DS 5220: Supervised Machine Learning and Learning Theory

Iteration #05 - Final Project

Ikonkar Kaur Khalsa and Matthew Gregorio

Deadline: 07/19/2024, 11:59 PM

Results and Discussions

1. **Present your results in a clear, organized manner: Use tables, graphs, and charts to visualize key findings. Include both quantitative metrics and qualitative observations. Provide examples of model outputs or predictions where relevant.**

   We have several quantitative metrics and qualitative meaures, as well as, visualizations throughout our code. Here are some examples, however all our metrics and visualizations can be seen in the jupyter file:

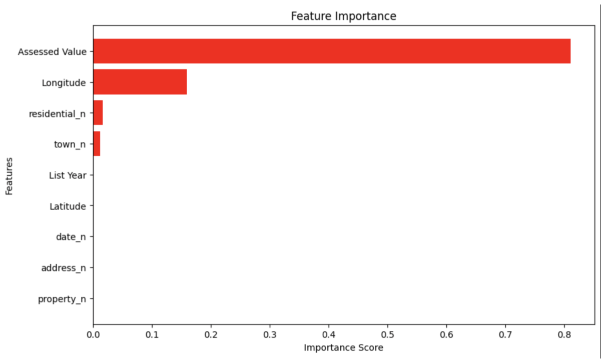
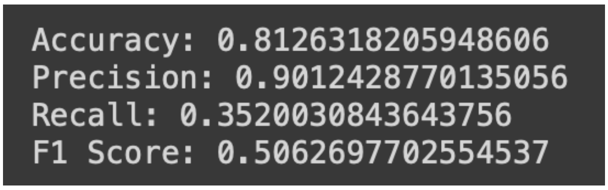


Figure 1: Example of Visualization



```
Accuracy: 0.8126318205948606
Precision: 0.9012428770135056
Recall: 0.3520030843643756
F1 Score: 0.5062697702554537
```

Figure 2: Example of quantitative data (accuracy scores of decision tree model

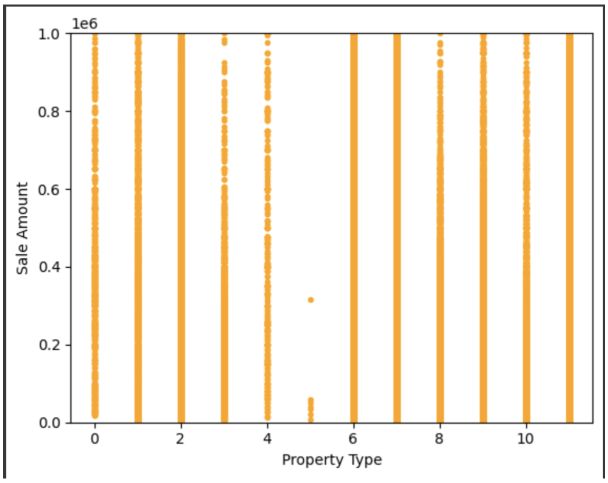
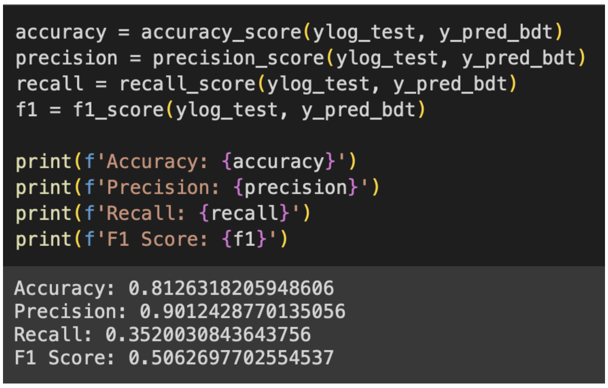


Figure 3: Example of Qualitative Data

2. **Analyze your results in depth: Compare performance across different models or approaches you tried. Discuss how your results align with or differ from initial expectations. Identify patterns or trends in your data and model performance.**

   We have provided two example of model performance. There are many more in the jupyter file that will be discussed in the technical report later on. In this case, decision tree model had better scores in all major metric categories (accuracy, precision, recall, f1-score). We thought neural networks would be effective for this dataset, so we were slightly surprised with the scores. In our exploratory data analysis that we created visualizations for every feature and its relationship to salary amount, we realized that assessed value and residential type feature heavily affected the data compared to every other feature. We then noticed after doing some feature importance analysis that longitude also had some significant affect on the data.

