Timeline

| Date | Agenda | Homework |
| --- | --- | --- |
| Thursday 29th August (Week 5) | Discuss and confirm topic |  |
| Thursday 5th September (Week 6) | * Teammates familarise with MOA metadata, EDA (done before meeting, 30mins) * Teammates each learn and present meaning of code 30mins * Lijia introduces suitable deep learning model 30mins * Confirm approach | * Run code * Find usable data set and topic   + Explored MOA, mushrooms (would be nice to keep going in these directions with previous knowledge) * Conduct EDA to justify direction |
| Tuesday 10th September (Week 7) | Confirm data set and topic - shelter animal outcome project  Github for codes - <https://github.sydney.edu.au/open0051/OptiPaw> | Write proposal (project timeframe, proposed methodologies)  [Research Project Proposal.docx](https://docs.google.com/document/d/1oXyRhQOopjeU0Ul_pHiyBPwkBp1J9c9g/edit?usp=sharing&ouid=104860681190284120005&rtpof=true&sd=true)   * ~~Introduction (Olivia)~~ * ~~Project aims (Olivia)~~ * ~~Project time frame~~ * ~~Proposed methodologies (YC)~~   Code  YC, CY - learn new codes  Olivia - data cleaning, visualizations (plotly) |
| Sunday 15th September (Week 7)  5:00pm | Progress Report 1  [Research Project Proposal.docx](https://docs.google.com/document/d/1oXyRhQOopjeU0Ul_pHiyBPwkBp1J9c9g/edit?usp=sharing&ouid=116963608304276472247&rtpof=true&sd=true) |  |
| Friday 20th September 6-8pm  (Week 8) | Census date to discontinue from project | Data re-cleaning (Monday)   * Mistakes of merging corrected - Olivia * Remove all missing values (can try check with upset plot) - Olivia * Check for categorical imbalance → balance sampling (tbc) - Chloe (Tues-Wed)   Data visualization (Tues-Wed)   * Apply suggestions - Olivia   Data processing (Tues-Wed)   * Plan out transformations per attribute (ie circular time) - YC   Start shiny app learning - everyone |
| Wednesday 25th September 6:30-8:30pm  (Week 9) |  | Olivia:   * ~~Box plot visualization~~ * ~~Upload visualization codes into drive/github~~ * ~~Logistic regression (cross validation, with and without PCA, evaluation function by YC)~~ * ~~KNN (cross validation, with and without PCA, evaluation function by YC)~~   LY: Interaction effect, nested CV,  Recommendation system.  Chloe:   * Balanced resampling (with comments) * BIC/AIC * Neural network * Regression model |
| Sunday 6th October 7:30pm-9:30pm (Mid-Sem Break) | ML model updates   * PCA - Cannot combine train and test data sets (use lijia’s new one) * Use austin data set only - split train and test based on intake date (before 2023 train, after 2023 test) * Cross validation (austin training set only) * Try data with 1 animal type, then all animal types   Split data set into 5 folds → 5 values per score → average score per score type (5 total)  Repeat process 10 times → obtaining 10 averages per score → for each score, plot 10 averages to boxplot → 5 boxplots (one for each score) for 1 ML model  Recommendation system   * Using lijia’s model * Replace onehotencoder with YC’s preprocessing function * Optimize running time   Dropping recommendation system or shiny app? - drop shiny app   * Visualizations * Outcome predictions * Name recommendation system   Remove outliers of comparative boxplot visualization to extend all boxplots to scale for clarity | Lijia: send all scripts to the group.[done] Read the EDA report of data, and start to think how to organize the structure of presentation. [done] Test datasets on Dog or Cat and also on kaggle data.  Olivia   * Edit comparative boxplot visualization * Add visualization html file to drive/github * Edit ML models (austin only, new pca, cv scoring and boxplot) * Run recommendation system   Chloe:   * Run the recommendation script |
| Friday 11th October  6-8pm  (Week 10) | Progress Report 2   * Shiny app workflow instead of actual one | * Multiclass Logistics Regression (Olivia) * Add in multiclass logistics and KNN to scoring box plot and dot plot code (YC) * Fix prediction models, kNN, space issue (YC) * Analyze popular pet names (chloe) * Slides (everyone) * Write and finish report (everyone) * Small workflow of how to build shiny app with python code (Lijia) [done]   Public repository: <https://github.com/oliviapeng26/OptiPaw> |
| Sunday 13th October  5-7pm  (Week 11) | ML figures  Boxplot:   * Sub-boxes (non-pca, pca) for each ML model * Interpret meaning of metrics and why for each model   Final prediction score:   * Make a table instead of scatter plot   XG boost bar graph  Recommendation   * Remove preprocessing inside each function to reduce running time * Experiment more with kNN to match TF-IDF | Everyone:  Evaluation form  Attendance form  **SLIDES on canva:** [**https://www.canva.com/design/DAGT579KtXQ/vXDpBclaLYJ64DwFxMrUDQ/edit?utm\_content=DAGT579KtXQ&utm\_campaign=designshare&utm\_medium=link2&utm\_source=sharebutton**](https://www.canva.com/design/DAGT579KtXQ/vXDpBclaLYJ64DwFxMrUDQ/edit?utm_content=DAGT579KtXQ&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton)  YC:  Edit ML figures (box plot for CV, table of final scores)  XG boost bar graph  Edit recommendation system code  Chloe:  Workflow graph for shiny app  Olivia:  Interpret box plot of CV, table of final scores, XG boost bar graph  OptiPaw logo |
| Monday 21st October  5-7pm  (Week 12) | Check slides  Presentation run-through |  |
| Thursday 24th October 6-9pm (Week 12) | Presentation Night |  |

