

Applied Data Science Lab x My Path Module | WorldO x work/ds-curricu (3) - Jupy x Data-Science-Lab(4) explor x Introducing ChatGPT x New chat x

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

041-sqlite.ipynb x 044-demographics.ipynb x 045-assignment.ipynb x

Python 3 (ipykernel)

4.5. Earthquake Damage in Kavrepalanchok NP

In this assignment, you'll build a classification model to predict building damage for the district of **Kavrepalanchok**.

```
[54]: import warnings
import sqlite3

warnings.simplefilter(action="ignore", category=FutureWarning)
warnings.resetwarnings()

[55]: # Import libraries here
import sqlite3
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
from category_encoders import OneHotEncoder
from category_encoders import OrdinalEncoder
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.pipeline import Pipeline, make_pipeline
from sklearn.tree import DecisionTreeClassifier, plot_tree
```

Prepare Data

Connect

Run the cell below to connect to the `nepal.sqlite` database.

Simple 0 9 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

Applied Data Science Lab x My Path Module | WorldO x work/ds-curricu (3) - Jupy x Data-Science-Lab(4) explor x Introducing ChatGPT x New chat x

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

041-sqlite.ipynb x 044-demographics.ipynb x 045-assignment.ipynb x

Python 3 (ipykernel)

Prepare Data

Connect

Run the cell below to connect to the `nepal.sqlite` database.

```
[56]: %load_ext sql
%sql sqlite:///home/jovyan/nepal.sqlite

The sql extension is already loaded. To reload it, use:
%reload_ext sql

[56]: 'Connected: @/home/jovyan/nepal.sqlite'
```

Warning: Be careful with your SQL queries in this assignment. If you try to get all the rows from a table (for example, `SELECT * FROM id_map`), you will cause an Out of Memory error on your virtual machine. So always include a `LIMIT` when first exploring a database.

Task 4.5.1: What districts are represented in the `id_map` table? Determine the unique values in the `district_id` column.

```
[58]: %sql
SELECT name
FROM sqlite_schema
WHERE type = "table"

+ sqlite:///home/jovyan/nepal.sqlite
Done.
```

```
[58]:
```

name
id_map
building_structure
building_damage
household_demographics

Simple 0 9 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

Applied Data Science Lab | My Path Module | WorldOf | work/ds-curricu (3) - Jupyter | Data-Science-Lab(4) explor | Introducing ChatGPT | New chat

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
040-earthquake-damage-in-nepal	
041-sqlite.ipynb	44 minutes ago
041-sqlite.ipynb	25 minutes ago
041-sqlite.ipynb	a month ago
041-sqlite.ipynb	44 minutes ago
041-sqlite.ipynb	a month ago
041-sqlite.ipynb	a month ago
041-sqlite.ipynb	42 minutes ago
041-sqlite.ipynb	6 minutes ago
041-sqlite.ipynb	a month ago

Python 3 (ipykernel)

```
Python master
Score: 1
```

Python 3 (ipykernel) | Idle

Mode: Command | Ln 1, Col 1 | English (United States) | 045-assignment.ipynb

35°C Smoke

Search

Applied Data Science Lab | My Path Module | WorldOf | work/ds-curricu (3) - Jupyter | Data-Science-Lab(4) explor | Introducing ChatGPT | New chat

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
040-earthquake-damage-in-nepal	
041-sqlite.ipynb	an hour ago
041-sqlite.ipynb	25 minutes ago
041-sqlite.ipynb	a month ago
041-sqlite.ipynb	44 minutes ago
041-sqlite.ipynb	a month ago
041-sqlite.ipynb	a month ago
041-sqlite.ipynb	43 minutes ago
041-sqlite.ipynb	6 minutes ago
041-sqlite.ipynb	a month ago

Python 3 (ipykernel)

```
Python master
Score: 1
```

What's the district ID for Kavrepalanchok? From the lessons, you already know that Gorkha is 4; from the textbook, you know that Ramechhap is 2. Of the remaining districts, Kavrepalanchok is the one with the largest number of observations in the id_map table.

Task 4.5.2: Calculate the number of observations in the id_map table associated with district 1.

```
Python master
Score: 1
```

Yes! Your hard work is paying off.

Task 4.5.3: Calculate the number of observations in the id_map table associated with district 3.

Python 3 (ipykernel) | Idle

Mode: Command | Ln 1, Col 46 | English (United States) | 045-assignment.ipynb

35°C Smoke

Search

Applied Data Science Lab | My Path Module | WorldOf | work/ds-curricu (3) - Jupyter | Data-Science-Lab(4) explor | Introducing ChatGPT | New chat

Applied Data Science LabMy Path Module | WorldQuant Uwork/ds-curricu (3) - JupyterLabData-Science-Lab/5 JOIN.py at : +

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

041-sqlite.ipynb044-demographics.ipynb045-assignment.ipynb

Python 3 (pykernel)

Filter files by name

/ / ds-curriculum / 040-earthquake-damage-in-nepal /

Name	Last Modified
.ipynb_chec...	an hour ago
data	26 minutes ago
images	a month ago
041-sqlite.l...	an hour ago
042-logistic...	a month ago
043-decisio...	a month ago
044-demog...	44 minutes ago
045-assign...	7 minutes ago
046-data-d...	a month ago

SQLite:///home/jovyan/nepal.sqlite

Done.

[84]:
count(*)
36112

✓

Yes! Your hard work is paying off.
Score: 1

Task 4.5.3: Calculate the number of observations in the `id_map` table associated with district 3.

[85]:
result = [_.DataFrame().astype(float).squeeze()] # noqa F821
wqet_grader.grade("Project 4 Assessment", "Task 4.5.2", result)

✓

Very impressive.
Score: 1

Task 4.5.4: Join the unique building IDs from `Kavrepalanchok` in `id_map`, all the columns from `building_structure`, and the `damage_grade` column from `building_damage`, limiting. Make

Simple 0 Python 3 (pykernel) | Idle

Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

35°C Smoke

Applied Data Science LabMy Path Module | WorldQuant Uwork/ds-curricu (3) - JupyterLabData-Science-Lab/5 JOIN.py at : +

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

041-sqlite.ipynb044-demographics.ipynb045-assignment.ipynb

Python 3 (pykernel)

Filter files by name

/ / ds-curriculum / 040-earthquake-damage-in-nepal /

Name	Last Modified
.ipynb_chec...	an hour ago
data	27 minutes ago
images	a month ago
041-sqlite.l...	an hour ago
042-logistic...	a month ago
043-decisio...	a month ago
044-demog...	44 minutes ago
045-assign...	seconds ago
046-data-d...	a month ago

Markdown

Task 4.5.4: Join the unique building IDs from `Kavrepalanchok` in `id_map`, all the columns from `building_structure`, and the `damage_grade` column from `building_damage`, limiting. Make sure you rename the `building_id` column in `id_map` as `b_id` and limit your results to the first five rows of the new table.

[90]:
SELECT distinct(i.building_id) AS b_id,
s.*,
d.damage_grade
FROM id_map AS i
JOIN building_structure AS s ON i.building_id = s.building_id
JOIN building_damage AS d ON i.building_id = d.building_id
WHERE district_id = 3
LIMIT 5
SQLite:///home/jovyan/nepal.sqlite
Done.

[90]:
b_id building_id count_floors_pre_eq count_floors_post_eq age_building plinth_area_sq_ft height_ft_pre_eq height_ft_post_eq land_surface_condition foundation_type roof_type ground_floor_type
87473 87473 2 1 15 382 18 7 Flat Mud mortar-Stone/Brick Bamboo/Timber-Light roof Mud
87479 87479 1 0 12 328 7 0 Flat Mud mortar-Stone/Brick Bamboo/Timber-Light roof Mud
87482 87482 2 1 23 427 20 7 Flat Mud mortar-Stone/Brick Bamboo/Timber-Light roof Mud
87491 87491 2 1 12 427 14 7 Flat Mud mortar-Stone/Brick Bamboo/Timber-Light roof Mud
87496 87496 2 0 32 360 18 0 Flat Mud mortar-Stone/Brick Bamboo/Timber-Light roof Mud

[91]:
result = [_.DataFrame().set_index("b_id")] # noqa F821
wqet_grader.grade("Project 4 Assessment", "Task 4.5.4", result)

✓

Python master
Score: 1

Simple 0 Python 3 (pykernel) | Idle

Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

35°C Smoke

Applied Data Science LabMy Path Module | WorldQuant Uwork/ds-curricu (3) - JupyterLabData-Science-Lab/5 JOIN.py at : +

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

Applied Data Science LabMy Path Module | WorldQuantwork/ds-curricu (3) - JupyterLabData-Science-Lab(6) wrangle0.p.x +

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
.ipynb_chec...	an hour ago
data	28 minutes ago
images	a month ago
041-sqlite...	an hour ago
042-logic...	a month ago
043-decisio...	a month ago
044-demog...	an hour ago
045-assign...	a minute ago
046-data-d...	a month ago

041-sqlite.ipynb044-demographics.ipynb045-assignment.ipynb

Python 3 (pykernel)

Import

Task 4.5.5: Write a `wrangle` function that will use the query you created in the previous task to create a DataFrame. In addition your function should:

- Create a "severe_damage" column, where all buildings with a damage grade greater than 3 should be encoded as 1. All other buildings should be encoded at 0.
- Drop any columns that could cause issues with leakage or multicollinearity in your model.

```
[92]: # Build your "wrapgle" function here
def wrangle(db_path):
    # Connect to database using connect method
    conn = sqlite3.connect(db_path)

    # Construct query
    query = """
    SELECT distinct(i.building_id) AS b_id,
           s.*,
           d.damage_grade
    FROM id_map AS i
    JOIN building_structure AS s ON i.building_id = s.building_id
    JOIN building_damage AS d ON i.building_id = d.building_id
    WHERE district_id = 3
    """

    # Read query results into DataFrame
    df = pd.read_sql(query, conn, index_col="b_id")

    # Identify leaky columns
    drop_cols = [col for col in df.columns if "post_eq" in col]

    # Create binary target
    df["damage_grade"] = df["damage_grade"].str[-1].astype(int)
    df["severe_damage"] = (df["damage_grade"] > 3).astype(int) # encode as 0's and 1's

    # Drop old target
    drop_cols.append("damage_grade")

    # Drop multicollinearity column
    drop_cols.append("count_floors_pre_eq")

    return df
```

Simple09Python 3 (pykernel) | Busy

Mode: CommandLn 1, Col 1English (United States)045-assignment.ipynb

35°CSmoke

Search

ENG12:2513-03-2023

Applied Data Science LabMy Path Module | WorldQuantwork/ds-curricu (3) - JupyterLabData-Science-Lab(6) wrangle0.p.x +

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
.ipynb_chec...	an hour ago
data	28 minutes ago
images	a month ago
041-sqlite...	an hour ago
042-logic...	a month ago
043-decisio...	a month ago
044-demog...	an hour ago
045-assign...	a minute ago
046-data-d...	a month ago

041-sqlite.ipynb044-demographics.ipynb045-assignment.ipynb

Python 3 (pykernel)

```
# Drop high categorical features
drop_cols.append("building_id")

# Drop columns
df.drop(columns=drop_cols, inplace=True)

return df
```

Use your `wrangle` function to query the database at `"/home/jovyan/nepal.sqlite"` and return your cleaned results.

```
[93]: df = wrangle("/home/jovyan/nepal.sqlite")
df.head()
```

```
[93]:
```

b_id	age_building	plinth_area_sq_ft	height_ft_pre_eq	land_surface_condition	foundation_type	roof_type	ground_floor_type	other_floor_type	position	plan_configuration	superstructure	severe_dam
87473	15	382	18	Flat	Mud mortar-Stone/Brick	Bamboo/Timber-Light roof	Mud	Timber/Bamboo-Mud	Not attached	Rectangular	Stone, mud mortar	
87479	12	328	7	Flat	Mud mortar-Stone/Brick	Bamboo/Timber-Light roof	Mud	Not applicable	Not attached	Rectangular	Stone, mud mortar	
87482	23	427	20	Flat	Mud mortar-Stone/Brick	Bamboo/Timber-Light roof	Mud	Timber/Bamboo-Mud	Not attached	Rectangular	Stone, mud mortar	
87491	12	427	14	Flat	Mud mortar-Stone/Brick	Bamboo/Timber-Light roof	Mud	Timber/Bamboo-Mud	Not attached	Rectangular	Stone, mud mortar	
87496	32	360	18	Flat	Mud mortar-Stone/Brick	Bamboo/Timber-Light roof	Mud	Timber/Bamboo-Mud	Not attached	Rectangular	Stone, mud mortar	

```
[94]: wget_grader.grade(
    "Project 4 Assessment", "Task 4.5.5", wrangle("/home/jovyan/nepal.sqlite")
)
```

Yup. You got it.
Kudos!

Simple09Python 3 (pykernel) | Idle

Mode: CommandLn 1, Col 1English (United States)045-assignment.ipynb

35°CSmoke

Search

ENG12:2613-03-2023

Applied Data Science Lab | My Path Module | WorldQuant | work/ds-curricu (3) - JupyterLab | Data-Science-Lab(7) barChart.py | +

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

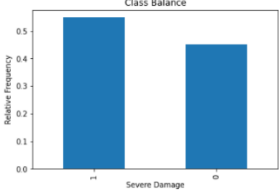
041-sqlite.ipynb | 044-demographics.ipynb | 045-assignment.ipynb

Python 3 (ipykernel)

Explore

Task 4.5.6: Are the classes in this dataset balanced? Create a bar chart with the normalized value counts from the "severe_damage" column. Be sure to label the x-axis "Severe Damage" and the y-axis "Relative Frequency". Use the title "Kavrepalanchok, Class Balance".

```
[95]: # Plot value counts of "severe_damage"
df["severe_damage"].value_counts(normalize=True).plot(
    kind="bar", xlabel="Severe Damage", ylabel="Relative Frequency", title="Class Balance"
);
# Don't delete the code below
plt.savefig("Images/4-5-6.png", dpi=150)
```



```
[96]: with open("Images/4-5-6.png", "rb") as file:
wqet_grader.grade("Project 4 Assessment", "Task 4.5.6", file)
```

Excellent work.
Score: 1

Task 4.5.7: Is there a relationship between the footprint size of a building and the damage it sustained in the earthquake? Use seaborn to create a boxplot that shows the distributions of the