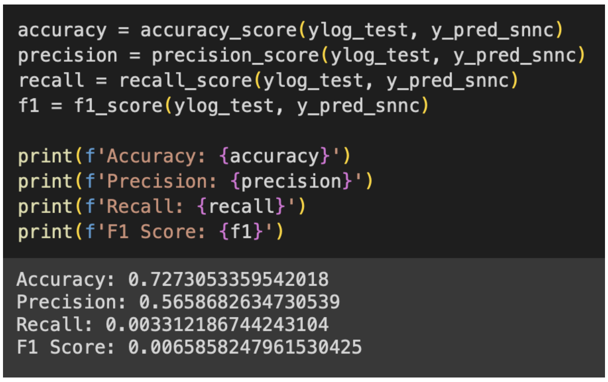


```
accuracy = accuracy_score(ylog_test, y_pred_bdt)
precision = precision_score(ylog_test, y_pred_bdt)
recall = recall_score(ylog_test, y_pred_bdt)
f1 = f1_score(ylog_test, y_pred_bdt)

print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1 Score: {f1}')
```

```
Accuracy: 0.8126318205948606
Precision: 0.9012428770135056
Recall: 0.3520030843643756
F1 Score: 0.5062697702554537
```

Figure 4: Decision Tree Model Scores



```
accuracy = accuracy_score(ylog_test, y_pred_snnc)
precision = precision_score(ylog_test, y_pred_snnc)
recall = recall_score(ylog_test, y_pred_snnc)
f1 = f1_score(ylog_test, y_pred_snnc)

print(f'Accuracy: {accuracy}')
print(f'Precision: {precision}')
print(f'Recall: {recall}')
print(f'F1 Score: {f1}')
```

```
Accuracy: 0.7273053359542018
Precision: 0.5658682634730539
Recall: 0.003312186744243104
F1 Score: 0.0065858247961530425
```

Figure 5: Simple Neural Network Scores

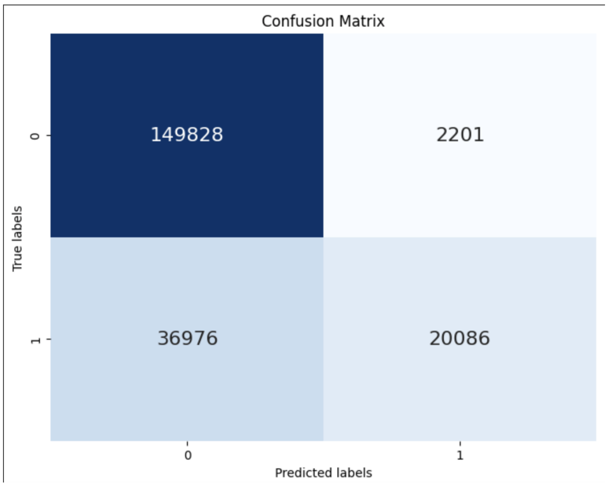
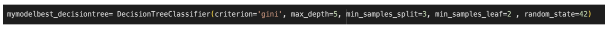


Figure 6: Here is an example of confusion matrix for the decision tree. Our model predicted that houses would be affordable 149,828 times and it actually was. It predicted it would be affordable 36,976 but it actually wasn't. It predicted that it wasn't affordable 2,201 but it actually was. It predicted it was not affordable 20,086 times, but it actually was not.