EDA tools    
Have a look from page 14 to 27 : <https://docs.google.com/presentation/d/1T8fF-g9sJl_nHyA0g_lg-lOuyRpZoEON/edit?usp=sharing&ouid=116963608304276472247&rtpof=true&sd=true>

New scatter plot:

X - intake date

Y - outcome date

Each dot is one animal

Color by different attributes - make 1 scatter plot per attribute, resulting in 5+ sets

Box plot:

For every outcome type, make a box plot on the distribution for length of time in shelter

Codes to be learned (week 7)  
  
CY - <https://www.kaggle.com/code/fsmithus/reduced-model/report>  
YC - <https://www.kaggle.com/code/ayushnitb/naive-multiclass-classification-approach-h2oauto>

[Naive Multiclass-Classification Approachto](https://docs.google.com/document/u/0/d/1Qr1w33zEj_4p46Mqxw_vU2oTY9KUjmQeZeseMMw5UVY/edit)

Codes to be learned (week 5)

Olivia: <https://www.kaggle.com/code/viktorurushkin/scaler-pca-cv-logistic-regression>

YC: <https://www.kaggle.com/code/garywei944/pca-lr-ridge-rf-nn-tuning-hyper-parameters>

[PCA, LR, Ridge, RF, NN tuning hyper-parameters](https://docs.google.com/document/u/0/d/1NuYcHz-y1Cbv8CV72x0SrTGYH_o9eVM0WJhoY-eyDZs/edit)

CY: <https://www.kaggle.com/code/dennyli/nn-with-k-folds>

Ideas

Search for: Tabular data, Image data

Predict poisonous mushrooms

* Model → use model → website to upload images/survey questions and apply model
* <https://www.kaggle.com/competitions/playground-series-s4e8/data?select=train.csv>
* <https://www.kaggle.com/datasets/earobinson/massachusetts-mushrooms>

Mechanisms of Action (MoA) Prediction

* gene expression and cell viability data

[Mechanisms of Action (MoA) Prediction | Kaggle](https://www.kaggle.com/competitions/lish-moa) (csv)

~~Predicting Molecular Properties~~

Can you measure the magnetic interactions between a pair of atoms?

* Predict scalar couplings

<https://www.kaggle.com/competitions/champs-scalar-coupling> (xyz, csv)

Novozymes Enzyme Stability Prediction

* develop a model to predict/rank the thermostability of enzyme variants based on experimental melting temperature data

<https://www.kaggle.com/competitions/novozymes-enzyme-stability-prediction> (csv, pdb)

~~Jane Street Market Prediction~~

* quantitative trading model to maximize returns using market data from a major global stock exchange

<https://www.kaggle.com/competitions/jane-street-market-prediction> (csv)

Optiver Realized Volatility Prediction

<https://www.kaggle.com/competitions/optiver-realized-volatility-prediction/data> (parquet, csv)

~~H&M Personalized Fashion Recommendations~~

* develop product recommendations based on data from previous transactions, as well as from customer and product meta data

<https://www.kaggle.com/competitions/h-and-m-personalized-fashion-recommendations> (jpg, csv)

Predict music trend

<https://www.kaggle.com/datasets/saurabhshahane/music-dataset-1950-to-2019> (csv)

<https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset> (csv)

* Each track has some audio features associated with it. The data is in CSV format which is tabular and can be loaded quickly.

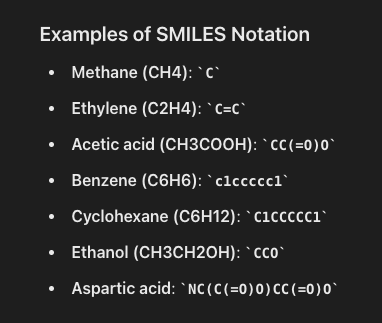
<https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation> (csv, json)

* Extract music used/gone viral on social media (reels, short videos, radio, artist mentions/comments, likes, shares, saves)
* Music videos, lyric videos views (youtube)
* Streaming services (spotify) number of listens
* Co-listening habits (artists, genres trending together)
* Music awards
* Concert ticket sales
* Cultural events (ie superbowl, olympics, music festivals)
* Time-series analysis, natural language processing, clustering and classification, deep learning for audio analysis

Geospatial - predict disasters and outcomes

* <https://www.kaggle.com/datasets/cathetorres/geospatial-environmental-and-socioeconomic-data>
* A country or the world
* Classify types of disasters and probability of occurrence
* Wild fires - forest layout
* Areas prone to disaster - population density, infrastructure quality, historical disaster
* Time series analysis
* Evacuation planning (GPS data that tracks movements, most likely movements)

