1 Python Basics

For help review Assignment 1 material.

1.1 File paths

What is the difference between a relative and absolute path? Give an example of each.

Solution

An absolute path contains the complete directory list required to locate the file (e.g. /home/user/cpsc330). A relative path contains the directory list to locate a file relative to the current location, so it needs to be combined with another path in order to access a file (e.g. user/cpsc330).

1.2 Data Structures

What type is returned with the following code?

$$df[[\text{"}age\text{"}]] \tag{1}$$

Solution

A DataFrame is returned.

$$df[\text{"}age\text{"}] \tag{2}$$

Solution

A series is returned.

1.3 Slicing

What is the difference between the following two lines of code?

$$df.iloc[0:1] \tag{3}$$

$$df.loc[0:1] \tag{4}$$

Solution

Both pieces of code return a slice.
iloc returns 1 row: the first row at index location 0.
loc returns two rows: the two rows labeled 0 and 1.

2 Machine Learning Fundamentals

For help review Assignment 2 material.

2.1 EDA

During exploratory data analysis, you create 2 histograms of a feature to show the distribution of the feature values in the training set, separated for positive (target=1) and negative (target=0) examples. You observe that the histograms for this particular feature look identical between the two target classes. Would the feature be useful for predicting the target class? Briefly justify your answer.

Solution

This is not enough information to conclude whether or not the feature would be useful in predicting the target class. The particular feature may be predictive in conjunction with other features.

2.2 Text Processing

Why are free-text column features difficult to use in machine learning algorithms?

Solution

Text columns often require multiple transformations before they can be used because machine learning algorithms typically work on numeric data. These transformations are difficult in part because of their complexity. They may require multiple steps such as encoding, embedding, handling of missing values, etc.

2.3 Cross-Validation

Below are the subscores from five folds of cross-validation for 2 machine learning models. Which model appears to be overfitting the most? Which model's average validation score would you trust the least? Which model would you choose to use and why?

Model A	test_score	train_score
0	0.68	0.99
1	0.63	0.99
2	0.70	0.99
3	0.72	0.99
4	0.60	0.99

Model B	test_score	train_score
0	0.76	0.91
1	0.82	0.90
2	0.85	0.90
3	0.79	0.90
4	0.80	0.91

Solution

Model A is overfitting more than Model B. The gap between the validation scores and training scores for Model A are wider, and the training scores are very high.

The average validation score for Model A is 0.666, whereas the average validation score for Model B is 0.804. I would trust the average validation score for Model B (std dev = 0.034) than model A (std dev = 0.045) because there is less variation between test_scores for the given folds.

I would choose to use Model B because its average validation score is much higher and it appears to be overfitting less than Model A.

3 Preprocessing

For help review Assignment 3 material.

3.1 Model Requirements

I just finished an initial exploratory data analysis of my dataset and split it into X_train, y_train, X_test, and y_test subsets. I tried to train sklearn's SVC model on X_train and y_train, but it didn't work. Give two reasons why it may have failed.

Solution

1. There may be categorical columns that need to be processed first.
2. There may be NaN values in numeric columns that need to be processed first.

3.2 C Parameter

I'm tuning an RBF SVC model to pick the best hyperparameter value. I ran cross-validation on the SVC() model in sklearn with a variety of different C values to compare their mean test_score and train_scores. What should I consider when I choose the best C value? What effect does increasing the C value have on this model?

Solution

We should pick the C value that produces the highest cross-validation score. The C value affects the fundamental tradeoff: a larger C value results in a more complex model.

3.3 Transformations

You're tasked to train a machine learning model to predict whether a student in CPSC 330 will get an A+ on the final exam. You gather some preliminary information about the dataset.

```
df.head()
```

	enjoy	major	attnd	year	assign1	assign2	assign3	midterm	final
0	yes	CPSC	Excellent	3	92	93.0	84	92	A+
1	yes	MENG	Average	2	94	90.0	80	91	Not A+
2	yes	MATH	Poor	3	78	85.0	83	80	Not A+
3	no	MATH	Excellent	3	91	NaN	92	89	A+
4	yes	PSYC	Good	4	77	83.0	90	85	A+