3. **Highlight significant findings: Emphasize any novel or unexpected results. Discuss the potential impact of your findings on the problem domain. Relate your results to existing research or industry practices.**

   One novel or unexpected result was that the Longitude was the second most important feature because in the exploratory data analysis it did not seem that it had much of a significance. There was no clear relationship visible in the EDA. Secondly, it was surprising to see that the neural network was not as effective compared to the other models. Neural networks are supposed to supposed to handle complex relationship, however this data does not seem to have a complex relationship.



```
mymodelbest_decisiontree= DecisionTreeClassifier(criterion='gini', max_depth=5, min_samples_split=3, min_samples_leaf=2 , random_state=42)
```

Figure 7: For the Best Decision Tree Model, we changed the hyper-parameters. We changed the max_depth, min_samples_split, and min_samples leaf. We did the grid search cross validation and found that these hyper- parameters values gave rise to the most accurate model of this family.

If you make more accurate predictions for sale amount then you can more consistently sell homes for the maximal price. if not underselling homes profit will be greater. this is a competitive advantage over competitors and is highly valuable for businesses competing for market share.

4. **Address challenges and limitations: Describe any technical obstacles you encountered during implementation. Discuss data-related issues (e.g., quality, quantity, bias) and their impact. Explain how you mitigated challenges or adapted your approach.**

   Some technical obstacles we encountered were missing values, filling in NaN values with numerical means, label encoding several categorical values, and normalizing all our features for more efficient model training. We had to remove duplicated in our data that were about 8000 which would have created a heavy bias, however we were able to tackle this issue. The quantity of our data prevented us from using several models such as SVM and KNN. We left out models that were not suited for our data. We fille in missing values with the mean, we removed duplicates, label encoded categorical values, and normalized our data for efficient and accurate model training.

5. **Reflect on the learning process: Discuss key insights gained about machine learning techniques. Describe how your understanding of the problem evolved. Suggest potential improvements or future work based on your findings.**

   Several models are used for different purposes and some worked well with our dataset and some don't. Regression is usually easy to calculate whether it is accurate or not but classifiers can have a difficult time in accurate predictions. Potential improvements: we can do a more in depth grid search and cross validation for decision tree.

Evaluation Matrix:

1. **Provide a comprehensive overview of your evaluation approach: Explain why you chose specific evaluation metrics for your project. Discuss how these metrics relate to the goals of your machine learning task.**

   We chose accuracy precision recall f1 score for classification as they measure the well the binary nature of predicted outcomes versus actual outcomes. We chose mean squared error mean absolute error and r squared value for regression since they measure residuals and correlation. This was best way to compare different classification and regression models for prediction power. These metrics inform us on which model will be most accurate to make predictions. We can choose the best predictive model as our featured model.

2. **Detail the metrics used in your evaluation matrix:Define each metric (e.g., accuracy, precision, recall, F1-score, ROC-AUC). Explain how each metric is calculated and interpreted. Discuss the strengths and limitations of each chosen metric.**

   Metric Definition

   Accuracy: Accuracy gets the ratio of correctly predicted instances to the total number of instances. It measures how well our model is performing.

   Precision: Shows the ratio of the instances that were corrected predicted as positive to the total predicted positive instances. Precision measures the accuracy of the positive predictions that were made by the model, which overall it focuses on the positive prediction quality.

   Recall: Shows the ratio of how many instances were predicted positive correctly to the actual total positive instances. It measures the model's ability to identify all of the relevant instances that were positive, which focuses on how the model is capturing the actual positive instances well.

   F-1 Score: This is a balance between precision and recall of the model. It gives a measure of accuracy of the model that takes in both precision and recall which is a great tool when handling datasets that are imbalanced.

   Metric Calculations

   Accuracy = # of correct predictions / # of total predictions

   Precision = TrP/(TrP+FP)

   Recall = TrP/(TrP+FN)

   F1 = 2(precision*recall)/(precision+recall)

   Strengths and Limitations

   Accuracy: Accuracy is great for its simplicity and gives a good measure of the overall performance of the model. On the other hand, it can be misleading with datasets that are imbalanced.