Organic chemistry reaction pathways

* Predicting possible reaction pathways for organic molecules
* Molecular Representation and Feature Extraction
  + SMILES Simplified Molecular Input Line Entry System <https://www.kaggle.com/datasets/yanmaksi/big-molecules-smiles-dataset>
* Probability of different reaction outcomes, competing reactions
  + Sequence-to-Sequence (Seq2Seq) Models - reactants as input sequences
  + Graph neural networks (GNNs) - captures spatial structure of molecules
  + Traditional Machine Learning Models (Random Forests, SVMs)
  + Generative Models (e.g., VAEs, GANs) - explore new reaction pathways, predict a distribution of possible outcomes.
  + Reinforcement Learning - optimizing reaction pathways or conditions through iterative decision-making processes

Data sets

Financial Fraud Detection Dataset

* Kaggle : <https://www.kaggle.com/datasets/sriharshaeedala/financial-fraud-detection-dataset>
* <https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud>

World university Rankings

* <https://www.kaggle.com/datasets/alitaqi000/world-university-rankings-2023>

Music Trend over the years

* <https://www.kaggle.com/datasets/saurabhshahane/music-dataset-1950-to-2019>

Fun Spotify  
 - <https://www.kaggle.com/datasets/maharshipandya/-spotify-tracks-dataset>

- <https://www.kaggle.com/datasets/bricevergnou/spotify-recommendation>

Sign Language

* <https://www.kaggle.com/datasets/harshvardhan21/sign-language-detection-using-images>

Uni admission

* <https://www.kaggle.com/datasets/amanace/student-admission-dataset>

Netflix recommendation

* <https://www.kaggle.com/datasets/shivamb/netflix-shows>

Chat gpt vs humans

* <https://www.kaggle.com/datasets/mahdimaktabdar/chatgpt-classification-dataset>

Kidney transplant failure genes

* <https://shiny.maths.usyd.edu.au/PROMAD/>

Deep learning unit: [https://www.dropbox.com/scl/fo/8xb81npms2h6h2ml8l2g5/h?rlkey=hz7orq5ztekgmsdlargat2619&st=yrgo63j6&dl=0](https://url.au.m.mimecastprotect.com/s/54F1CwV1vMf0KMQo1FVfMuJM9z9?domain=dropbox.com)

Deep learning 101

Perceptron - if sum of X\*W > threshold, perceptron will trigger and send signal.

* Sum of X\*M < threshold or wx - b < 0 → output y = -1
* Sum of X\*M > threshold or wx - b > 0 → output y = 1

Perceptron linear equation:

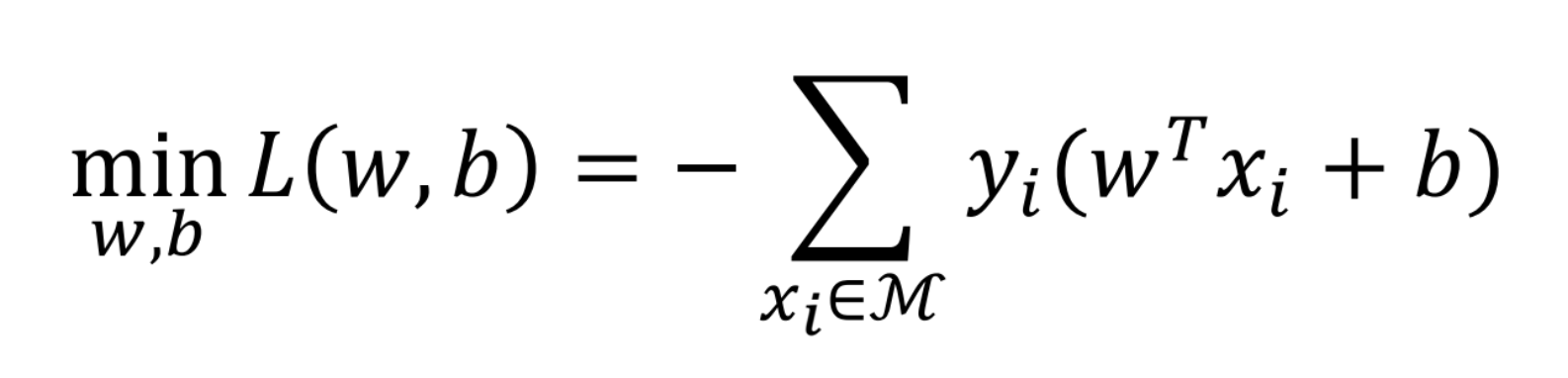
y = transpose(w) \* vector(x) + b

* transpose(w) - weight vector
* vector(x) - input vector
* b - bias

Comparing between true and predicted (multiplying true and predicted)

* y\*(transpose(w) \* vector(x) + b) > 0 - good model
* y\*(transpose(w) \* vector(x) + b) < 0 - bad model