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 370 entries, 0 to 369
Data Columns (total 10 columns):
```

#	Column	Non-Null Count	Dtype
0	enjoy	370 non-null	object
1	major	370 non-null	object
3	attnd	370 non-null	object
4	year	370 non-null	int64
5	assign1	370 non-null	int64
6	assign2	369 non-null	float64
7	assign3	370 non-null	int64
8	midterm	370 non-null	int64
9	final	370 non-null	object

```
train_df, test_df = train_test_split(df, test_size=0.2, random_state=42)
train_df['attnd'].unique()
```

```
array(['Excellent', 'Average', 'Poor', 'Good'],dtype=object)
```

Define a preprocessor ct which can be used to apply necessary transformations to the data. Your preprocessor will be used in cross-validation of an SVC() classifier model.

```
# Your code here

ct = # Your code here
pipe = make_pipeline(ct, SVC(random_state=42))
scores = cross_validate(pipe, X_train, y_train, return_train_score=True)
```

Solution

```
# Define column types
numeric_feats = [ 'years', 'assign1', 'assign2', 'assign3', 'midterm']
ordinal_feats = ['attnd']
binary_feats = ['enjoy']
categorical_feats = ['major']

# Define transformations to apply
numeric_transformer = make_pipeline(SimpleImputer(), StandardScaler())
attendance_levels = ['Poor', 'Average', 'Good', 'Excellent']
ordinal_transformer = OrdinalEncoder(categories=[attendance_levels], dtype=int)
binary_transformer = OneHotEncoder(drop='if_binary', dtype=int)
categorical_transformer = OneHotEncoder(handle_unknown='ignore', sparse_output=False)

# Build column transformer object
ct = make_column_transformer(
        (numeric_transformer, numeric_feats),
        (ordinal_tranformer, ordinal_feats),
        (binary_transformer, binary_feats),
        (categorical_transformer, categorical_feats))