   Precision: Precision is great when indicating the quality of positive predictions. However, it precision does not consider any false negatives.

   Recall: It has the ability to capture all the actual positives. However, it does not consider any false positives.

   F1 Score: The F1-score is great for imbalanced datasets and balances both precision and recall. On the other hand, it can be not as intuitive when giving insights on false positive and negatives separately.

3. **Describe additional evaluation criteria: Explain how you assessed these qualitative aspects of your model.**

   During EDA we noticed visually a linear pattern with assessed amount and sale amount. We also noticed that property type 5 would yielded consistently lower sale amounts. No other qualitative patterns were discernible.

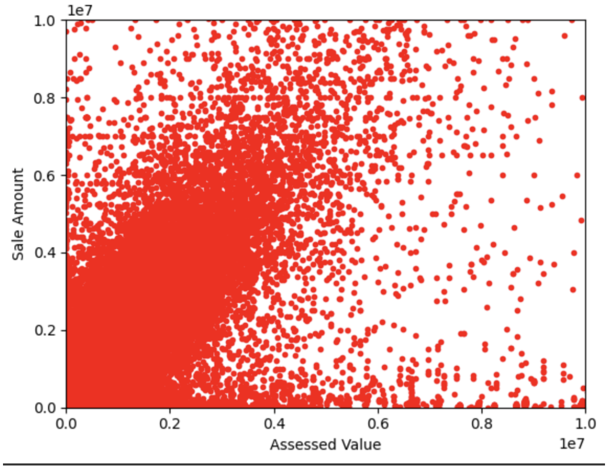
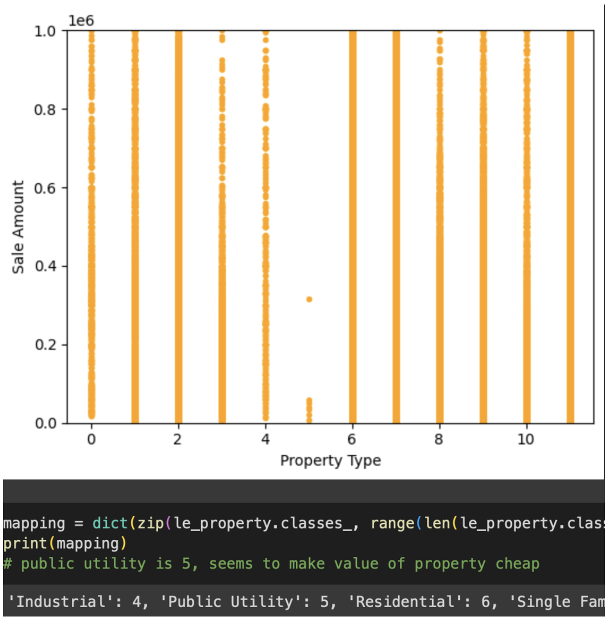


Figure 8: Assessed Value and Sale Amount



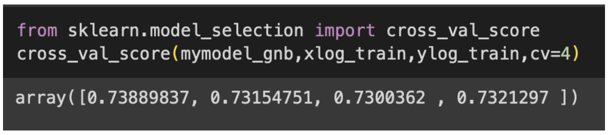
```
mapping = dict(zip(le_property.classes_, range(len(le_property.class
print(mapping)
# public utility is 5, seems to make value of property cheap

 'Industrial': 4, 'Public Utility': 5, 'Residential': 6, 'Single Fam
```

Figure 9: Here the Residential Type seems to have a lot of impact on Sale Amount than other property types.

4. **Outline the evaluation process: Describe your data splitting strategy (e.g., train/validation/test sets). Explain any cross-validation techniques used. Discuss how you ensured fair and robust evaluation.**

   Each training set was 80% and test set was 20%. for some models utilizing cross validation we split it 4-6 ways - which sets the number of subsets of the training data.