Objective function of perceptron to calculate minimum loss, M for misclassified examples (perceptron equation < 0)

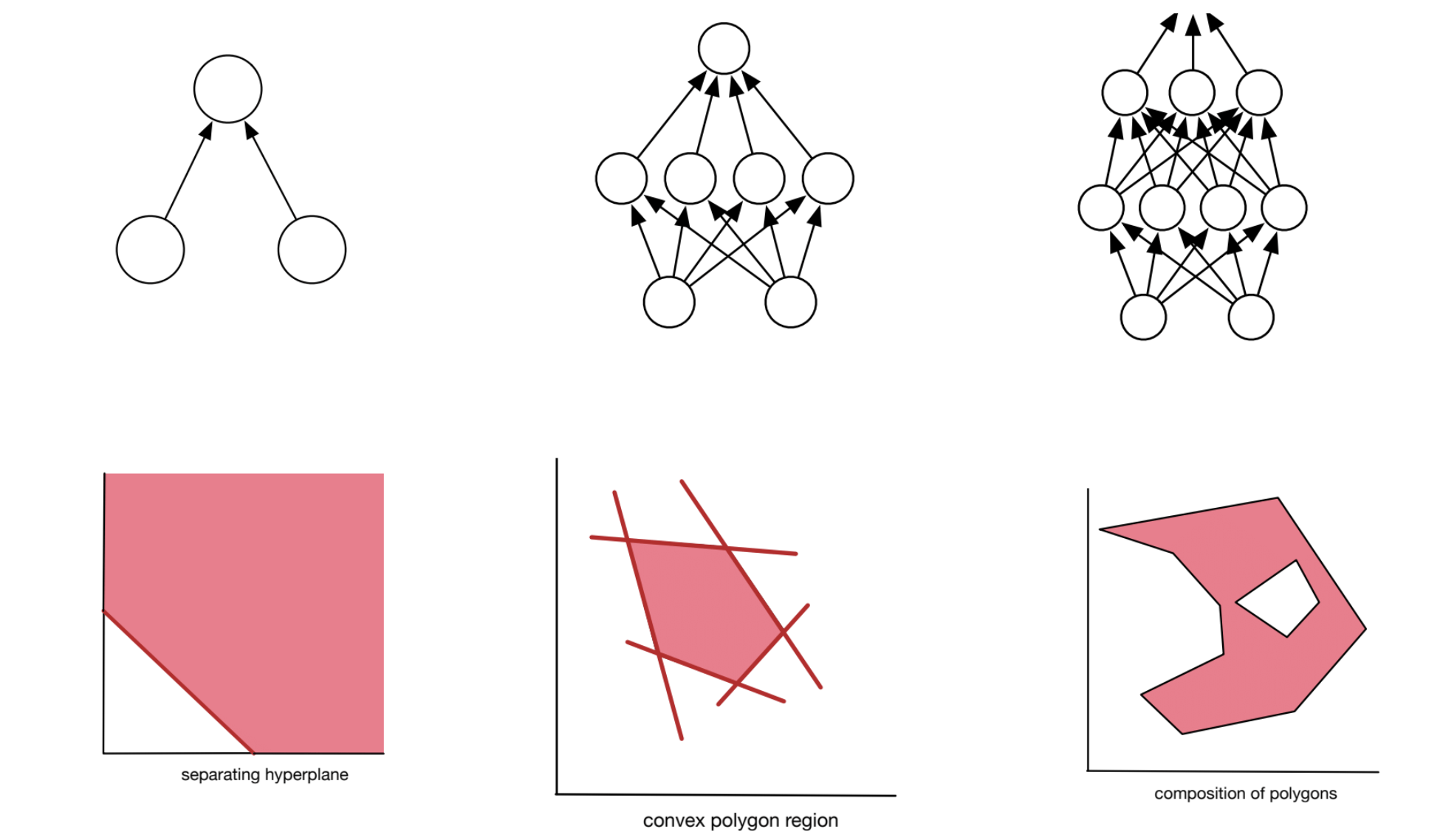


minL produces parabolic values, choose minimum

Apply negative sign to turn negative summation output into positive, therefore greatest positive loss is better than most negative loss hence the best model

Multilayer neural network - Each layer has their own perceptron equation, output becomes input of the next layer

Purpose of multilayer



Activation function - function that defines the activation of the next layer when threshold exceeded

* Nonlinear function
* i.e. Relu function

Some Datasets

Amazon review using NLP - <https://www.kaggle.com/datasets/kritanjalijain/amazon-reviews/code>  
  
2018 Data - <https://cseweb.ucsd.edu/~jmcauley/datasets.html#amazon_reviews>

Bbc Classification - <https://www.kaggle.com/c/learn-ai-bbc>  
  
<https://www.ml4devs.com/articles/datasets-for-machine-learning-and-data-science/>

Genomics of Drug Sensitivity in Cancer (GDSC)

<https://www.kaggle.com/datasets/samiraalipour/genomics-of-drug-sensitivity-in-cancer-gdsc?select=GDSC2-dataset.csv>

* GDSC2\_dataset.csv - drug used for cancer, its pathway, dose, and effectiveness
* Cell\_Lines\_Details.xlsx - cell line samples, changes to gene in cell line (binary Y/N)

Age prediction using genomic data

<https://www.kaggle.com/datasets/thedevastator/age-prediction-for-individuals-using-multi-omic>

* 20 genes from nine different samples used to predict the age of an individual using machine learning methods

Multi-Class Prediction of Obesity Risk

<https://www.kaggle.com/competitions/playground-series-s4e2/data?select=train.csv>

Crime prediction

San francisco - <https://www.kaggle.com/competitions/sf-crime>

South Aus - <https://www.kaggle.com/datasets/kanchana1990/south-australia-crime-data-2022-2023>

NSW (2010 - 2024 data) - <https://data.gov.au/dataset/ds-sa-860126f7-eeb5-4fbc-be44-069aa0467d11/details?q=crime>

* Use 2010 - 2022 data to predict 2023-2024 crime outcomes
* Map visualizing crime hotspots
* Is an area more likely to experience a type of offense than another?
* <https://bocsar.nsw.gov.au/>

Shelter animal outcomes upon leaving animal shelter

<https://www.kaggle.com/competitions/shelter-animal-outcomes> (not image data, description of pets)

Indiana, US (2017): <https://data.world/city-of-bloomington/94d3f457-57b5-45be-bee0-a0106f59b7ed>

Austin, US (2018): <https://data.world/siyeh/austin-animal-center-live-data> (original version of kaggle dataset, add “.csv” to the end of each file to get the csv file)

California, US: <https://data.longbeach.gov/explore/dataset/animal-shelter-intakes-and-outcomes/export/?disjunctive.animal_type&disjunctive.primary_color&disjunctive.sex&disjunctive.intake_cond&disjunctive.intake_type&disjunctive.reason&disjunctive.outcome_type&disjunctive.outcome_subtype&disjunctive.intake_is_dead&disjunctive.outcome_is_dead>

* Similar to mushroom idea: create website with survey questions that allow audience to enter description of any pet within range → predict outcome of that pet
  + Something like this: <https://www.shelteranimalscount.org/>
* Visualization about pet names, breed
* Randomly generate a name for the pet based on gender, breed and color

Data sets available to be used:

1. Kaggle dataset - train-outcome.csv, test-outcome.csv
2. Austin dataset - intake.csv, outcome.csv
3. California dataset - intake&outcome.csv
4. Indiana dataset - intake&outcome.csv

Machine learning

Train 2 3 4 - test 1 test.csv

Train 2 3 - test 4

Train 3 4 - test 2 combined stripped of outcome

(but predicted can be compared to combined.csv “mark scheme”)