pipe = make_pipeline(ct, SVC(random_state=42))
scores = cross_validate(pipe, X_train, y_train, return_train_score=True)
```

4 Hyperparameter Optimization

For help review Assignment 4 material.

4.1 GridSearchCV()

Two individuals are tuning a logistic regression model for text classification using count vectorizer. The goal is to find the optimal hyperparameter values for C and max_features.

Person A follows a manual approach, tuning one hyperparameter at a time. They first experiment with

different values of C while keeping max_features constant, and then vice versa. After a series of cross-validation calls, Person A selects the best hyperparameter values.

Person B, however, opts for a more automated approach and uses GridSearchCV to find the best hyperparameter values. Person B obtains different best values for C and max_features compared to Person A.

How is it possible that Person A and Person B ended up with different optimal hyperparameter values? Generally speaking, should their values agree? Why or why not?

Solution

In general their values will not necessarily agree. GridSearchCV is able to capture interactions between hyperparameters you wouldn't see if you tune the parameters one at a time. Additionally, GridSearchCV can conduct a more exhaustive search than manual tuning.

4.2 Baselines

Describe a scenario where the absence of a baseline model might lead to misguided conclusions or suboptimal decisions in the model development process.

Solution

Example: The absence of a baseline model could mislead a development team into thinking they have created a robust classification model, while in reality, the model might be overfitting to noise and/or inadequately addressing the imbalanced nature of a dataset.

For example, imagine a machine learning project is focused on developing a fraud detection system for credit card transactions. The dataset is imbalanced, with a small percentage of transactions being fraudulent. The team proceeds to train an ensemble model without a benchmark for comparison. The model achieves an impressive accuracy of 98%, leading the team to believe they have created a highly effective fraud detection system.

A simple baseline model, like a DummyClassifier predicting the majority class (non-fraudulent transactions), would have highlighted the challenges associated with this imbalanced dataset. It could create an opportunity for the team to consider addressing the class imbalance and implement alternative scoring techniques for measuring model performance.

5 Performance & Interpretation

For help review Assignment 5 material.

5.1 F1 Score

Discuss a situation where a high F1 score might be more desirable than a high accuracy.

Solution

In situations where there are significant consequences for both types of errors (false positives and false negatives), achieving a high F1 score becomes crucial. For example, in fraud detection, a high F1 score ensures that both minimizing false positives (genuine transactions flagged as fraud) and false negatives (fraudulent transactions missed) are equally important. Accuracy alone may be misleading, especially when classes are imbalanced, as it might be driven by the majority class.

5.2 Interpreting Results

You're using the Ridge() model from sklearn to predict housing prices using the following features:

- Square Feet (eg. 1,800)
- Neighborhood Quality (categories = [poor, fair, good, excellent])
- House Style (categories = [ranch, colonial, split, cape]).

After training, your model learned the following coefficients:

sqft	250
neighborhood	25,000
style_ranch	5,000
style_colonial	25,000
style_split	15,000
style_cape	20,000

ridge.intercept_ returns 16,000

Part 1

Calculate the predicted price for House A given the following information:

	sqft	neighborhood	style
A	1,200	poor	ranch