35°C Smoke | Search | 12:26 13-03-2023

Applied Data Science Lab | My Path Module | WorldQuant | work/ds-curricu (3) - JupyterLab | Data-Science-Lab(8) boxplot.py | +

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

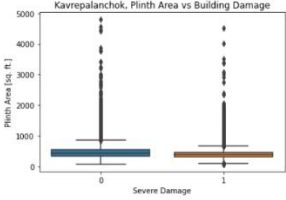
041-sqlite.ipynb | 044-demographics.ipynb | 045-assignment.ipynb

Python 3 (ipykernel)

Task 4.5.7:

Is there a relationship between the footprint size of a building and the damage it sustained in the earthquake? Use seaborn to create a boxplot that shows the distributions of the "plinth_area_sq_ft" column for both groups in the "severe_damage" column. Label your x-axis "Severe Damage" and y-axis "Plinth Area [sq. ft.]". Use the title "Kavrepalanchok, Plinth Area vs Building Damage".

```
[97]: sns.boxplot(x="severe_damage", y="plinth_area_sq_ft", data=df)
plt.xlabel("Severe Damage")
plt.ylabel("Plinth Area [sq. ft.]")
plt.title("Kavrepalanchok, Plinth Area vs Building Damage");
# Don't delete the code below
plt.savefig("Images/4-5-7.png", dpi=150)
```



```
[98]: with open("Images/4-5-7.png", "rb") as file:
wqet_grader.grade("Project 4 Assessment", "Task 4.5.7", file)
```

Python master 🏆
Score: 1

Task 4.5.8: Are buildings with certain roof types more likely to suffer severe damage? Create a pivot table of df where the index is "roof_type" and the values come from the "severe_damage" column, aggregated by the mean.

Applied Data Science Lab

My Path Module | WorldQuant U

Data-Science-Lab/9) pivot_table

work/ds-curricu (3) - JupyterLab

+

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name

Last Modified

ipynb_chec...

an hour ago

data

29 minutes ago

images

seconds ago

041-sqlite1...

an hour ago

042-logistic...

a month ago

043-decisio...

a month ago

044-demog...

an hour ago

045-assign...

seconds ago

046-data-d...

a month ago

041-sqlite.ipynb

044-demographics.ipynb

045-assignment.ipynb

Python 3 (ipykernel)

Python master

Score: 1

Task 4.5.8: Are buildings with certain roof types more likely to suffer severe damage? Create a pivot table of `df` where the index is `"roof_type"` and the values come from the `"severe_damage"` column, aggregated by the mean.

[99]:

```
roof_pivot = pd.pivot_table(df, index="roof_type", values="severe_damage", aggfunc=np.mean, # roof_type: column in table).sort_values(by="severe_damage")roof_pivot
```

[99]:

	severe_damage
roof_type	
RCC/RB/RBC	0.040715
Bamboo/Timber-Heavy roof	0.569477
Bamboo/Timber-Light roof	0.604842

[100]:

```
wget_grader.grade("Project 4 Assessment", "Task 4.5.8", roof_pivot)
```

Awesome work.

Score: 1

Split

Task 4.5.9: Create your feature matrix `X` and target vector `y`. Your target is `"severe_damage"`.

[101]:

```
X = df.drop(columns="severe_damage") # feature matrix: all columns apart from severe_damagey = df["severe_damage"] # target vectorprint("X shape:", X.shape)print("y shape:", y.shape)X shape: (76533, 11)y shape: (76533,)
```

[102]:

```
wget_grader.grade("Project 4 Assessment", "Task 4.5.9a", X)
```

Yup. You got it.

Score: 1

[104]:

```
wget_grader.grade("Project 4 Assessment", "Task 4.5.9b", y)
```

Python master

Score: 1

Task 4.5.10: Divide your dataset into training and validation sets using a randomized split. Your validation set should be 20% of your data.

[]:

```
X_train, X_val, y_train, y_val = ...print("X_train shape:", X_train.shape)
```

Mode: CommandLn 1, Col 1English (United States)045-assignment.ipynb

35°CSmoke

Search

ENG12:2713-03-2023

Applied Data Science Lab

My Path Module | WorldQuant U

Data-Science-Lab/9) vertical_sp

work/ds-curricu (3) - JupyterLab

+

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name

Last Modified

ipynb_chec...

an hour ago

data

30 minutes ago

images

a minute ago

041-sqlite1...

an hour ago

042-logistic...

a month ago

043-decisio...

a month ago

044-demog...

an hour ago

045-assign...

a minute ago

046-data-d...

a month ago

041-sqlite.ipynb

044-demographics.ipynb

045-assignment.ipynb

Python 3 (ipykernel)

Python master

Score: 1

Split

Task 4.5.9: Create your feature matrix `X` and target vector `y`. Your target is `"severe_damage"`.

