ = precision_recall_curve(y_test, y_scores)

    # Find the recall where precision is closest to 0.75
    precision_diff = np.abs(precision - 0.75)
    idx_recall_at_precision = np.argmin(precision_diff)
    recall_at_precision = recall[idx_recall_at_precision]

    # ROC Curve
    fpr, tpr, _ = roc_curve(y_test, y_scores)

    # Find the true positive rate when the false positive rate is closest to 0.16
    fpr_diff = np.abs(fpr - 0.16)
    idx_fpr = np.argmin(fpr_diff)
    tpr_at_fpr = tpr[idx_fpr]

    # Return the recall at precision=0.75 and the TPR at FPR=0.16
    return recall_at_precision, tpr_at_fpr
recall_value, tpr_value = answer_five()
print(f"Recall at precision 0.75: {recall_value}")
print(f"True Positive Rate at FPR 0.16: {tpr_value}")
```

```
Recall at precision 0.75: 0.825
True Positive Rate at FPR 0.16: 0.9375
```

In [12]:

Grade cell: `cell-17abc112ffe76f05` Score: 0.0 / 1.0

```
You have failed this test due to an error. The traceback has been removed because it may cor

AssertionError: Q5: True positive rate has incorrect value.
```

Question 6¶

Perform a grid search over the parameters listed below for a Logisitic Regression classifier, using recall for scoring and default 3-fold cross validation. (Suggest to use `solver='liblinear'` , more explanation here (https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html))

`'penalty': ['l1', 'l2']`

`'C':[0.01, 0.1, 1, 10]`

From `.cv_results_` , create an array of the mean test scores of each parameter combination. i.e.

	`l1`	`l2`
`0.01`	?	?
`0.1`	?	?
`1`	?	?
`10`	?	?

This function should return a 4 by 2 numpy array with 8 floats.

Note: do not return a DataFrame, just the values denoted by `?` in a numpy array.

In [13]:

Student's answer

```
def answer_six():
    import numpy as np
    from sklearn.linear_model import LogisticRegression
    from sklearn.model_selection import GridSearchCV

    # Define the parameter grid
    param_grid = {
        'penalty': ['l1', 'l2'],  # Penalty types
        'C': [0.01, 0.1, 1, 10]   # Regularization strengths
    }

    # Create the Logistic Regression model
    lr = LogisticRegression(solver='liblinear')

    # Create the GridSearchCV object, using recall as the scoring metric and 3-fold cross-v
    grid_search = GridSearchCV(lr, param_grid, scoring='recall', cv=3)

    # Fit the model on the training data
    grid_search.fit(X_train, y_train)

    # Get the mean test scores for each parameter combination from the grid search results
    mean_test_scores = grid_search.cv_results_['mean_test_score']

    # Reshape the mean test scores into a 2x4 matrix (since we have 2 penalties and 4 C val
    mean_test_scores = mean_test_scores.reshape(4,2)

    # Return the 2D array of mean test scores
    return mean_test_scores
answer_six()
```

Out[13]:

```
array([[0.66666667, 0.76086957],
       [0.80072464, 0.80434783],
       [0.8115942 , 0.8115942 ],
       [0.80797101, 0.8115942 ]])
```

In [14]:

Grade cell: `cell-6632a909e296b185` Score: 0.0 / 0.0

```
You have failed this test due to an error. The traceback has been removed because it may cor

AssertionError: Q6: The answer at index [0,0] is incorrect.
```

This assignment was graded by mooc_adswpy:e5e20d3b91dd, v1.45.052423