Train 2 4 - test 3

Data visualizations

Combine all 2 3 4 - visualize together

Task

Combining both austin, intake and outcome.csv

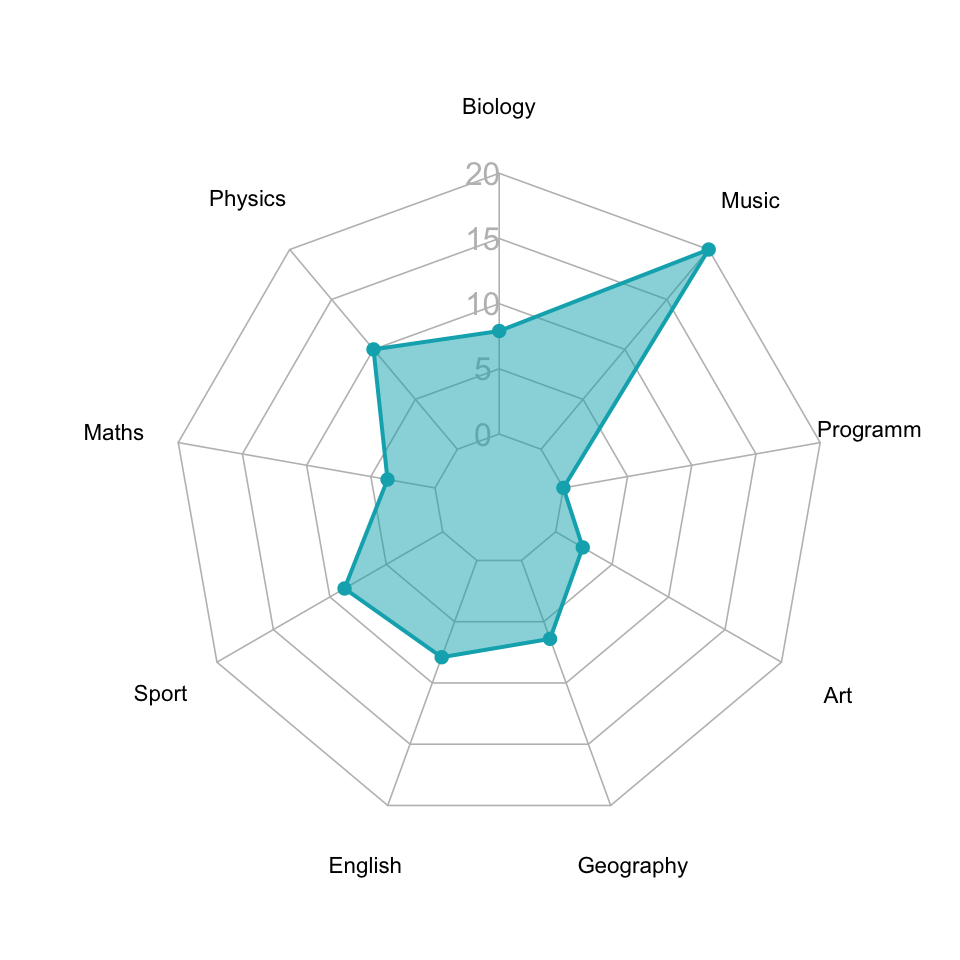
77u7

Data visualization options: <https://r-graph-gallery.com/time-series.html>

Time of intake, time of outcome, time in shelter (months)

3 areas - Austin, California, Indiana

IV - intake type, animal type, gender, age, breed, color, (name), time in shelter (months)

DV - outcome type 

1. Pie chart - showing how each IV affects the DV
2. Radar chart - showing how strongly each IV affects the DV
3. Bubble plot
   1. X - intake type
   2. Y - outcome type
   3. Color - animal type
   4. Size - region
4. Color-coded scatter plot for time in shelter as color
5. Time-series scatter plot for Popularity of adoption by month
6. Scatter plot for Time in shelter vs time of death/death as an outcome
   1. Color coded by animal type
   2. Different lines for different animal types
7. Pie chart - most popular names marked out

Data Modelling

Cross Validation (train+validation, nested CV (grid search of parameters) ) - done for every model, 80% training, 20% validating

**Data Engineering (share code for everyone)**

PCA - olivia

Balance Sampling (Chloe)

Backward/Forward BIC (Chloe)

**Methods (try with and without PCA):**

XGBoost/GBM/lightGBM (YC)

Neural Network (no need PCA) (Chloe, YC)

Logistic Regression - olivia

Linear Regression (Chloe)

Random Forest (YC)

K N N (K Nearest Neighbors) - olivia

S V M (Support Vector Machines) (YC)

\* Can consider Ensemble/Stack

*Not Priority*

* Grid Search
* Featured Engineering - check this post, <https://www.kdnuggets.com/2018/12/feature-engineering-explained.html>

Steps:  
1. PCA -> model with CV (80-20)  
2. Without PCA, use all features - > model with CV (80-20Steps:

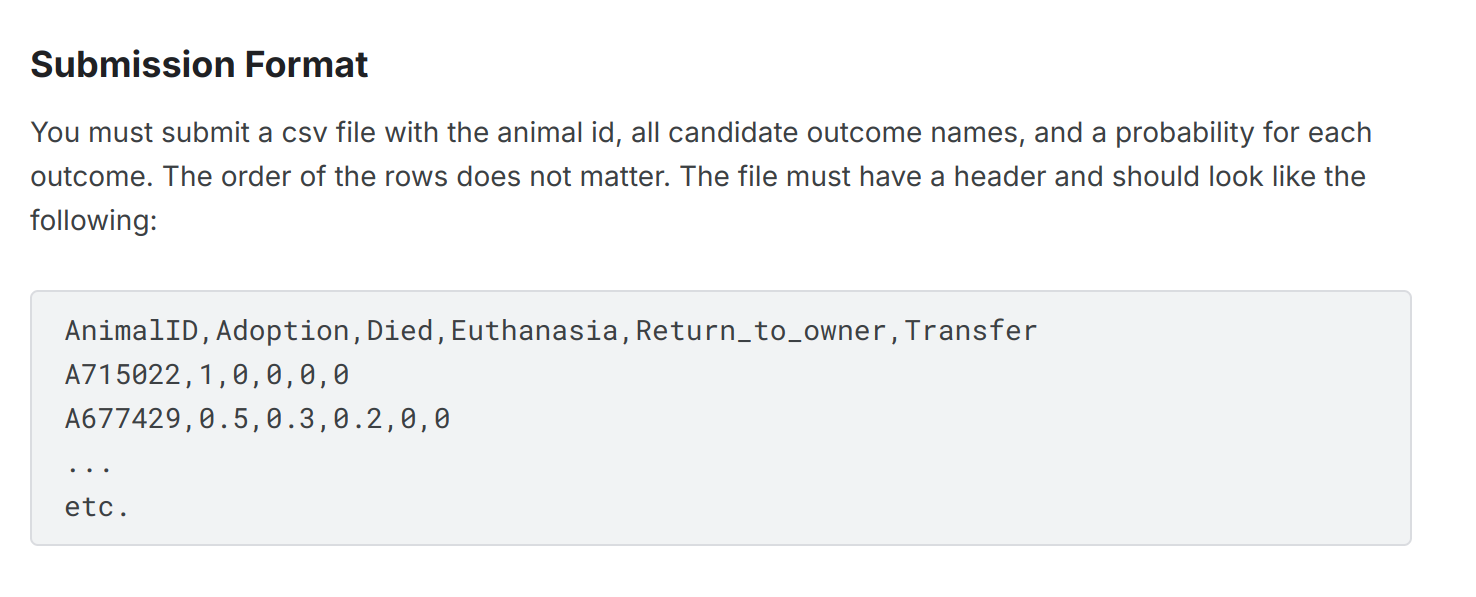
1. PCA -> model

2. Without PCA, use all features - > model

3. Evaluation result of each model)

3. Evaluation result of each model (CV graph boxplot,final test evaluation result)

**Evaluation**

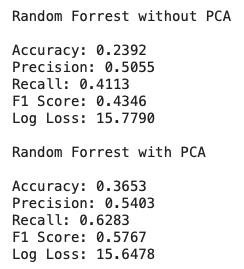


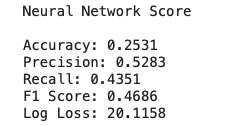
Accuracy, Precision and Recall, F1, Log loss

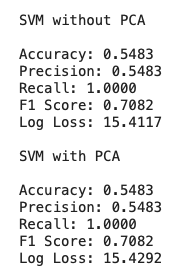
<https://www.kaggle.com/competitions/shelter-animal-outcomes/overview>

USE THIS FORMAT (Probability)

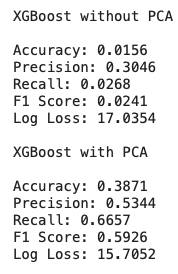
YC







\*SVM + PCA is fastt



Nested CV  
  
<https://scikit-learn.org/stable/auto_examples/model_selection/plot_nested_cross_validation_iris.html>

you **should not perform PCA** on the combined train and test data together. PCA (Principal Component Analysis) is a form of dimensionality reduction, and like other feature transformations (e.g., scaling, normalization), it should only be **fitted on the training data** and then applied to both the training and test sets separately.

Here’s why:

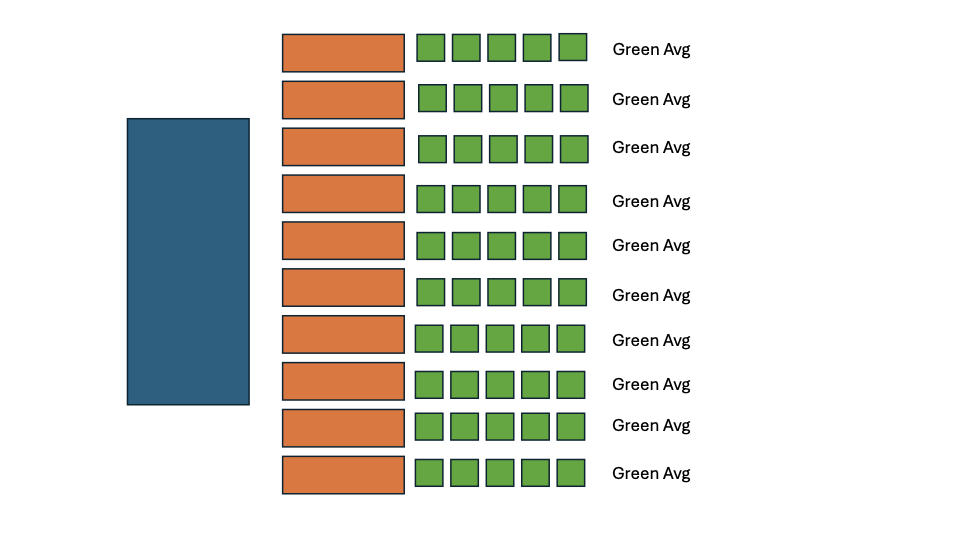
1. **Data Leakage**: If you fit PCA on both the training and test data together, the test data influences the PCA transformation. This introduces data leakage, meaning that information from the test set is used during the training process, which could artificially inflate performance metrics.
2. **Proper Generalization**: The PCA transformation should learn patterns only from the training set. The test set should remain unseen and only be transformed using the PCA model fitted on the training data.

### **Correct Steps for Applying PCA:**

1. **Fit PCA on the training data**: Learn the principal components using only the training data.
2. **Transform both the training and test data**: Use the learned principal components to transform both the training and test sets.

Cross validation - Box plot (only with training dataset) [use 5 for the models]

Test > 2023, train < 2023   
  
Perform cross validation on the train - 5 cross validation (output scoring function on train in box plot)



YC Reference use 10 green avg for a box plot

**Recommendation system methods**

**Cosine similarity**: finds whether 2 vectors (input, train vectors) are pointing towards a similar direction by measuring the cosine of the angle between 2 vectors

* Range: -1 to 1, -1 means vectors are 180° to each other (completely different), 1 means vectors are identical, 0 means vectors are orthogonal (no similarity)