[101]:

```
X = df.drop(columns="severe_damage") # feature matrix: all columns apart from severe_damagey = df["severe_damage"] # target vectorprint("X shape:", X.shape)print("y shape:", y.shape)X shape: (76533, 11)y shape: (76533,)
```

[102]:

```
wget_grader.grade("Project 4 Assessment", "Task 4.5.9a", X)
```

Yup. You got it.

Score: 1

[104]:

```
wget_grader.grade("Project 4 Assessment", "Task 4.5.9b", y)
```

Python master

Score: 1

Task 4.5.10: Divide your dataset into training and validation sets using a randomized split. Your validation set should be 20% of your data.

[]:

```
X_train, X_val, y_train, y_val = ...print("X_train shape:", X_train.shape)
```

Mode: CommandLn 1, Col 1English (United States)045-assignment.ipynb

35°CSmoke

Search

ENG12:2713-03-2023

Applied Data Science Lab

My Path Module | WorldQuant U

Data-Science-Lab/92 horizontal

work/ds-curricu (3) - JupyterLab

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
041-sqlite...	an hour ago
data	30 minutes ago
images	2 minutes ago
041-sqlite...	an hour ago
042-logistic...	a month ago
043-decisio...	a month ago
044-demog...	an hour ago
045-assign...	2 minutes ago
046-data-d...	a month ago

Python 3 (ipykernel)

wget_grader.grade("Project 4 Assessment", "Task 4.5.9b", y)

Python master

Score: 1

Task 4.5.10: Divide your dataset into training and validation sets using a randomized split. Your validation set should be 20% of your data.

```
[105]: X_train, X_val, y_train, y_val = train_test_split(
      X, y, test_size=0.2, random_state=42
      )
      print("X_train shape:", X_train.shape)
      print("y_train shape:", y_train.shape)
      print("X_val shape:", X_val.shape)
      print("y_val shape:", y_val.shape)

X_train shape: (61226, 11)
y_train shape: (61226,)
X_val shape: (15307, 11)
y_val shape: (15307,)
```

```
[106]: wget_grader.grade("Project 4 Assessment", "Task 4.5.10", [X_train.shape == (61226, 11)])
```

You got it. Dance party time!

Score: 1

Build Model

Baseline

Task 4.5.11: Calculate the baseline accuracy score for your model.

Single 0 9 Python 3 (ipykernel) | Idle

Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

35°C Smoke

Search

Applied Data Science Lab

Data-Science-Lab/94 log_reg.py

My Path Module | WorldQuant U

work/ds-curricu (3) - JupyterLab

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

Name	Last Modified
041-sqlite...	an hour ago
data	31 minutes ago
images	3 minutes ago
041-sqlite...	an hour ago
042-logistic...	a month ago
043-decisio...	a month ago
044-demog...	an hour ago
045-assign...	seconds ago
046-data-d...	a month ago

Python 3 (ipykernel)

[106]: wget_grader.grade("Project 4 Assessment", "Task 4.5.10", [X_train.shape == (61226, 11)])

You got it. Dance party time!

Score: 1

Build Model

Baseline

Task 4.5.11: Calculate the baseline accuracy score for your model.

```
[107]: acc_baseline = y_train.value_counts(normalize=True).max() # normalize gives you the relative freq
      print("Baseline Accuracy:", round(acc_baseline, 2))

Baseline Accuracy: 0.55
```

```
[108]: wget_grader.grade("Project 4 Assessment", "Task 4.5.11", [acc_baseline])
```

Very impressive.

Score: 1

Iterate

Task 4.5.12: Create a model `model_lr` that uses logistic regression to predict building damage. Be sure to include an appropriate encoder for categorical features.

```
[110]: model_lr = make_pipeline(
      OneHotEncoder(use_cat_names=True),
      LogisticRegression(max_iter=1000-3000) #max_iter: varies; suppresses the 'ConvergenceWarning'
```

Single 0 9 Python 3 (ipykernel) | Idle

Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

35°C Smoke

Search

Applied Data Science Lab

Data-Science-Lab/94 log_reg.py

My Path Module | WorldQuant U

work/ds-curricu (3) - JupyterLab

Applied Data Science Lab | My Path Module | WorldQuant | Data-Science-Lab/94 log_reg.py | work/ds-curricu - JupyterLab

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher 045-assignment.ipynb

Filter files by name

Name	Last Modified
.ipynb_chec...	an hour ago
data	an hour ago
images	29 minutes ago
041-sqlite...	an hour ago
042-logic...	a month ago
043-decisio...	a month ago
044-demog...	an hour ago
045-assign...	seconds ago
046-data-d...	a month ago

Iterate

Task 4.5.12: Create a model `model_lr` that uses logistic regression to predict building damage. Be sure to include an appropriate encoder for categorical features.

```
[50]: model_lr = make_pipeline(
    OneHotEncoder(use_cat_names=True),
    LogisticRegression(max_iter=1000) # max_iter: varies: suppresses the 'ConvergenceWarning'
)
# Fit model to training data
model_lr.fit(X_train, y_train)
```

```
[50]: Pipeline
      OneHotEncoder
      LogisticRegression
```

```
[51]: wqget_grader.grade("Project 4 Assessment", "Task 4.5.12", model_lr)
```

