Creating Optimal Model

Joshua Lympany, Haotian Sun

Goal of Assignment

Find a model that outputs the most amount of profit

- +1.5$ for correct prediction 0

*Predict customer wouldn’t buy, and actually wouldn’t have***. Correctly send customer discount.**

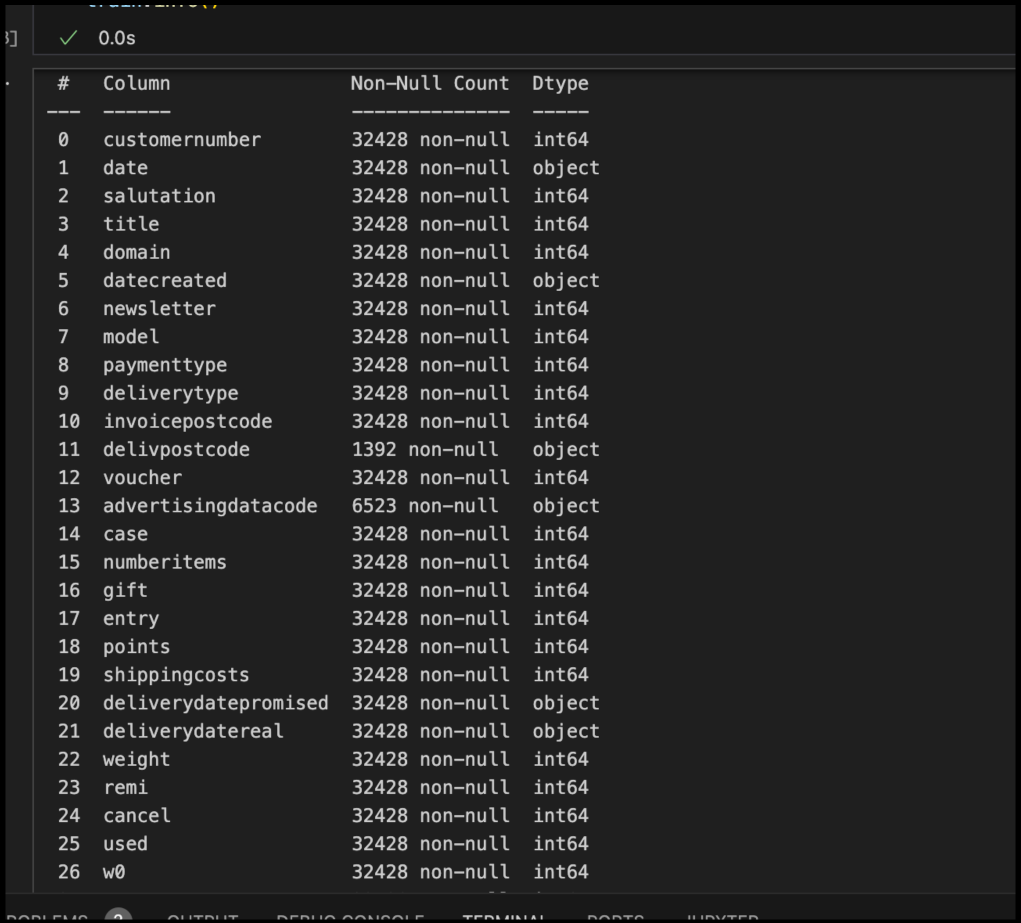
- -5$ for false prediction 1

Predict customer wouldn’t buy, but actually would have. ***Falsely send customer discount.***

Maximize Precision (0) and Recall (1)

Precision more important, as nearly 5x more instances of prediction 0.

Preprocessing



```
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   customernumber       32428 non-null  int64
 1   date                 32428 non-null  object
 2   salutation           32428 non-null  int64
 3   title                32428 non-null  int64
 4   domain               32428 non-null  int64
 5   datecreated          32428 non-null  object
 6   newsletter           32428 non-null  int64
 7   model                32428 non-null  int64
 8   paymenttype          32428 non-null  int64
 9   deliverytype         32428 non-null  int64
 10  invoicepostcode      32428 non-null  int64
 11  delivpostcode        1392 non-null   object
 12  voucher              32428 non-null  int64
 13  advertisingdatacode  6523 non-null   object
 14  case                 32428 non-null  int64
 15  numberitems          32428 non-null  int64
 16  gift                 32428 non-null  int64
 17  entry                32428 non-null  int64
 18  points               32428 non-null  int64
 19  shippingcosts        32428 non-null  int64
 20  deliverydatepromised 32428 non-null  object
 21  deliverydatereal     32428 non-null  object
 22  weight               32428 non-null  int64
 23  remi                 32428 non-null  int64
 24  cancel               32428 non-null  int64
 25  used                 32428 non-null  int64
 26  w0                   32428 non-null  int64
```