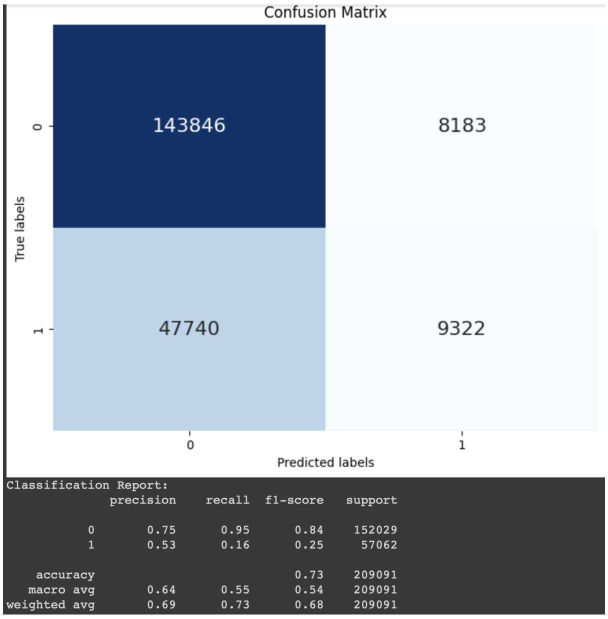


```
from sklearn.model_selection import cross_val_score
cross_val_score(mymodel_gnb,xlog_train,ylog_train,cv=4)
```

```
array([0.73889837, 0.73154751, 0.7300362 , 0.7321297 ])
```

Figure 10: GaussianNB Cross Val Score

5. **Present a structured evaluation matrix: Create a table or visual representation of your evaluation criteria. Include columns for metrics, descriptions, and target values or benchmarks. Show how different models or iterations performed across all criteria.**



```
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.95      0.84    152029
           1       0.53      0.16      0.25     57062

    accuracy                           0.73    209091
   macro avg       0.64      0.55      0.54    209091
weighted avg       0.69      0.73      0.68    209091
```