Solution

$\hat{y} = w_1 * x_1 + w_2 * x_2 + ... + b$
$\hat{y} = (250 * 1,200) + (25,000 * 0) + (5,000 * 1) + 16,000$
$\hat{y} = \$321,000$

Part 2

If House A were moved to an excellent neighborhood, how would that affect the predicted price?

Solution

Moving House A to an excellent neighborhood would add $75,000 to the predicted price (ie. the new calculation for that feature would be 25,000*3).

Part 3

Say we had applied scaling to the numeric feature sqft before training our model. Would this affect our calculation of House A's predicted price from Part 1? Why or why not?

Solution

Yes. If the sqft feature underwent scaling, then the sqft coefficient represents the price per scaled unit. Before calculating the predicted price, we would want to apply the same scaling transformation to House A's sqft value.

5.3 AUC

You created the following plot to visualize performance of three machine learning models (Model A, Model B, and Model C).

Based on the information available in the plot, which model do you think is demonstrating the best overall performance? Does the Model B have a higher false positive rate than Model C? Briefly explain.

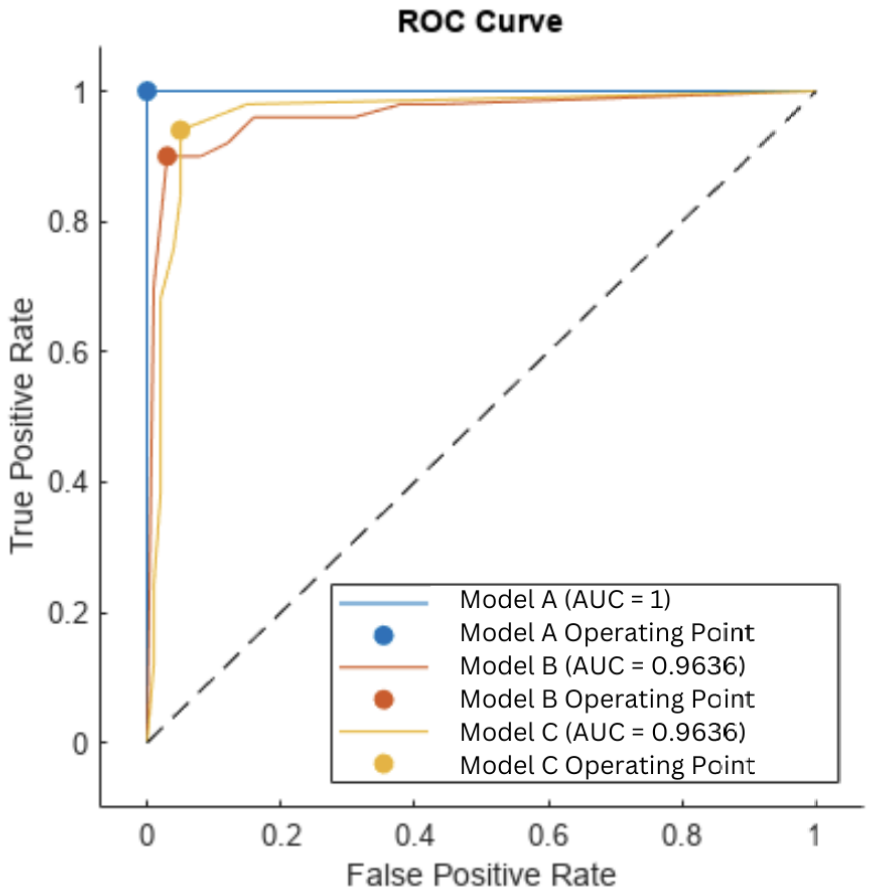


Figure 1: Note the operating point here is the deployed threshold.

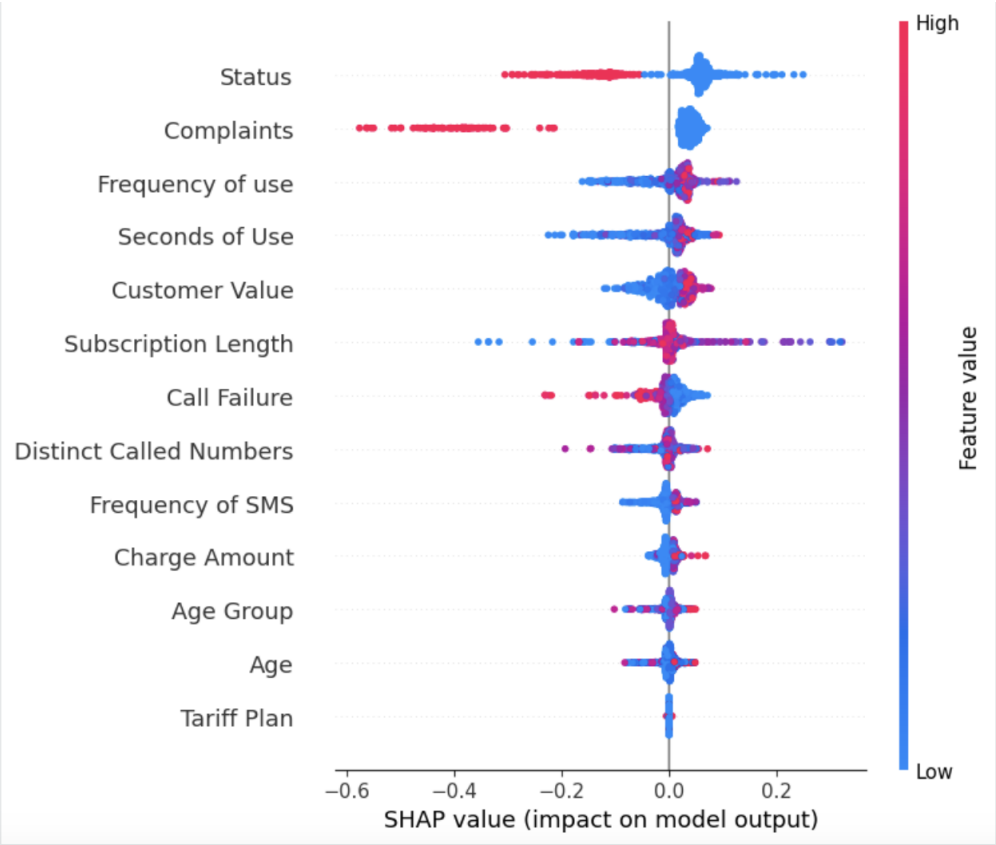
Solution

The Model A appears to have the best performance of the given classifiers. It has the highest AUC value and has a perfect ROC curve.

Whether Model B has a higher false positive rate than the Model C depends on which threshold you are using for your comparison. Most points of the Model B's curve appear to be to the right of Model C's curve, which would indicate a higher false positive rate. However, at some thresholds their curves overlap and it is not clear if one model would have a higher false positive rate than the other.

5.4 SHAP

Below is a SHAP summary plot for label 0 assigned by a RandomForestClassifier model trained on a telecommunication company's customer churn dataset:



Which feature is most important for the model output? Comment on the relationship of this feature and the target class 0.

Solution

The feature names are on the Y-axis and are in order of importance from top to bottom. In this case, the feature "Status" has the greatest importance in contributing to the target class. Presence of status seems to have lower shap values for class 0, whereas absence seems to have larger shap values for class 0.

6 Clustering

For help review Assignment 6 material.

6.1 DBSCAN

When might I want to use DBSCAN instead of K-Means for clustering?

Solution

DBSCAN may perform better when identifying clusters with complex and non-spherical shapes. DBSCAN also does not require specifying the number of clusters beforehand.

6.2 eps & min_samples

Match the following combination of hyperparameter values and their impact on clustering.

a.	High eps, high min_samples	Results in larger, more loosely-defined clusters. Outliers more likely included.
b.	High eps, low min_samples	Leads to small, dense clusters. Algorithm tends to be less sensitive to noise
c.	Low eps, high min_samples	Tends to produce large, sparse clusters. Outliers are less likely to be included.
d.	Low eps, low min_samples	Tends to identify small, dense regions as clusters. Highly sensitive to noise.

Solution

a. High eps, high min_samples: Tends to produce large, sparse clusters. Outliers are less likely to be included.

b. High eps, low min_samples: Results in larger, more loosely-defined clusters. Outliers are more likely to be included.

c. Low eps, high min_samples: Leads to small, dense clusters. Algorithm tends to be less sensitive to noise.

d. Low eps, low min_samples: Tends to identify small, dense regions as clusters. Highly sensitive to noise.

7 Topic Modeling

For help review Assignment 7 material.

7.1 Text Pre-Processing

You're given a corpus of news articles and are tasked to identify high-level themes in the collection of documents. You decide to use topic modeling. Identify two text-cleaning steps you would include in preprocessing of the text and briefly explain why they would be useful.

Solution

Some examples of common text preprocessing steps for topic modeling include:

- Getting rid of slashes, new-line characters, or any other non-informative characters: helps to ensure that irrelevant symbols do not interfere with the topic modeling process. These kinds of characters typically do not contribute to the semantic meaning of the text and may introduce noise.
- Sentence segmentation and tokenization: these tools break down the text into individual sentences, words, and/or tokens, which are essential for creating a structured and analyzable representation of the text.
- Replacing urls, email addresses, or numbers with generic tokens such as "URL," "EMAIL," "NUM.": again, this text may not carry meaningful semantic information for topic modeling and can be treated as noise.
- Getting rid of other fairly unique tokens which are not going to help us in topic modeling: tokens that are too unique or rare might not contribute significantly to understanding of topics. Removing these helps in focusing on more common and representative words.
- Excluding stopwords and punctuation: excluding these helps reduce dimensionality and focuses on more content-rich words that are likely to be more indicative of topics.

- Lemmatization: this step reduces words to their base or root form, which is essential for grouping different forms of the same word together. This in turn helps to capture the core meaning of words and reduces sparsity in the data.

7.2 Content-Based Filtering

Briefly explain how content-based filtering works in the context of recommender systems.

Solution

The idea is to recommend items to a user based on the features or characteristics of items the user has liked or interacted with in the past. This approach builds a user profile based on the features of items the user has shown interest in, and then recommends items with features similar to those in the user profile.

7.3 Recommender Systems

Describe a scenario where collaborative-based filtering for a recommender system would have an advantage over content-based filtering.

Solution

Collaborative-based filtering works better in situations where little information is available about a new user or item. It can recommend items a user might not have discovered based on their own preferences because the preferences of users who are similar to the target user are considered.

8 Time Series

For help review Assignment 8 material.

8.1 Splitting the Data

You are working with CitiBike to train a model to forecast the number of bike rentals next month. CitiBike provided your team with a dataset containing rental history information formatted as a time-ordered sequence of data points. To evaluate the model's ability to generalize, your team divides the time series data into training and testing subsets as follows:

```
citibike = pd.read_csv("data/citibike.csv", index_col=0)
train_df, test_df = train_test_split(citibike, test_size=0.2, random_state=123)
```

What will happen when we split the data this way? Briefly explain.

Solution

If we split the data this way, we are training on data that came after our test data. Since our goal is to forecast, we aren't allowed to know what will happen in the future. Instead, we should split the data by time.
The following general steps can be used for working with time series data: look at the time duration of the data (eg. the min and max timestamps), check for consistent intervals of time between events, pick a date to split your data by and split it).