- Drop features for various reasons (too few values, unmeaningful, no predictiveness, data type, …)
- Created dummy variables
- Splitting training / test

Different Models

Classifiers for all

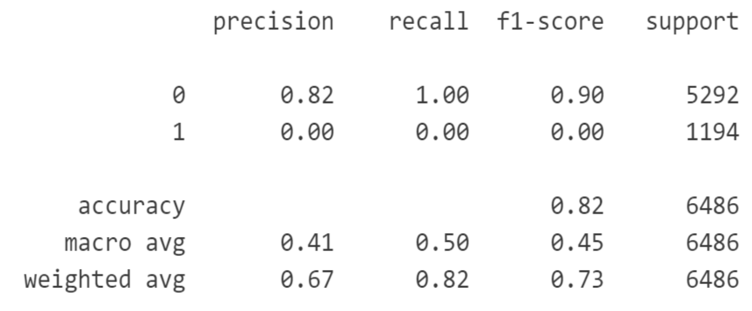
- Cross-Validation on many parameters for: xgBoost / RandomForest / KNN
- SVC

- Checked classification report on all variations
- **Focus:** Maximize Precision (0) and Recall (1)

The best model

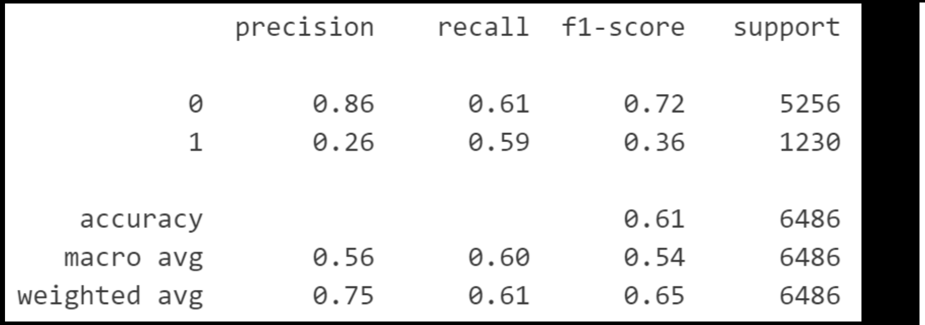
- RandomForestClassifier
- Class_weight=‘balanced’
- max_depth=9, min_samples_split=12, min_samples_leaf=17, max_features=5, n_estimators=37, random_state=71

Without Class_weight=‘balanced’



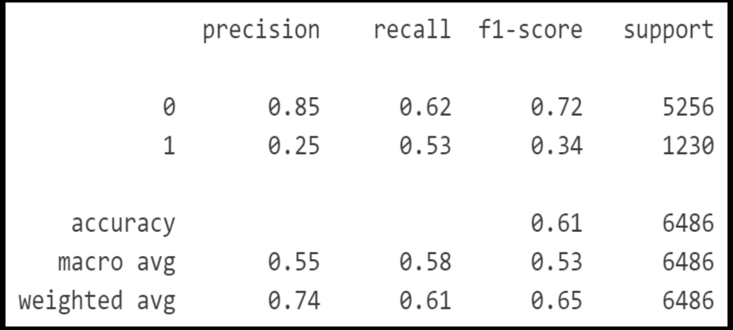
```
              precision    recall  f1-score   support

           0       0.82      1.00      0.90      5292
           1       0.00      0.00      0.00      1194

    accuracy                           0.82      6486
   macro avg       0.41      0.50      0.45      6486
weighted avg       0.67      0.82      0.73      6486
```

With Class_weight=‘balanced’