Figure 11: GaussianNB Confusion Matrix

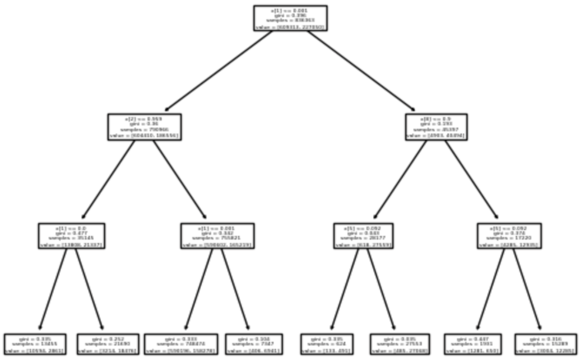


Figure 12: Decision Tree

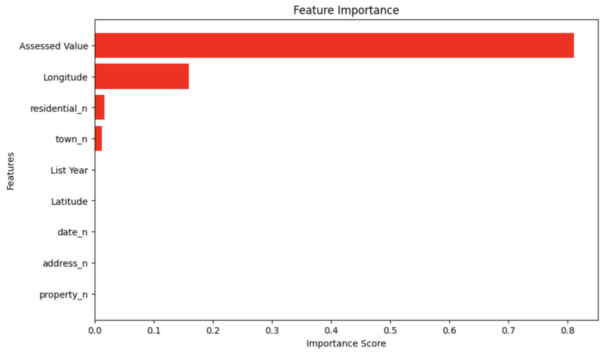


Figure 13: Feature Importance from Decision Tree

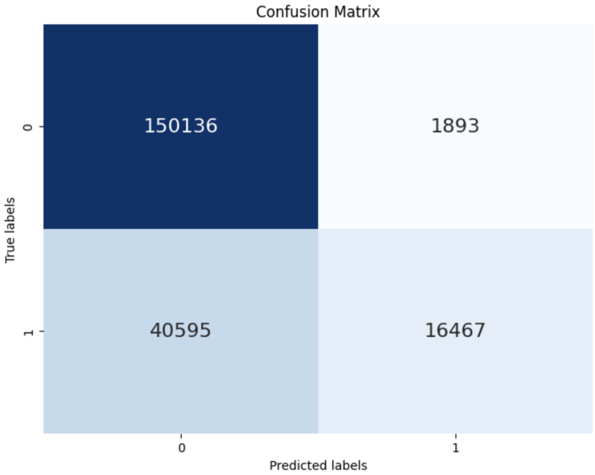


Figure 14: Confusion Matrix from Decision Tree

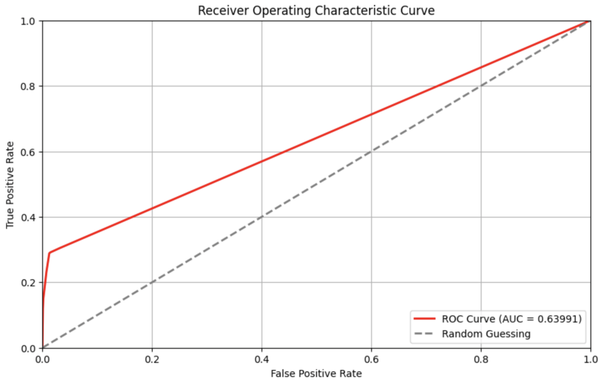


Figure 15: Receiver Operating Characteristic Curve from Decision Tree

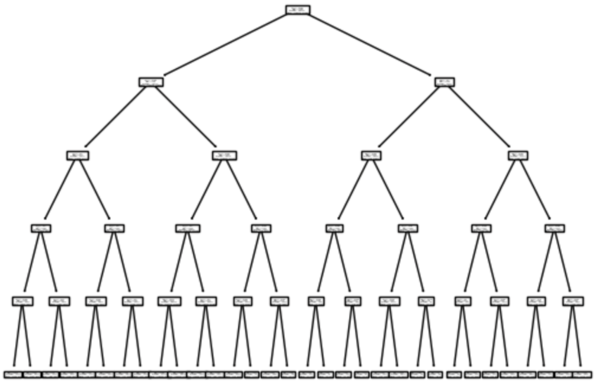


Figure 16: Best Model Decision Tree

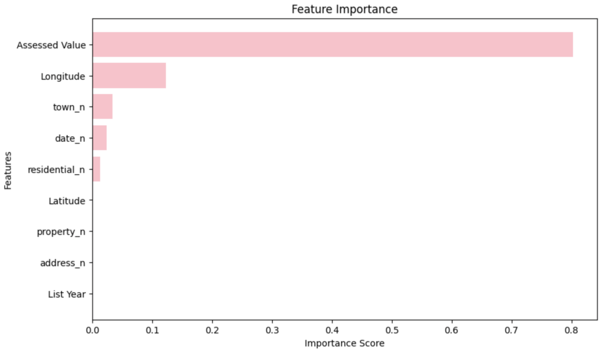


Figure 17: Best Model Decision Tree Feature Importance

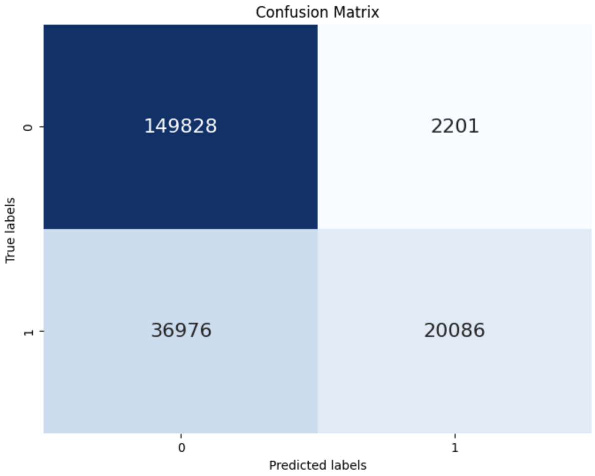


Figure 18: Best Model Decision Tree Confusion Matrix

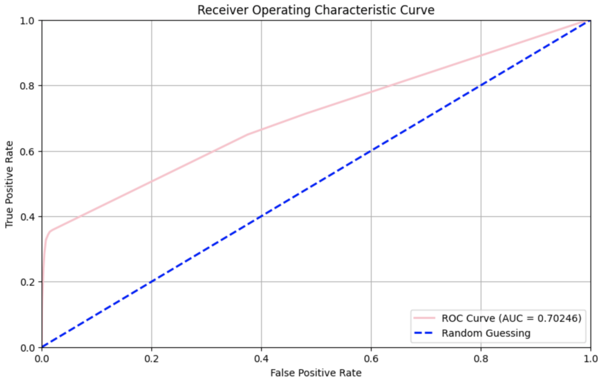
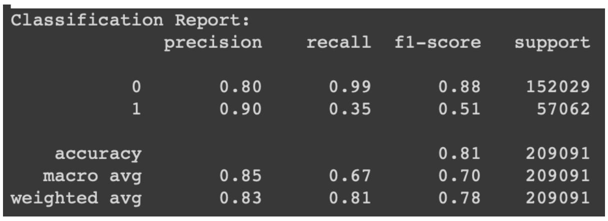


Figure 19: Receiver Operating Characteristic Curve for Best Model Decision Tree

6. **Interpret the evaluation results: Explain how you used the matrix to compare and rank different approaches. Discuss any trade-offs between different evaluation criteria. Justify your final model selection based on the evaluation matrix.**

   After looking at the classification reports of the confusion matrix from both best model decision tree and GaussianNB, it turned out that best model decision tree had better metrics than GaussianNB. Here are the metrics below:

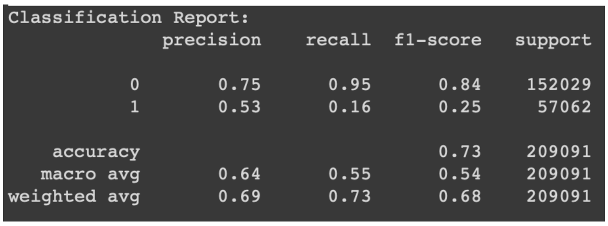


```
Classification Report:
              precision    recall  f1-score   support

           0       0.80      0.99      0.88    152029
           1       0.90      0.35      0.51     57062

    accuracy                           0.81    209091
   macro avg       0.85      0.67      0.70    209091
weighted avg       0.83      0.81      0.78    209091
```

Figure 20: Classification Report from Best Model Decision Tree



```
Classification Report:
              precision    recall  f1-score   support

           0       0.75      0.95      0.84    152029