✓ You = coding
Score: 1

Task 4.5.13: Calculate training and validation accuracy score for `model_lr`.

```
[ ]: lr_train_acc = accuracy_score(y_train, model_lr.predict(X_train))
lr_val_acc = model_lr.score(X_val, y_val)

print("Logistic Regression, Training Accuracy Score:", lr_train_acc)
print("Logistic Regression, Validation Accuracy Score:", lr_val_acc)
```

Simple 0 9 Python 3 (ipykernel) | Idle Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

Applied Data Science Lab | My Path Module | WorldQuant | Data-Science-Lab/96 decision.py | work/ds-curricu - JupyterLab

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher 045-assignment.ipynb

Filter files by name

Name	Last Modified
.ipynb_chec...	an hour ago
data	an hour ago
images	30 minutes ago
041-sqlite...	an hour ago
042-logic...	a month ago
043-decisio...	a month ago
044-demog...	an hour ago
045-assign...	2 minutes ago
046-data-d...	a month ago

```
[52]: lr_train_acc = accuracy_score(y_train, model_lr.predict(X_train))
lr_val_acc = model_lr.score(X_val, y_val)

print("Logistic Regression, Training Accuracy Score:", lr_train_acc)
print("Logistic Regression, Validation Accuracy Score:", lr_val_acc)

Logistic Regression, Training Accuracy Score: 0.6513735994512135
Logistic Regression, Validation Accuracy Score: 0.6530998889397008
```

```
[53]: submission = [lr_train_acc, lr_val_acc]
wqget_grader.grade("Project 4 Assessment", "Task 4.5.13", submission)
```

✓ Correct.
Score: 1

Task 4.5.14: Perhaps a decision tree model will perform better than logistic regression, but what's the best hyperparameter value for `max_depth`? Create a `for` loop to train and evaluate the model `model_dt` at all depths from 1 to 15. Be sure to use an appropriate encoder for your model, and to record its training and validation accuracy scores at every depth. The grader will evaluate your validation accuracy scores only.

```
[*]: depth_hyperparams = range(1, 16) # for max_depth
training_acc = []
validation_acc = []
for d in depth_hyperparams:
    model_dt = make_pipeline(
        OrdinalEncoder(),
        DecisionTreeClassifier(max_depth=d, random_state=42)
    )
    # Fit model to training data
    model_dt.fit(X_train, y_train)
    # Calculate training accuracy score and append to 'training_acc'
    training_acc.append(model_dt.score(X_train, y_train))
    # Calculate validation accuracy score and append to 'validation_acc'
    validation_acc.append(model_dt.score(X_val, y_val))

print("Training Accuracy Scores:", training_acc[1:16])
```

Simple 0 9 Python 3 (ipykernel) | Busy Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

Applied Data Science Lab

My Path Module | WorldQuant U

Data-Science-Lab/96 decision...

work/ds-curricu - JupyterLab

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ / ds-curriculum / 040-earthquake-damage-in-nepal /

Name	Last Modified
.ipynb_chec...	an hour ago
data	an hour ago
images	32 minutes ago
041-sqlite...	an hour ago
042-logic...	a month ago
043-decisio...	a month ago
044-demog...	an hour ago
045-assign...	a minute ago
046-data-d...	a month ago

045-assignment.ipynb

Launcher

Markdown

Python 3 (pykernel)

Task 4.5.14: Perhaps a decision tree model will perform better than logistic regression, but what's the best hyperparameter value for `max_depth`? Create a `for` loop to train and evaluate the model `model_dt` at all depths from 1 to 15. Be sure to use an appropriate encoder for your model, and to record its training and validation accuracy scores at every depth. The grader will evaluate your validation accuracy scores only.

```
[54]:
depth_hyperparams = range(1, 16) # for max_depth
training_acc = []
validation_acc = []
for d in depth_hyperparams:
    model_dt = make_pipeline(
        OrdinalEncoder(),
        DecisionTreeClassifier(max_depth=d, random_state=42)
    )
    # Fit model to training data
    model_dt.fit(X_train, y_train)
    # Calculate training accuracy score and append to 'training_acc'
    training_acc.append(model_dt.score(X_train, y_train))
    # Calculate validation accuracy score and append to 'validation_acc'
    validation_acc.append(model_dt.score(X_val, y_val))

print("Training Accuracy Scores:", training_acc[:6])
print("Validation Accuracy Scores:", validation_acc[:6])

Training Accuracy Scores: [0.6303041191650606, 0.6303041191650606, 0.642292490118577, 0.653529546271192, 0.6543951915852743, 0.6576617776761506]
Validation Accuracy Scores: [0.6350035931273273, 0.6350035931273273, 0.6453909975828053, 0.6527732410080493, 0.6529039001763899, 0.6584569151368654]

[55]:
submission = pd.Series(validation_acc, index=depth_hyperparams)
wqet_grader.grade("Project 4 Assessment", "Task 4.5.14", submission)



✓



That's the right answer. Keep it up!



Score: 1



Task 4.5.15: Using the values in training_acc and validation_acc, plot the validation curve for model_dt. Label your x-axis "Max Depth" and your y-axis "Accuracy Score". Use the title "Validation Curve, Decision Tree Model", and include a legend.