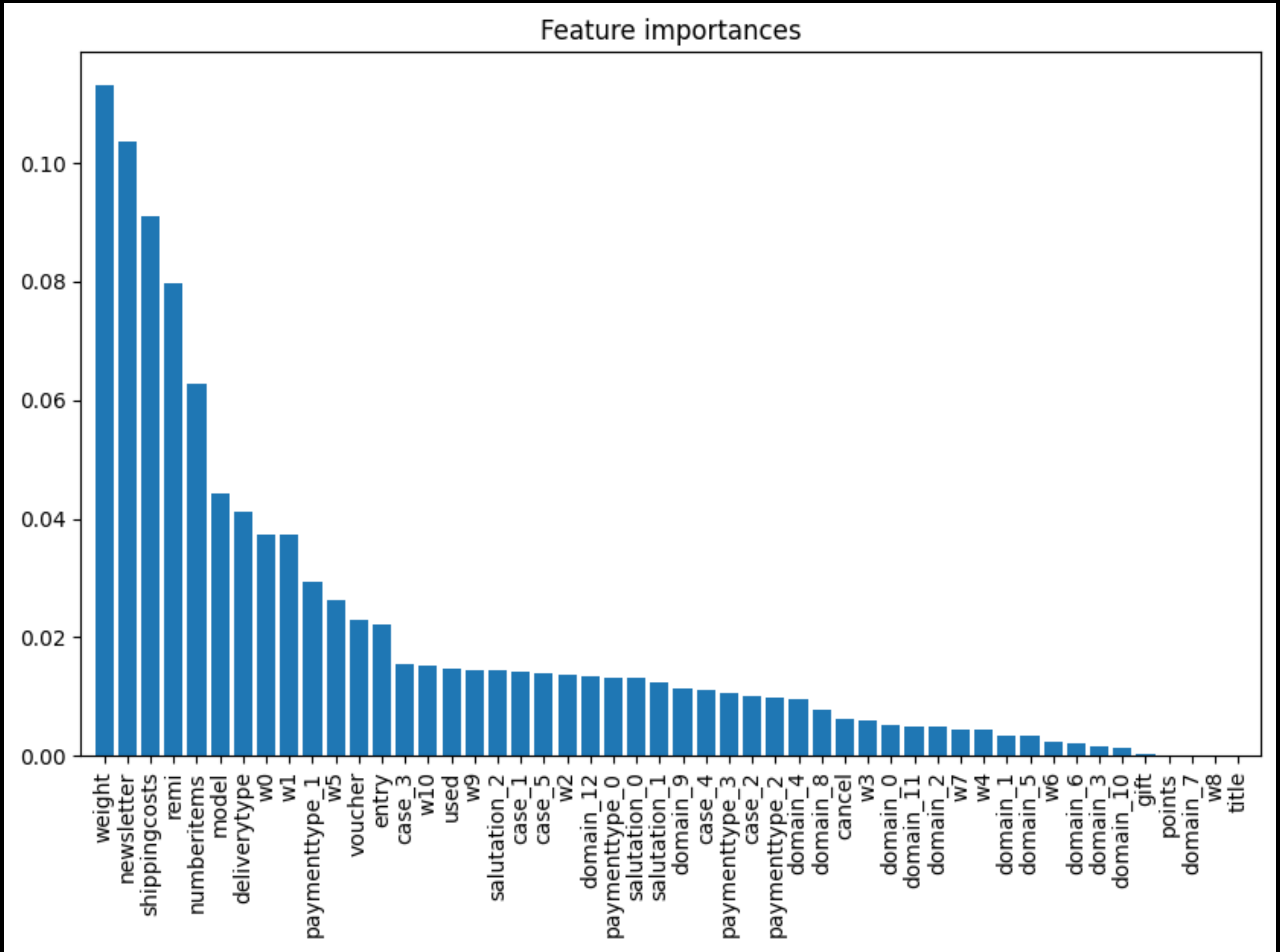
```
              precision    recall  f1-score   support

           0       0.86      0.61      0.72      5256
           1       0.26      0.59      0.36      1230

    accuracy                           0.61      6486
   macro avg       0.56      0.60      0.54      6486
weighted avg       0.75      0.61      0.65      6486
```

