Phase 4 Report

Megan Joseph

April 27, 2025

1 Summary

The goal of this phase is to study feature importance. The features of my chosen dataset are Age, Gender, AnnualIncome, NumberOfPurchases, ProductCategory, TimeSpentOnWebsite, and LoyaltyProgram. First, I created seven models trained on a single feature and plotting the validation accuracies of each.

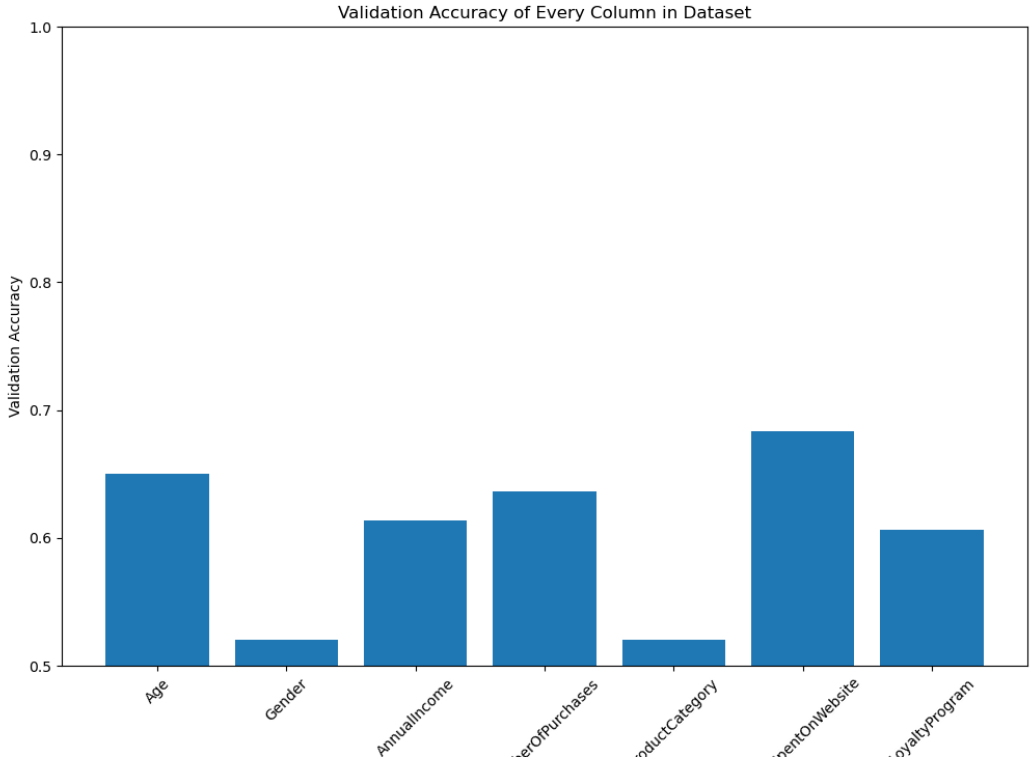

Figure 1: Validation accuracies for models trained on a single feature in the dataset

Next, I started removing the least important features (the features where the respective model had the lowest validation accuracy) from the full model and found the validation accuracy. I did this until the model only had the feature that gave the highest accuracy. In this case, I removed Gender, then ProductCategory, and so on until I was left with only TimeSpentOnWebsite. Then, I plotted the validation accuracies for each. The x-axis labels were too long, so I put them in a legend to make the graph easier to read.

After this, I compared the accuracies of the feature-reduced model with the best validation accuracy to the best model with full features. The feature-reduced model has an accuracy of 0.803 while the full-feature model has an accuracy of 0.817.