```

Simple

Python 3 (pykernel) | Idle

Mode: Command

Ln 1, Col 1

English (United States)

045-assignment.ipynb

36°C

Smoke

Search

12:58

13-03-2023

Applied Data Science Lab

My Path Module | WorldQuant U

Data-Science-Lab/97 validation...

work/ds-curricu - JupyterLab

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Filter files by name

/ / ds-curriculum / 040-earthquake-damage-in-nepal /

Name	Last Modified
.ipynb_chec...	an hour ago
data	an hour ago
images	seconds ago
041-sqlite...	an hour ago
042-logic...	a month ago
043-decisio...	a month ago
044-demog...	an hour ago
045-assign...	2 minutes ago
046-data-d...	a month ago

045-assignment.ipynb

Launcher

Markdown

Python 3 (pykernel)

```
[56]: # Validation curve
plt.plot(depth_hyperparams, training_acc, label="Training")
plt.plot(depth_hyperparams, validation_acc, label="validation")
plt.xlabel("Max Depth")
plt.ylabel("Accuracy Score")
plt.title("Validation Curve, Decision Tree Model")
plt.legend();

# build & fit again
final_model_dt = make_pipeline(
    OrdinalEncoder(),
    DecisionTreeClassifier(max_depth=10, random_state=42)
)

# Don't delete the code below
plt.savefig("images/4-5-15.png", dpi=150)
```

Max Depth	Training Accuracy Score	Validation Accuracy Score
1	0.630304	0.635004
2	0.630304	0.635004
3	0.642292	0.645391
4	0.653530	0.652773
5	0.654395	0.652904
6	0.657662	0.658457
7	0.660939	0.660939
8	0.664216	0.664216
9	0.667493	0.667493
10	0.670770	0.660939
11	0.674047	0.660939
12	0.677324	0.660939
13	0.680601	0.660939
14	0.683878	0.660939
15	0.687155	0.660939

```
[57]: with open("images/4-5-15.png", "rb") as file:
wqet_grader.grade("Project 4 Assessment", "Task 4.5.15", file)
```

Simple

Python 3 (pykernel) | Idle

Mode: Command

Ln 1, Col 1

English (United States)

045-assignment.ipynb

36°C

Smoke

Search

12:59

13-03-2023

Applied Data Science Lab | My Path Module | WorldQuant | Data-Science-Lab/98 tests.py | work/ds-curricu - JupyterLab

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher 045-assignment.ipynb Python 3 (pykernel)

You got it. Dance party time! 🎉
Score: 1

Task 4.5.16: Build and train a new decision tree model `final_model_dt`, using the value for `max_depth` that yielded the best validation accuracy score in your plot above.

```
[58]: # Fit model to training data
final_model_dt.fit(X, y) #final_model_dt.fit(X_train, y_train)

[59]:
Pipeline
- OrdinalEncoder
- DecisionTreeClassifier

[59]:
uqet_grader.grade("Project 4 Assessment", "Task 4.5.16", final_model_dt)
```

Yes! Keep on rockin'. e That's right.
Score: 1

Evaluate

Task 4.5.17: How does your model perform on the test set? First, read the CSV file `"data/kavrepalanchok-test-features.csv"` into the DataFrame `X_test`. Next, use `final_model_dt` to generate a list of test predictions `y_test_pred`. Finally, submit your test predictions to the grader to see how your model performs.

Tip: Make sure the order of the columns in `X_test` is the same as in your `X_train`. Otherwise, it could hurt your model's performance.

```
[61]: X_test = pd.read_csv("data/kavrepalanchok-test-features.csv", index_col="b_id")
y_test_pred = pd.Series(final_model_dt.predict(X_test))
y_test_pred[:5]
```

Simple 0 9 Python 3 (pykernel) | Idle Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

36°C Smoke

Applied Data Science Lab | My Path Dashboard | WorldQuant | Data-Science-Lab/98 tests.py | work/ds-curricu - JupyterLab | Introducing ChatGPT | New chat

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher 045-assignment.ipynb Python 3 (pykernel)

Evaluate

Task 4.5.17: How does your model perform on the test set? First, read the CSV file `"data/kavrepalanchok-test-features.csv"` into the DataFrame `X_test`. Next, use `final_model_dt` to generate a list of test predictions `y_test_pred`. Finally, submit your test predictions to the grader to see how your model performs.