Model: xgboost

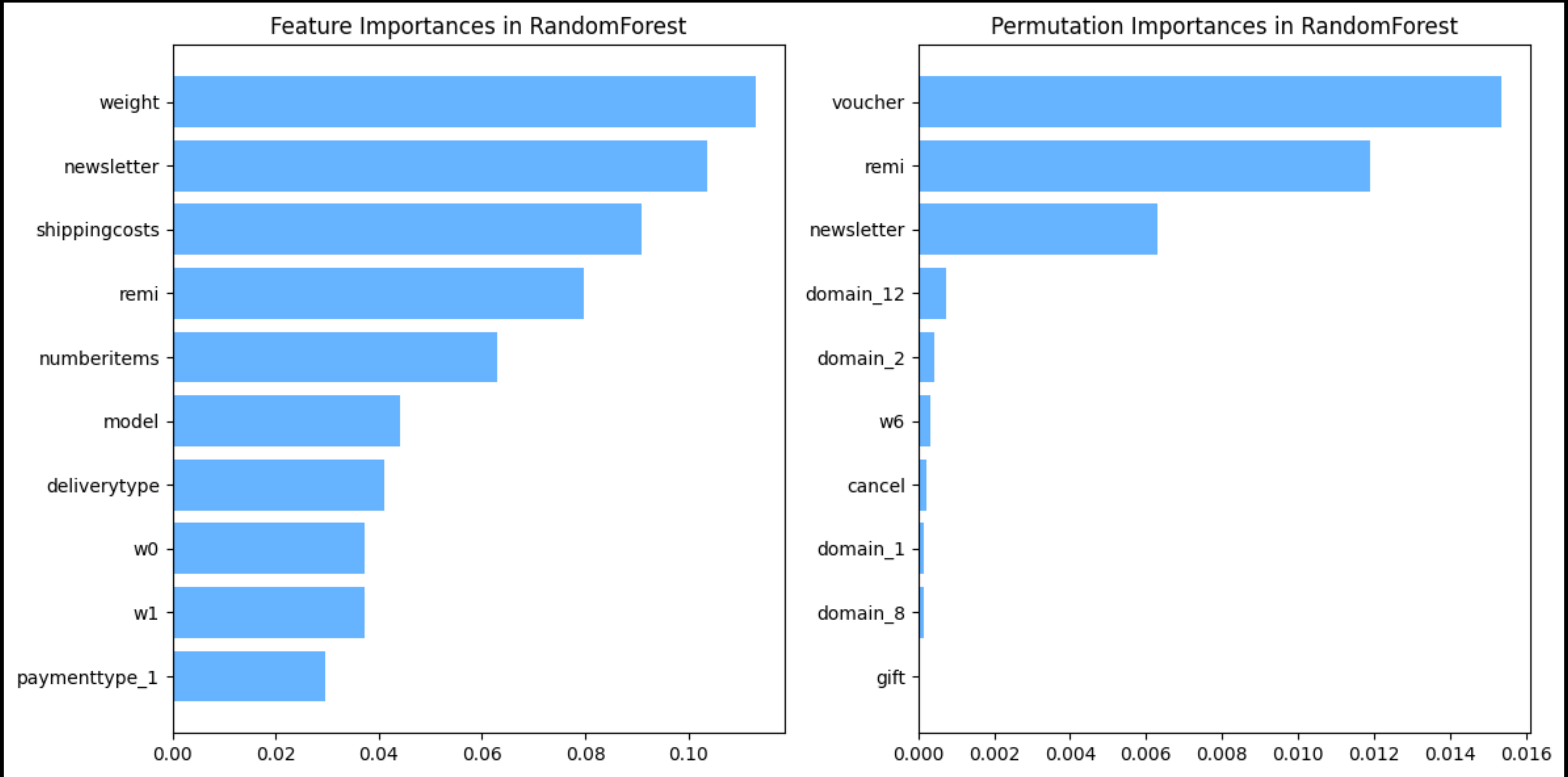


```
              precision    recall  f1-score   support

           0       0.85      0.62      0.72      5256
           1       0.25      0.53      0.34      1230

    accuracy                           0.61      6486
   macro avg       0.55      0.58      0.53      6486
weighted avg       0.74      0.61      0.65      6486
```

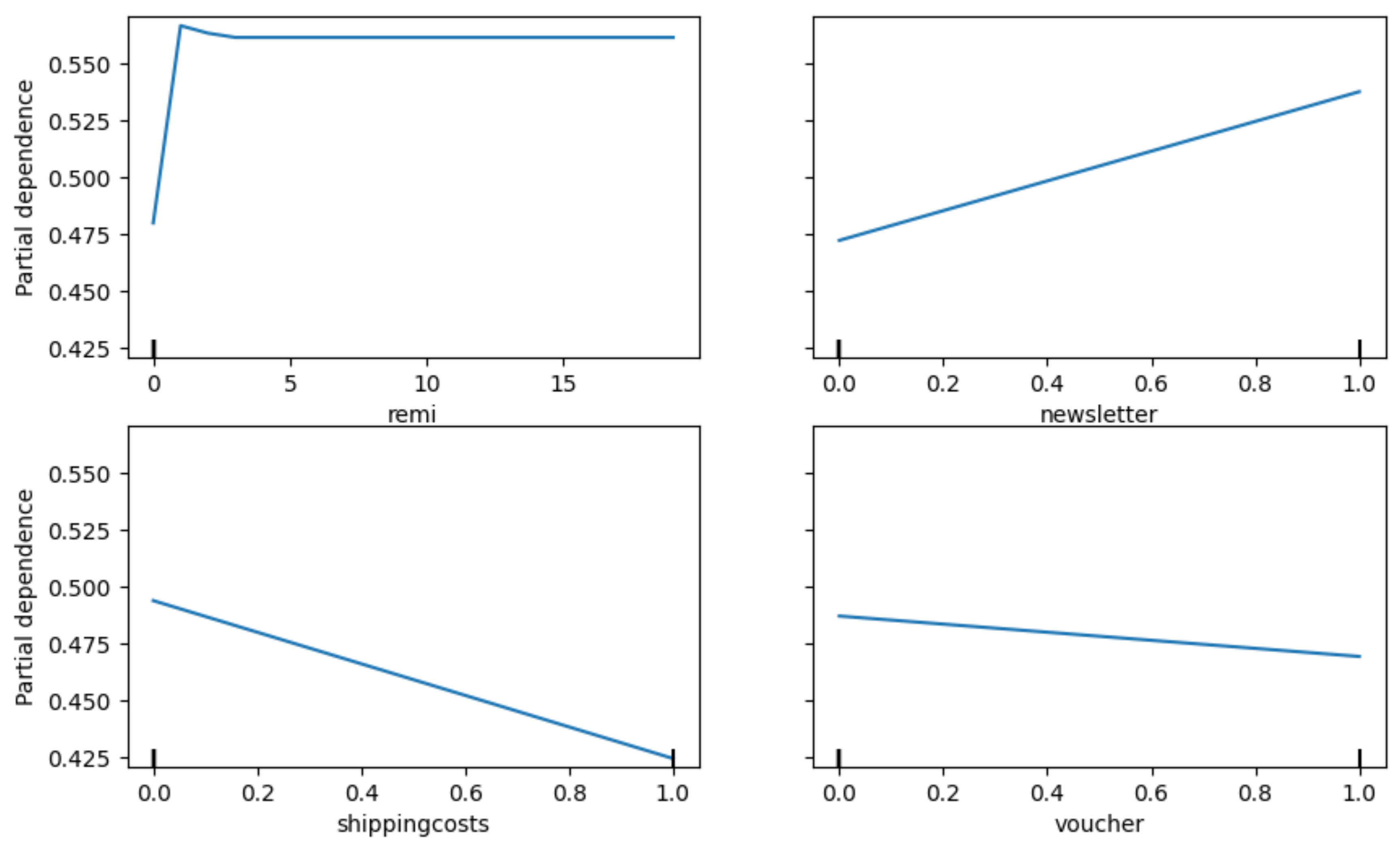
Global Interpretation: Feature importances



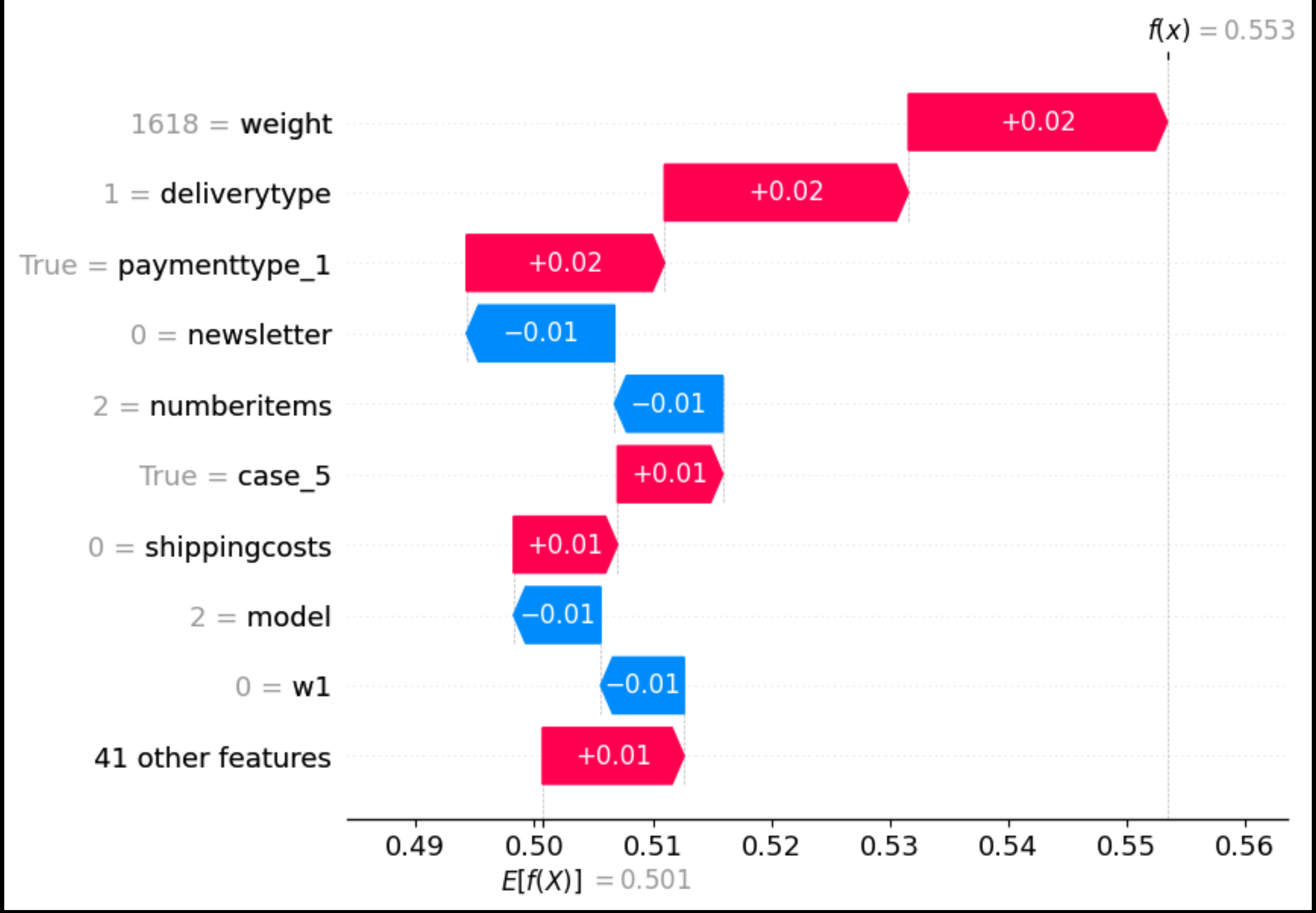
Global Interpretation: Permutation Importances



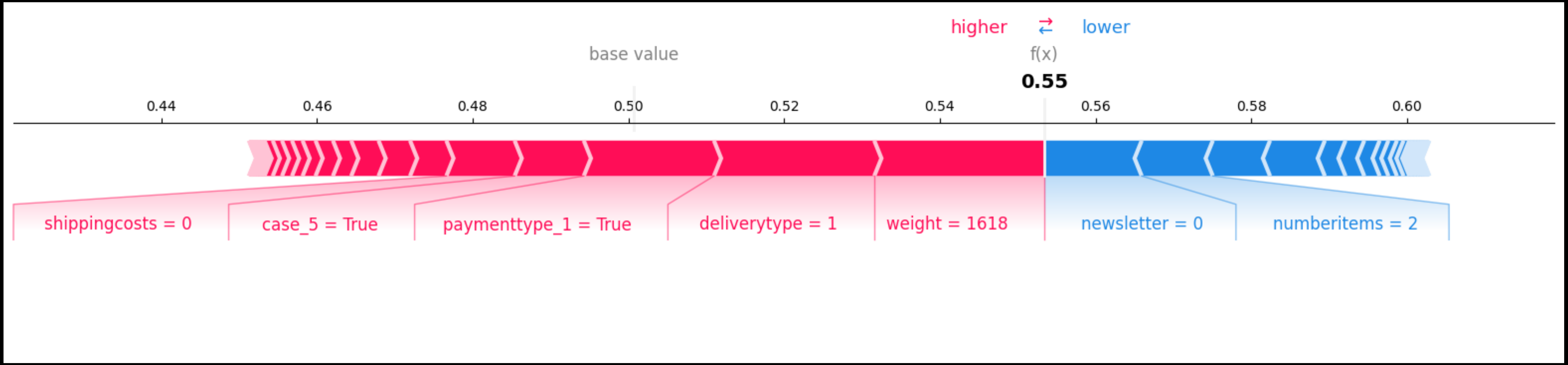
Global Interpretation: Partial Dependence Plots



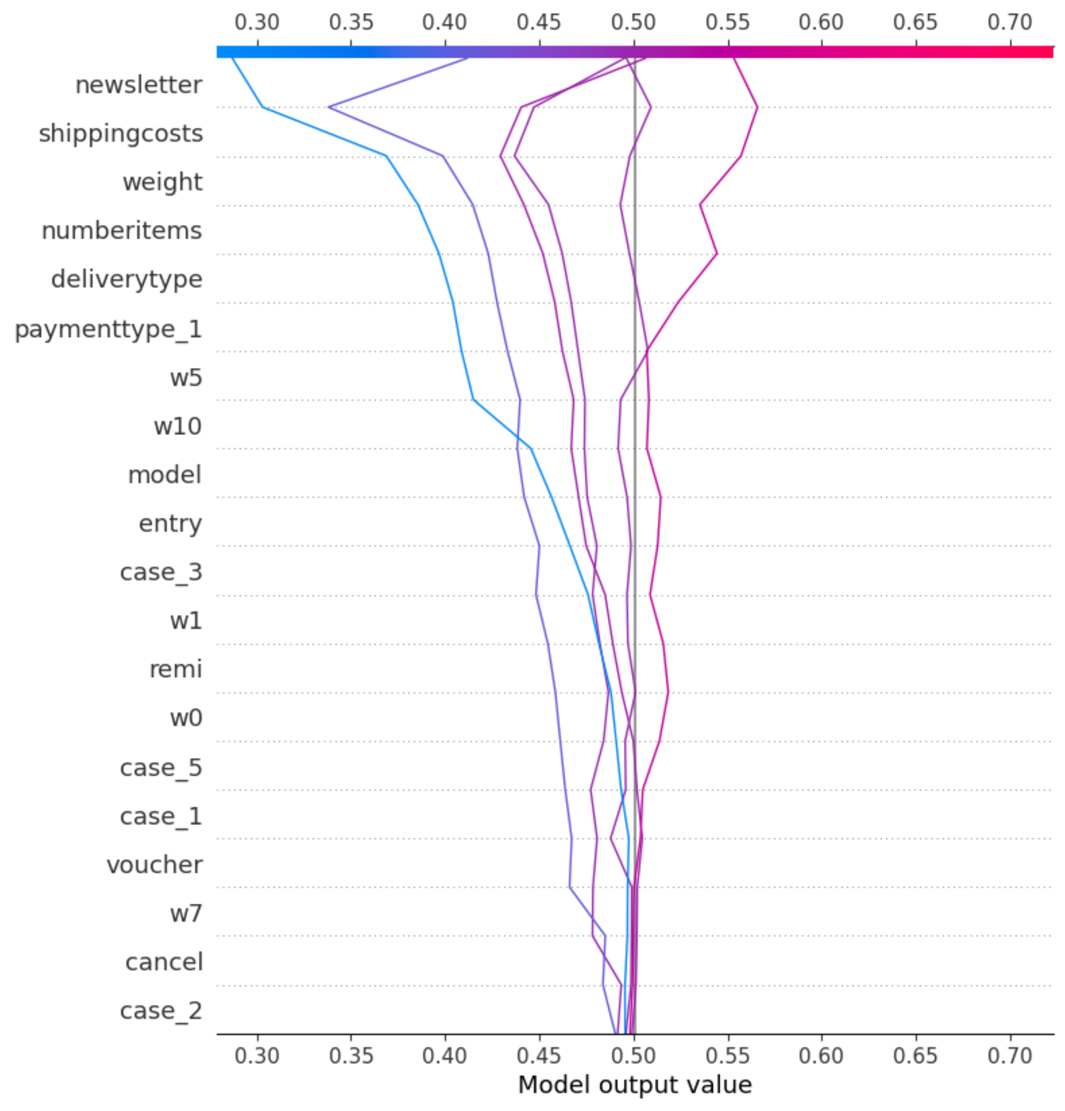
Local interpretation: Waterfall plot of SHAP values for the second row of data points



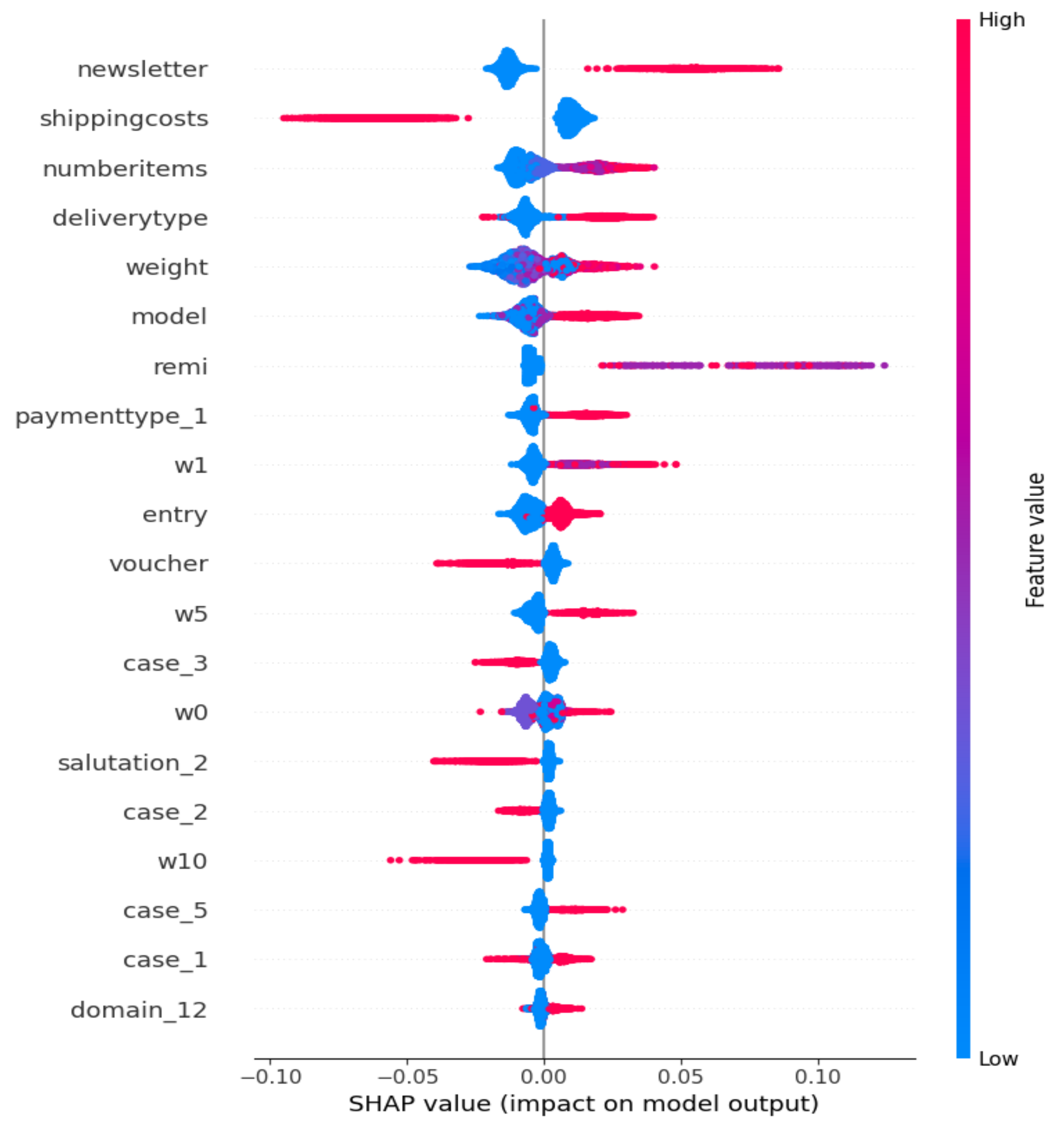
Local interpretation: SHAP power diagram for data point with index 18518



Local interpretation



Global and Local Interpretation



Conclusion and Profit

- Calculation:

(Actual_amount_of_0s*Precision_0*1.5) +
(Actual_amount_of_1s*(1-Recall_1)*(-5)

= **4289.01$** (y_test)