Final Project Mid-point Check-in

Group Members: Sean Lewis (shl225) **Project Title:** Steam Price Projection

Overview of Accomplishments So Far (Approx. 80% Complete)

We've built a full Python pipeline that loads and cleans data on about 65,000 Steam games, created ~67 game features without accidentally using future information (no data leaks from features like "Value For Money", "Free Game", etc.) and implemented from-scratch versions of Ridge Regression and Gradient Boosting using Decision Trees (GBDT) with stratified K-Fold cross-validation (CV) to evaluate them. We also added model saving and loading, feature importance analysis, and price elasticity/discount estimation with a test notebook showing loading of our best model and using it for prediction and analysis.

We have prototyped the interface for the Streamlit app with components based on our proposal diagram. It is currently barebones implementation, with the backend and frontend being separate for right now. We trained MLR (Ridge) and GBDT models on both direct price and log-transformed price targets. We added MAE and Price Bucket Accuracy to our evaluation metrics and evaluated the models using RMSE, MAE, MAPE, R^2, and Price Bucket Accuracy (±1 bucket). The GBDT model trained on log-price performed best, achieving a test set MAE of ~$4.81 (on the original price scale) and Price Bucket Accuracy (within ±1 bucket) of ~87.2%. Log-transformation significantly improved performance over direct price prediction. Top drivers of price included review counts, owner estimates, achievement data, and flags for certain genres/tags (like indie or simulation).

Our pricing analysis gives estimates for game-specific elasticity and recommended discounts as well as simulations showing potential revenue impact of price adjustments (average optimal adjustment was -5% for the test set). The backend is mostly complete with the best model being trained and saved for future use and potential finetuning. Logic for feature engineering new inputs, loading model, making predictions, making pricing/discount insights (like elasticity, optimal price, sale price) is implemented and verified in the test notebook. The largest remaining task is integrating backend logic into the prototype Streamlit front-end.

Revised Scope and Goals

We found some issues when implementing our project: skewness in Steam's price distribution, data leakage by certain features, and measuring model performance accurately. Our goal of predicting optimal Steam game prices/discounts is still the same, but our methodology improved a bit to address these issues as we added more evaluation metrics and preventions for data leakage and skew. This is done by implementing stratified K-Fold CV, identifying problematic features and using leak-free methods for elasticity and discount recommendations, as well as including log-transformed prices due to the data skewness, which significantly improved model accuracy.

Next Steps

Our next steps are connecting the backend (model, prediction/analysis functions) to the Streamlit frontend, thoroughly testing/validating the app with variety of game examples (different genres, price ranges, outliers), and verifying end-to-end use. Users should be able to input game metadata and get model's prediction for price and suggested discount. Finally, we need to perform additional confidence/logic checks on our model's outputs, professionalize the codebase with clear comments and complete the final project report.

Please see next page for References (Notebook link, images of UI, etc.)

References

Code/notebook link: https://github.com/shl225/paml-final

Photos of website user interface: