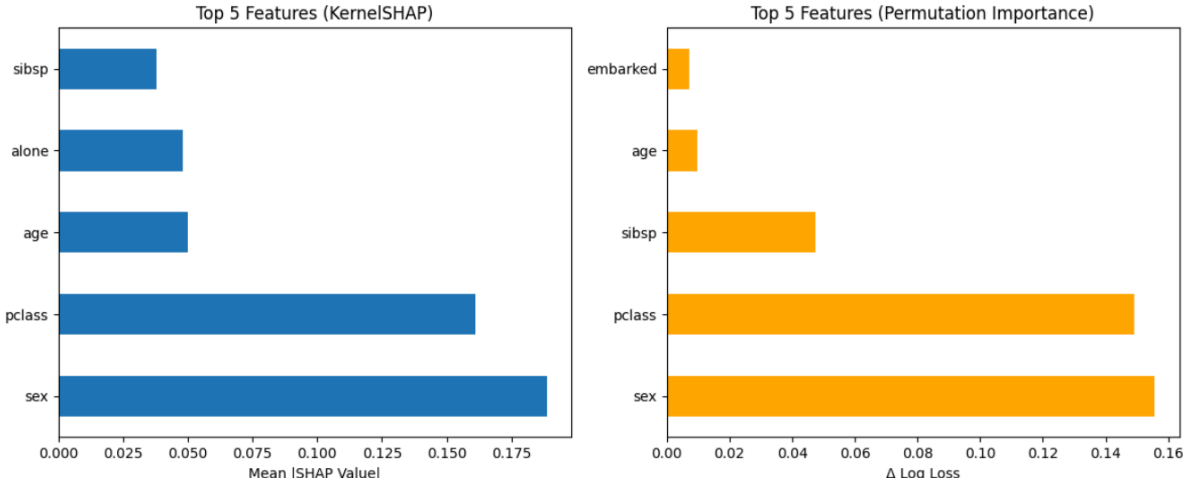
EAI ASSIGNMENT 3
AKANKSHA SINGAL 2021008

QUESTION 1:



Top 5 Feature Comparison:

KernelSHAP Rank	Permutation Rank
sex	sex
pclass	pclass
age	sibsp
alone	age
sibsp	embarked

KernelSHAP computes the contribution of each feature by approximating Shapley values using perturbation-based sampling. Permutation Importance manually measures importance by permuting each feature in isolation and measuring the average drop in model performance - loss function comparisons