Tip: Make sure the order of the columns in `X_test` is the same as in your `X_train`. Otherwise, it could hurt your model's performance.

```
[94]: X_test = pd.read_csv("data/kavrepalanchok-test-features.csv", index_col="b_id")
y_test_pred = pd.Series(final_model_dt.predict(X_test))
y_test_pred[:5]

[94]: 0 1
1 1
2 0
3 1
4 0
dtype: int64

[96]: test_acc = final_model_dt.score(X_test, y_test_pred)
print("Test Accuracy:", round(test_acc, 2))
Test Accuracy: 1.0

[98]: acc_train = accuracy_score(y_train, model_lr.predict(X_train))
acc_test = model_lr.score(X_test, y_test_pred)
print("LR Training Accuracy:", acc_train)
print("LR Validation Accuracy:", acc_test)

LR Training Accuracy: 0.6513735994512135
LR Validation Accuracy: 0.8438073862477046

[99]: submission = pd.Series(y_test_pred)
uqet_grader.grade("Project 4 Assessment", "Task 4.5.17", submission)
```

Simple 0 9 Python 3 (pykernel) | Idle Mode: Command Ln 1, Col 1 English (United States) 045-assignment.ipynb

36°C Smoke

Applied Data Science Lab | My Path Dashboard | World | Data-Science-Lab/98 tests | work/ds-curriculum - JupyterLab | Introducing ChatGPT | New chat

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher 045-assignment.ipynb Python 3 (ipykernel)

Filter files by name

Name Last Modified

- ipynb_chec... 2 hours ago
- data an hour ago
- images 17 minutes ago
- 041-sqlite... 2 hours ago
- 042-logistic... a month ago
- 043-decisio... a month ago
- 044-demog... 2 hours ago
- 045-assign... seconds ago
- 046-data-d... a month ago

Communicate Results

Task 4.5.18: What are the most important features for `final_model_dt`? Create a Series `feat_imp`, where the index labels are the feature names for your dataset and the values are the feature importances for your model. Be sure that the Series is sorted from smallest to largest feature importance.

```
[100]: features = X_train.columns
importances = final_model_dt.named_steps["decisiontreeclassifier"].feature_importances_
feat_imp = pd.Series(importances, Index=features).sort_values()
feat_imp.head()
```

```
[100]: plan_configuration      0.004032
       position              0.007129
       land_surface_condition 0.008241
       ground_floor_type     0.009741
       foundation_type        0.010620
       dtype: float64
```

```
[101]: features = model_lr.named_steps["onehotencoder"].get_feature_names()
importances = model_lr.named_steps["logisticregression"].coef_[0]
feat_imp = pd.Series(np.exp(importances), Index=features).sort_values()
feat_imp.head()
```

```
[101]: foundation_type_RC      0.430515
       superstructure_Brick, cement mortar 0.506657
       roof_type_RCC/RB/RCC 0.566908
       ground_floor_type_RC 0.599185
       superstructure_RC, non-engineered 0.616918
       dtype: float64
```

```
[102]: wqet_grader.grade("Project 4 Assessment", "Task 4.5.18", feat_imp)
```

Exception Traceback (most recent call last)

36°C Smoke

Applied Data Science Lab | My Path Dashboard | World | Data-Science-Lab/98 tests | work/ds-curriculum - JupyterLab | Introducing ChatGPT | New chat

vm.wqu.edu/lab/tree/work/ds-curriculum/040-earthquake-damage-in-nepal/045-assignment.ipynb

File Edit View Run Kernel Tabs Settings Help

Launcher 045-assignment.ipynb Python 3 (ipykernel)

Filter files by name

Name Last Modified

- ipynb_chec... 2 hours ago
- data an hour ago
- images 17 minutes ago
- 041-sqlite... 2 hours ago
- 042-logistic... a month ago
- 043-decisio... a month ago
- 044-demog... 2 hours ago
- 045-assign... seconds ago
- 046-data-d... a month ago

149 # Used only in testing

Exception: Could not grade submission: Could not verify access to this assessment: Received error from WQET submission API: You have already passed this course!

Task 4.5.19: Create a horizontal bar chart of `feat_imp`. Label your x-axis "Gini Importance" and your y-axis "Label". Use the title "Kavrepalanchok Decision Tree, Feature Importance".

Do you see any relationship between this plot and the exploratory data analysis you did regarding roof type?

```
[103]: # Create horizontal bar chart of feature importances
# horizontal bar chart
feat_imp.plot(kind="barh")
plt.xlabel("Importance")
plt.ylabel("Label")
plt.title("Feature Importance")
# Don't delete the code below
plt.tight_layout()
plt.savefig("Images/4-5-19.png", dpi=150)
```

```
[104]: with open("Images/4-5-19.png", "rb") as file:
wqet_grader.grade("Project 4 Assessment", "Task 4.5.19", file)
```