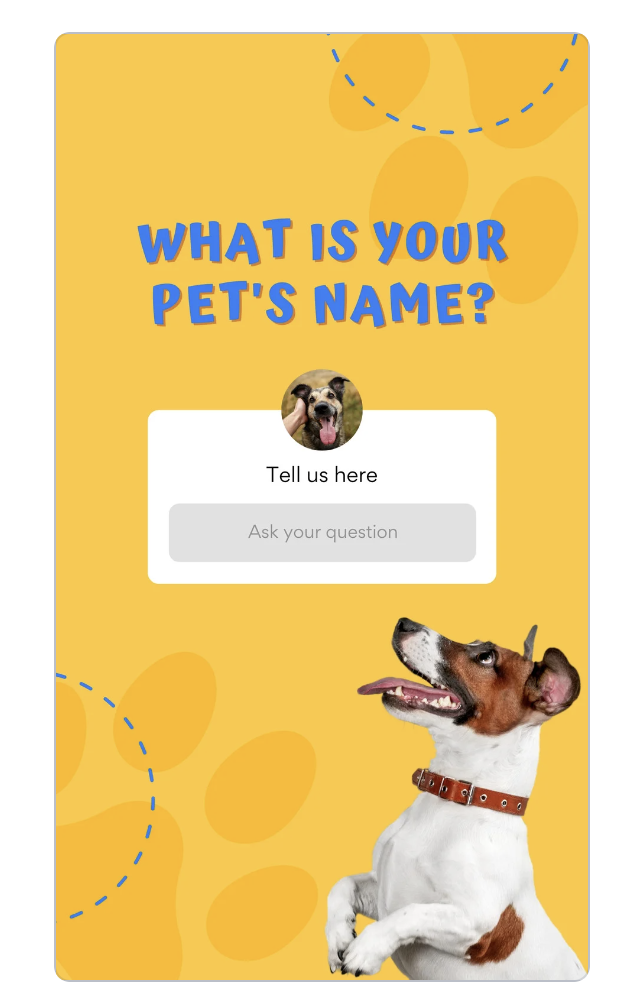
**TD-IDF encoding**: weighting system assigning weight to each word in a document based on its term frequency (tf) and reciprocal document frequency (idf) and multiplying them together to obtain TD-IDF

* TF - frequency, IDF - rarity
* Transforms text into numeric (i.e. each document represented by a vector where each element corresponds to the TF-IDF value of that word)
* higher TF-IDF score, more "important" that word is in representing the document

What if the user doesn’t know the answer to a specific survey question? (i.e. breed of animal)

**Presentation Points**:

8 MINUTES total

1. Introduction to the problem and datasets (30 sec)
2. EDA (What can we learn from the EDA? Imbalance, missing values, proportions, relevant features? What are the common names per breed, per pet type?, **how many pets types in your datasets?**) (1 min 30 sec)
   * Other visualizations (3)
3. Feature Engineering for Categorical Features - preprocessing function (YC can explain his method for feature engineering). (30 sec)
4. Prediction model and results (3 min)
   * Purpose of prediction, PCA, repeated 5-fold cross validation
   * LogReg, kNN, random forest, SVM, XGBoost, neural network
   * Explain scores and findings (best model, evaluation metrics)
5. Name recommendation (1 min)
   * Cosine similarity
   * TF-IDF
   * kNN - Types of distances
6. Demo of name recommendation (1 min 30 sec)
   * Distribute pictures of pets and their attribute values. Let the audience come up with names and compare with our system’s name.
   * Our system prevents predictions of most common names.

PCA (Principal Component Analysis) is generally not necessary before using Random Forest or XGBoost. Both algorithms are capable of handling high-dimensional data and can effectively manage correlated or redundant features.

Here’s why PCA is usually not needed:

Random Forest:

It is a tree-based algorithm that performs well without feature scaling or dimensionality reduction.

Random Forest selects the most important features during its splitting process, making it robust against high-dimensional data and irrelevant features.

XGBoost:

Like Random Forest, XGBoost is a tree-based method and doesn't require feature scaling or dimensionality reduction.

XGBoost includes built-in regularization techniques that handle overfitting and can manage a large number of features.

When might you consider PCA?

If you have very high-dimensional data (e.g., thousands of features) and want to reduce computational cost or speed up training.

If you suspect there’s a lot of noise or redundant features that could negatively impact the model performance, though Random Forest and XGBoost usually handle this well.

If you plan to use linear models or other algorithms that are sensitive to multicollinearity (like logistic regression or SVM), PCA can be helpful.

Workflow of Shiny app

graph TB

Planning[1. Planning and Requirements Gathering] --> Environment[2. Setting Up the Development Environment]

Environment --> DataPrep[3. Data Preparation]

DataPrep --> Integration[4. Integrating Python with R using reticulate]

Integration --> Structure[5. Structuring the Shiny App]

Structure --> UIDesign[UI Design]

Structure --> ServerLogic[Server Logic]

UIDesign --> PetRecom[6. Developing the Pet Recommendations Tab]

ServerLogic --> PetRecom

UIDesign --> DataViz[7. Developing the Data Visualization Tab]

ServerLogic --> DataViz

PetRecom --> Testing[8. Testing and Debugging]

DataViz --> Testing

Testing --> Enhancements[9. Enhancements and Optimizations]

Enhancements --> Deployment[10. Deployment]