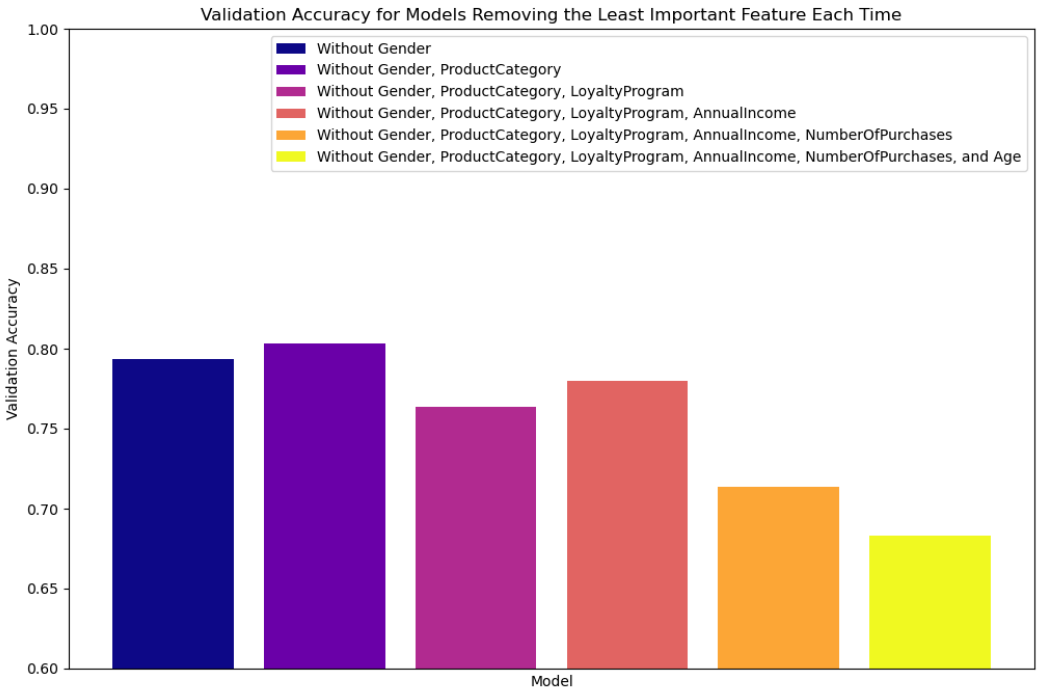

Figure 2: Validation accuracies for models trained on all features and removing the least important feature each time

To investigate feature importance further, I found the Shapley values. I chose to do this instead of LIME because I didn't have too many features, I didn't need a prediction model at the end, and just wanted to investigate the strengths of each feature. I just did a single iteration because I my Jupyter Notebook kept crashing if I did too many, but if I had the compute power, I would average over the entire dataset. The Shapley value I calculated is around 0.79. The column that is most important is AnnualIncome and the least imoprtant column is Age. Below is the code used to find the Shapley value:

```
def shapley_value(best_model, X_train):
    instance = X_train.iloc[[1], :]
    train_pred = best_model.predict(X_train).flatten()
    base_pred = np.mean(train_pred)


    cols = range(len(X_train.columns))
    subsets = [[]]
    for col in cols:
        subsets += [sub + [col] for sub in subsets]

    predictions = {}
    for sub in subsets:
        mod_instance = instance.copy()
        for i in cols:
            if i not in sub:
                mod_instance.iloc[0, i] = 0

        pred = best_model.predict(mod_instance).flatten()
        predictions[tuple(sub)] = pred

    avg_marg_contr = {}
```

```
    for col in cols:
        total_contr = 0
        num = 0
        for sub in subsets:
            if col not in sub:
                pred_wo_feature = predictions[tuple(sub)]
                sub_w_feature = sub + [col]
                pred_w_feature = predictions[tuple(sorted(sub_w_feature))]
                marg_contr = pred_w_feature - pred_wo_feature
                total_contr += marg_contr
                num += 1

        avg_contr = total_contr / num
        avg_marg_contr[col] = avg_contr

    final_pred = base_pred
    for col in cols:
        final_pred += avg_marg_contr[col]

    return final_pred, avg_marg_contr
```

Listing 1: Function to calculate the Shapley value for the model and determine the order of importance based on marginal contribution of each feature.