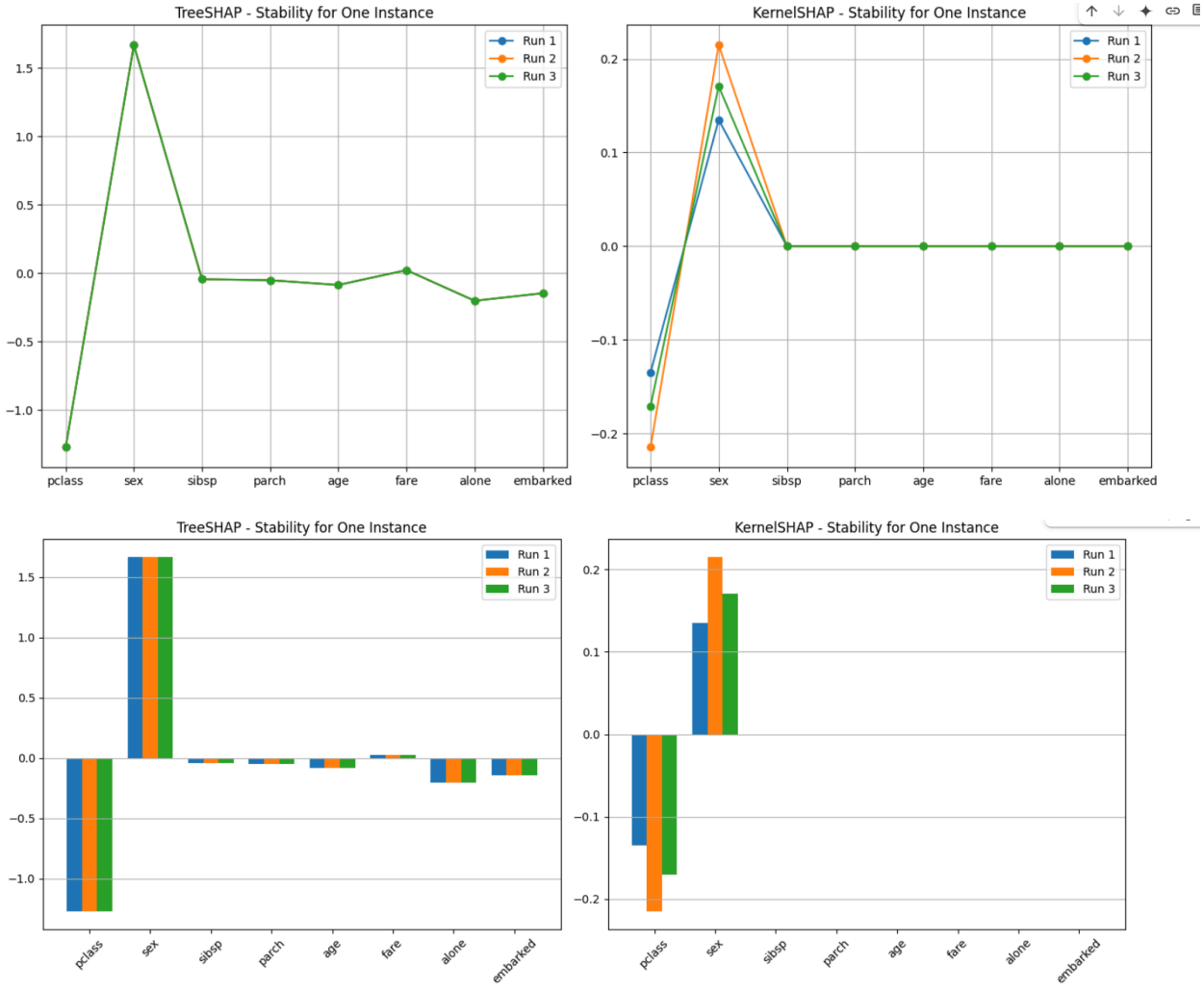
Both these methods agree on the top two features (sex, pclass) are consistently ranked highest by both methods. This indicates strong global influence of these features on the model's output. They disagree on lower-ranked features differ. KernelSHAP gives higher importance to alone and sibsp. Permutation gives more importance to embarked and demotes alone.

This may be because KernelSHAP is model-aware and considers feature interactions and conditional dependence between features. Permutation Importance is model-agnostic but assumes feature independence which is not true in Titanic dataset

Permutation Importance can be biased if a feature is highly correlated with others (masking its individual effect), while SHAP accounts for this via marginal contribution averaging.

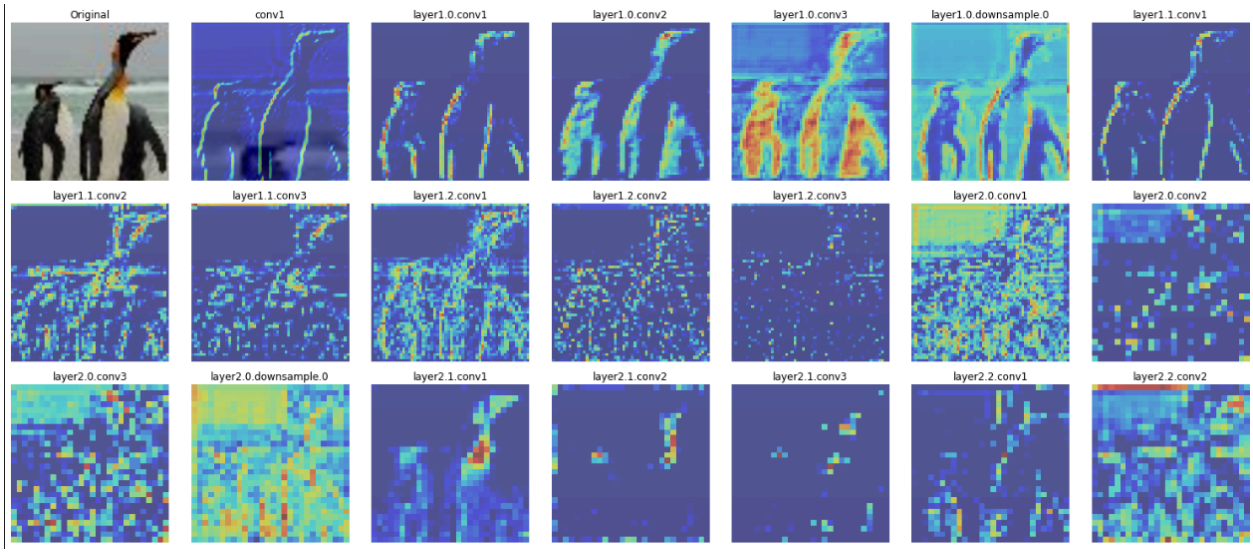
QUESTION 2:

```
TreeSHAP Avg Non-zero Features per Instance: 8.00
KernelSHAP Avg Non-zero Features per Instance: 2.00
```



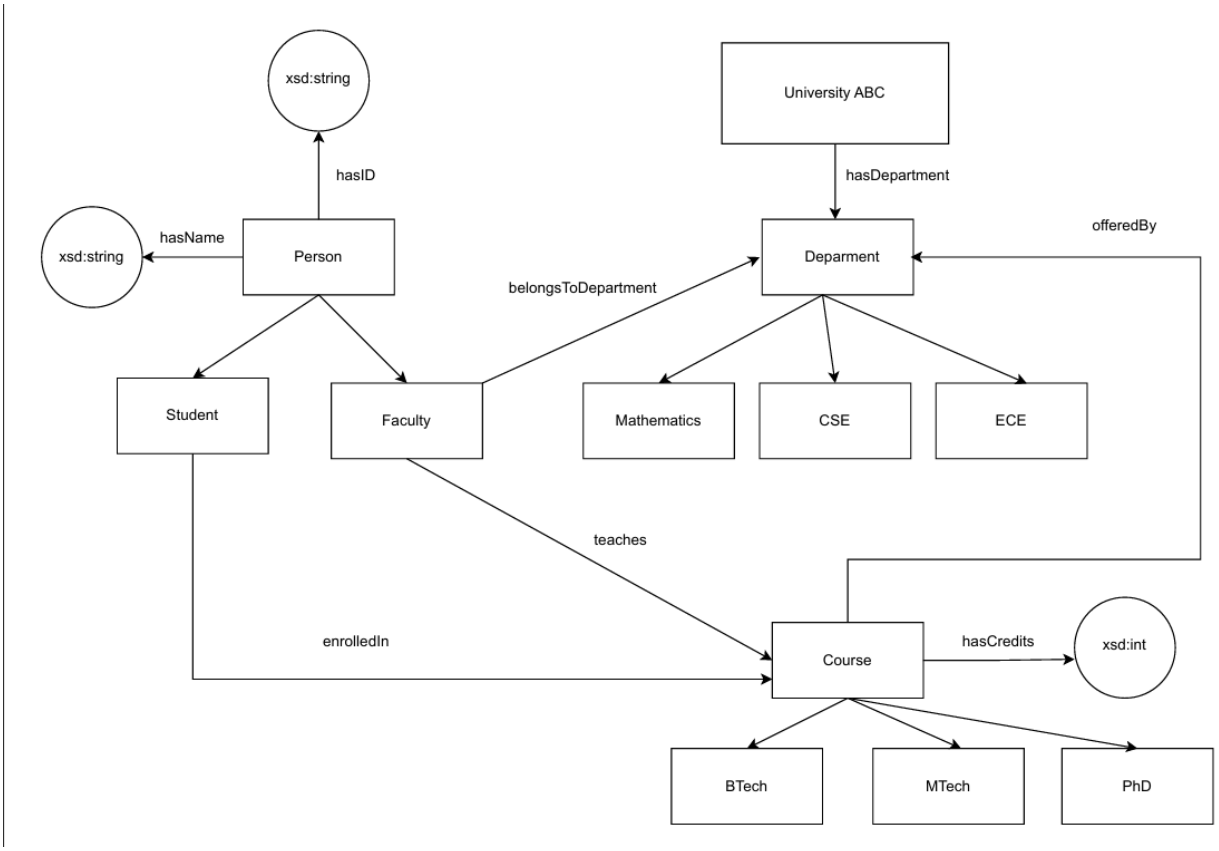
TreeSHAP resulted in an average of 8.00 non-zero features per instance, while KernelSHAP averaged only 2.00, indicating that KernelSHAP produces much sparser explanations. However, when analyzing stability across three repeated runs for a single instance, TreeSHAP demonstrated consistent SHAP values with negligible variation between runs. In contrast, KernelSHAP showed variability in SHAP values, with certain features receiving non-zero importance. This difference is attributed to the nature of the two methods: TreeSHAP is model-specific and leverages the internal structure of tree-based models to compute exact Shapley values deterministically, making it highly stable. KernelSHAP, on the other hand, is model-agnostic and relies on perturbation and sampling, making it inherently stochastic and more sensitive to the choice of background samples. As a result, TreeSHAP provides more reliable and reproducible feature attributions, whereas KernelSHAP offers more compact but less stable explanations.

QUESTION 3: In jupyter notebook



This is the snippet of the grad cam output for one of the images for the initial convolutional layers. In the early layers such as conv1, the model primarily focuses on detecting low-level features like edges, textures, and color gradients, evident from the broad activation around the penguins' outlines. As we move deeper into the network, the activations become more focused and structured, highlighting specific object parts such as the bodies and heads of the penguins. These mid-level layers capture more abstract and semantically meaningful features necessary for identifying object categories.

QUESTION 4:
SCHEMA:



Axioms in Description Logic Notation

Class Hierarchies

- BTech ⊑ Course
- MTech ⊑ Course
- PhD ⊑ Course
- CSE ⊑ Department
- ECE ⊑ Department
- Mathematics ⊑ Department

- Student ⊑ Person
- Faculty ⊑ Person
- Course ⊑ University

Object Properties

- enrolledIn: Student → Course
  Functional(enrolledIn)
  Student ⊑ (=1 enrolledIn.Course)
- teaches: Faculty → Course
  Functional(teaches)
  Faculty ⊑ (≤1 teaches.Course) ⊓ Faculty ⊑ (∃ teaches.Course)
- belongsToDepartment: Faculty → Department
  Functional(belongsToDepartment)
- offeredBy: Course → Department
  Functional(offeredBy)
- hasDepartment: University → Department

Data Properties

- hasName: Person → xsd:string
- hasID: Person → xsd:string
  Functional(hasID)
- hasCredits: Course → xsd:int

Design Decisions

The Course and Department subclasses allow extension and fine-grained reasoning about academic structure.
Properties are used for enrolledIn, teaches, and belongsToDepartment to enforce unique associations.
There are existence quantifier such that every student is enrolled in some course or faculty teaches some course.
Cardinality 1 is used on enrolledIn and teaches for unique course enrollments and teaching assignments per semester.
offeredBy and belongsToDepartment capture the organizational structure of departments, courses, and faculty.