           1       0.53      0.16      0.25     57062

    accuracy                           0.73    209091
   macro avg       0.64      0.55      0.54    209091
weighted avg       0.69      0.73      0.68    209091
```

Figure 21: Classification Report from GaussianNB.

   Model 1, the Best Decision Tree Model had an accuracy of 81% and Model 2, GaussianNB, had an accuracy of 73%. Overall Model 1 is the best overall option but there could be improvements made on further tuning the recall without compromising the precision too much.

7. **Address potential limitations of your evaluation approach: Discuss any biases or shortcomings in your evaluation methodology. Consider how**

well your evaluation generalizes to real-world scenarios. Suggest potential improvements to the evaluation process for future work.

Addressing potential limitations of our evaluation approach, we recognize that there are additional metrics, such as AUC-ROC scores, that should be considered for a more comprehensive assessment across different models, rather than limiting the evaluation to just two. In instances where metrics are close, exploring multiple classifications instead of binary classification could provide a better indicator of affordability. Our current methodology only checked metrics for highly affordable cases and did not consider other categories, such as not affordable, medium, or low affordability. Including these categories could yield more accurate predictions and better classification scores. For future work, expanding the evaluation to include these additional metrics and categories would likely improve the robustness of our results to real-world scenarios. We could have also given more time for SVM and KNN to train, but they were taking a long time to train. They could have turned out to be more accurate however we do not know that. We can also choose to compare data according to different time periods since prices change as a result of inflation.