Data Analysis and Exploration
Logistic Regression Project

Lisa Godwin 2437980

Nihal Ranchod 2427378

May 22, 2024

School of Computer Science & Applied Mathematics University of the Witwatersrand

Contents

Question 1 - Data Exploration

Summary of Data

We have 4 variables in the dataset:

- **Stay:** Whether or not the data Scientist is there three months later. 1 if they stay, 0 otherwise.
- **Pay:** Monthly Salary in Dollars.
- **Estimated Happiness:** This comes from a complicated model reviewing both the employees reported happiness and there comments on what they'd like to see changed. This is on a scale of 1-10.
- **Performance:** Results of managers performance review. This is on a scale of 1-10.

The table below (Table 1) provides key statistics for each variable in our dataset. These statistics offer insights into the distribution and characteristics of the data, shedding light on various aspects such as salary levels, performance ratings, estimated happiness scores, and retention rates among data scientists.

Table 1: Summary of Data

Variable	**Mean**	**Std. Dev.**	**Min**	**Max**
Pay	23602.000	13519.806	10000.000	295000.000
Performance	6.052	1.447	2.000	10.000
Estimated Happiness	6.440	0.984	4.000	11.000
Stay	0.918	0.282	0.000	2.000

Here's a breakdown of the data:

- **Pay:** The average monthly salary for data scientists in the dataset is \$23,602, with a standard deviation of \$13,519. The salaries range from \$10,000 to \$295,000. This indicates a wide range of salaries among data scientists, with some earning significantly more than others. The large standard deviation suggests considerable variability in salaries, highlighting disparities in compensation within the dataset.
- **Performance:** The average performance rating is 6.052, with a standard deviation of 1.446. Performance ratings range from 1 to 10. This suggests that, on average, data scientists are performing moderately well according to performance reviews, with some variability in performance levels among individuals.
- **Estimated Happiness:** The average estimated happiness score is 6.44, with a standard deviation of 0.984. Estimated happiness scores range from 1 to 10. This indicates that, on average, data scientists report a moderate level of happiness, although there is some variability in happiness levels among individuals. However, the anomaly of an estimated happiness score of 11 is concerning as it falls outside the expected range of 1 to 10, suggesting a potential data entry error or anomaly.
- **Stay:** The majority of data scientists (about 92%) stay in their positions three months later, as indicated by the mean of 0.918. The minimum value is 0 (indicating they didn't stay) and the maximum value is 2, which shouldn't be possible since 'Stay' is a binary

variable, typically represented as either 0 or 1. This discrepancy suggests a potential data anomaly.

In maintaining the integrity of our analysis, we have chosen to exclude entries containing anomalies. Specifically, entries associated with the following IDs have been omitted from the dataset: ID 148 (Pay: $24,000, Performance: 4, Estimated Happiness: 7, Stay: 2) and ID 151 (Pay: $24,000, Performance: 6, Estimated Happiness: 11, Stay: 1). This decision ensures that our analysis is based on reliable and accurate data, thereby enhancing the validity of our findings.

Correlation Matrix

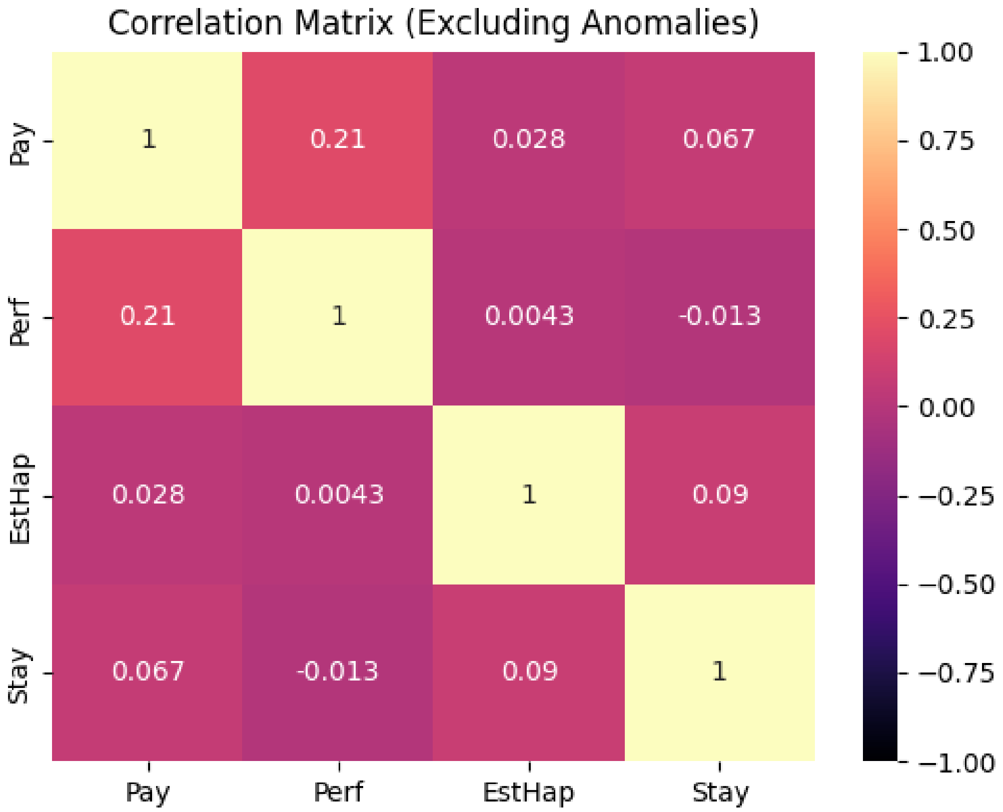


Figure 1: Correlation Matrix

- **Pay-Performance:** This is the correlation between 'Pay' and 'Perf'. It's 0.21, which indicates a positive correlation. This means that as 'Pay' increases, 'Perf' tends to increase as well, though the correlation isn't very strong.
- **Pay-Estimated Happiness:** This is the correlation between 'Pay' and 'EstHap'. It's 0.028, which is close to zero. This suggests a very weak positive correlation, indicating that there isn't much of a relationship between these two variables.
- **Pay-Stay:** This is the correlation between 'Pay' and 'Stay'. It's 0.067, which is also a weak positive correlation. It suggests that there's some tendency for higher pay to be associated with staying at the company, but again, the correlation is not very strong.

- **Performance-Estimated Happiness:** This is the correlation between ‘Perf’ and ’EstHap‘. It’s 0.0043, which is very close to zero. This indicates essentially no relationship between these two variables.

- **Performance-Stay:** This is the correlation between Perf’ and ‘Stay’. It’s -0.013, which is close to zero but slightly negative. This suggests a very weak negative correlation, implying that there’s a slight tendency for higher performance to be associated with staying at the company, but again, the correlation is very weak.

- **Estimated Happiness-Stay:** This is the correlation between ‘EstHap’ and ‘Stay’. It’s 0.09, which is a weak positive correlation. This suggests that there’s some tendency for estimated happiness to be associated with staying at the company, though the correlation is not very strong.

Plots

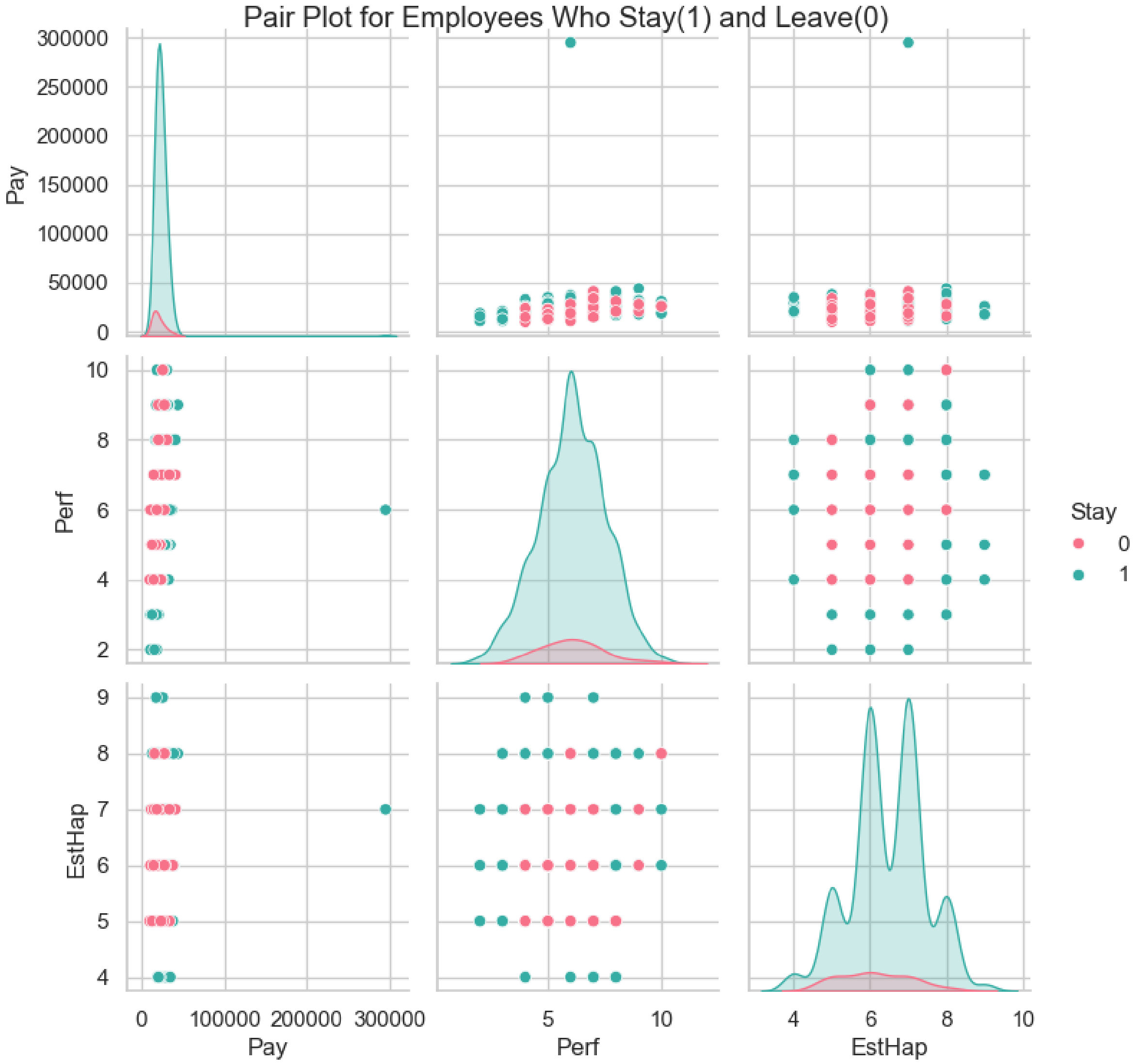


Figure 2: Plots showing the relationships between the different features

In Figure 2, there is an obvious outlier in the plots associated with pay. This outlier could indicate that this person is either at the top of their field and in demand, hence such a high pay, or it could be an error in the data. Considering that the average pay seen in Table 1 is $23,602.00, it is possible that an extra zero was mistakenly inputted when entering the data, resulting in a pay of $295,000 instead of $29,500. This could be a data anomaly so we are going to now ignore this data entry.

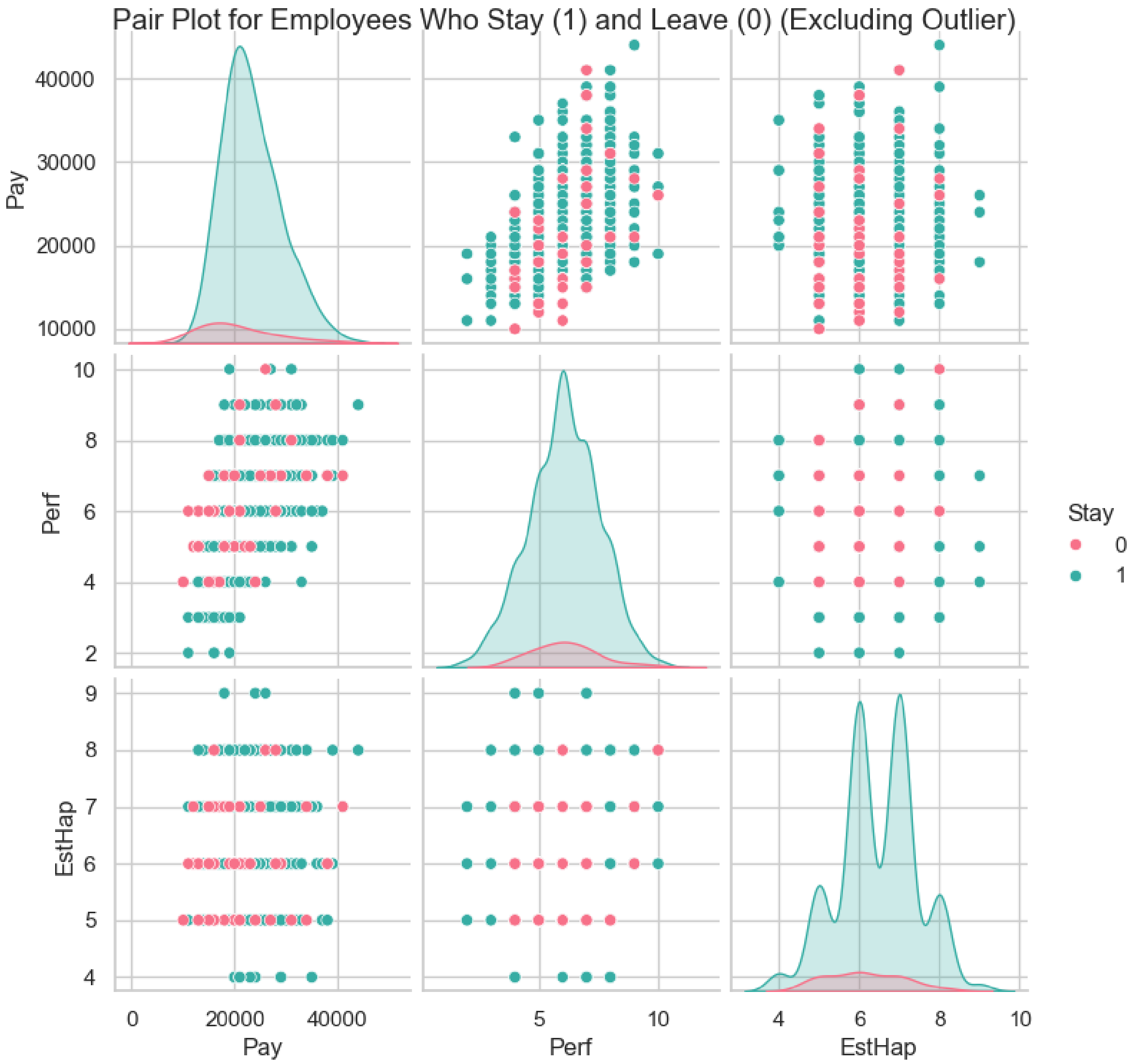


Figure 3: Plots showing the relationships between the different features excluding the outlier

The second pair plot (Figure 3) excludes this outlier and provides a clearer view of the data:

Pay vs. Pay

- Excluding the outlier, we see that most salaries are clustered between $10,000 and $40,000.
- The distribution is slightly skewed to the right, indicating a few higher salaries in the dataset.

Performance vs. Pay

- The scatter plot shows the relationship between ‘Perf’ and ‘Pay’.
- Employees with ‘Pay’ up to around $40,000 have a wide range of performance scores (from 2 to 10).
- Higher performance ratings slightly correlate with higher pay.
- Employees who stay (green) tend to have higher performance ratings compared to those who leave (pink), but this trend is not strong.

Estimated Happiness vs. Pay

- The scatter plot shows the relationship between ‘EstHap’ and ‘Pay’.
- Estimated happiness scores are spread across different pay levels without a clear trend.
- Density plots indicate two peaks in happiness scores around 6 and 8.
- There is no strong correlation between pay and estimated happiness.
- Employees who stay (green) tend to have slightly higher happiness scores, particularly around the peaks of 6 and 8.

Performance vs. Performance

- Most performance scores are centred around 6, with fewer employees scoring at the extremes.
- The plot shows that employees who stay (green) generally have higher performance scores compared to those who leave (pink).

Estimated Happiness vs. Performance

- The scatter plot shows the relationship between ‘EstHap’ and ‘Perf’.
- Happiness scores of employees are spread across different performance levels, without a clear trend.
- Employees who stay (green) tend to have slightly higher performance and happiness scores, but this relationship is weak.

Estimated Happiness vs. Estimated Happiness

- Happiness scores have peaks at 6 and 8.
- Employees who stay (green) are slightly more concentrated around the higher happiness scores (6 and 8), indicating that higher happiness is somewhat associated with staying, though the trend is not very strong.

Overall Analysis

Pay:

- Distribution shows most employees earn between \$10,000 and \$40,000.
- No strong correlation with performance or happiness.
- Slight tendency for higher pay to be associated with staying.

Performance:

- Scores are centred around 6, with a slight positive correlation with pay.
- Higher performance is weakly associated with staying.

Estimated Happiness:

- Scores peak at 6 and 8, with no strong correlation to pay or performance.
- Higher happiness scores are weakly associated with staying.

Staying vs. Leaving:

- Employees who stay tend to have slightly higher pay, performance, and happiness scores.

Question 2 - Variable Selection and Engineering

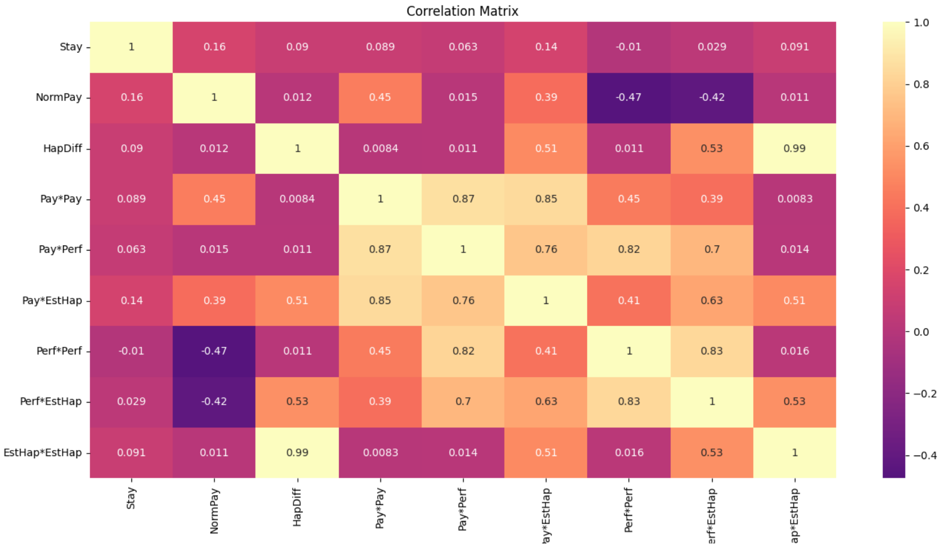


Figure 4: Correlation Matrix with Engineered Features

Basic Model with Original Variables

Description:

- We first look at the original variables Pay, Estimated Happiness, and Performance to predict Stay.

Findings:

- Correlation analysis (Figure 1) showed weak correlations between Pay, Performance, Estimated Happiness and Stay.
- None of these variables by themselves will be sufficient to predict whether an employee stays or not accurately.

Interaction Terms

Description:

- To improve the model, we introduced interaction terms between the variables. Interaction terms can capture the combined effect of two variables on the target variable.
- The interaction terms included were:
  - **Pay * Performance**
  - **Pay * Estimated Happiness**
  - **Performance * Estimated Happiness**
  - **Pay * Pay**
  - **Performance * Performance**

  - **Estimated Happiness * Estimated Happiness**

Findings:

- Including interaction terms provided a slight improvement in model performance (Figure 4), indicating that the interaction between variables has some impact on predicting whether an employee stays or leaves.
- Correlation matrix (Figure 4) showed that these interaction terms had low to moderate correlations with Stay, such as:
  - 0.144 for **Pay * Estimated Happiness**
  - 0.089 for **Pay * Pay**
  - 0.091 for **Estimated Happiness * Estimated Happiness**
- The low to moderate correlations suggest that while some interaction terms have a relationship with Stay, the improvement in prediction might be limited. The improvement was not substantial, suggesting that more complex relationships or additional features might be necessary.

Feature Engineering

Description:

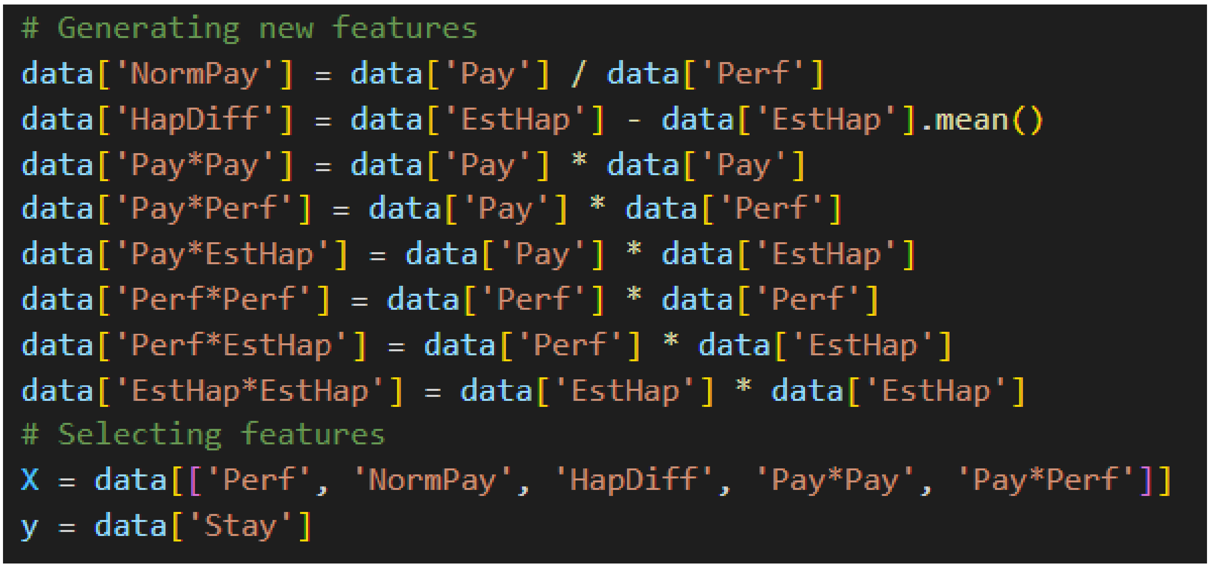
- To further enhance the model, additional features were engineered:
  - **Normalised Pay:** Pay normalised by performance level (`NormPay = Pay/Performance`).
  - **Happiness Difference:** Difference between estimated happiness and average happiness in the dataset (`HapDiff = Estimated Happiness - Mean(Estimated Happiness)`).
- These features aimed to capture more nuanced insights:
  - **Normalised Pay** accounts for the fact that higher performance might justify higher pay, making it a relative measure.
  - **Happiness Difference** identifies employees who are significantly happier or unhappier than the average, which might influence their decision to stay or leave.

Findings:

- Correlation matrix revealed that Normalised Pay had a correlation of 0.156 with Stay, and Happiness Difference had a correlation of 0.090, indicating their potential usefulness in prediction. Although these correlations are not super high, they are better than original variables in relation to Stay.

Question 3 - Fit the Model

Variable Selection



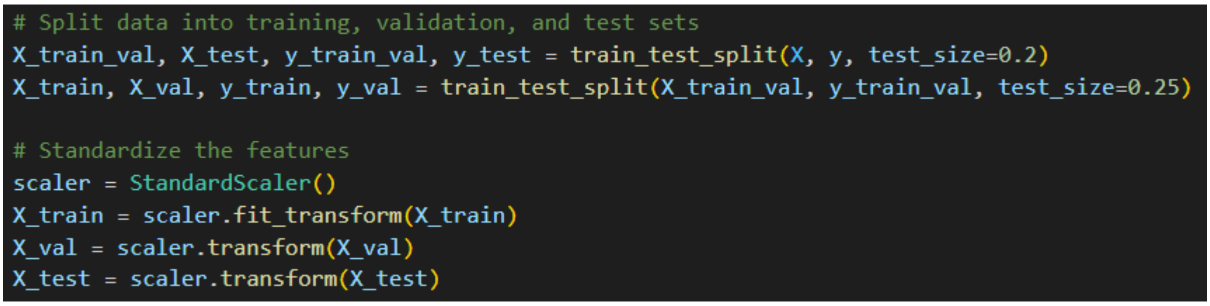
```
# Generating new features
data['NormPay'] = data['Pay'] / data['Perf']
data['HapDiff'] = data['EstHap'] - data['EstHap'].mean()
data['Pay*Pay'] = data['Pay'] * data['Pay']
data['Pay*Perf'] = data['Pay'] * data['Perf']
data['Pay*EstHap'] = data['Pay'] * data['EstHap']
data['Perf*Perf'] = data['Perf'] * data['Perf']
data['Perf*EstHap'] = data['Perf'] * data['EstHap']
data['EstHap*EstHap'] = data['EstHap'] * data['EstHap']
# Selecting features
X = data[['Perf', 'NormPay', 'HapDiff', 'Pay*Pay', 'Pay*Perf']]
y = data['Stay']
```

Figure 5: Variable Selection Code Snippet

In Figure 5 we perform feature engineering to create new features that might help improve the model's performance. We calculate normalised pay (NormPay) and the difference from the mean estimated happiness (HapDiff). Additionally, we generate interaction terms such as Pay*Pay, Pay*Perf, Pay*EstHap, Perf*Perf, Perf*EstHap, and EstHap*EstHap.

To determine the most effective features for our model, we executed a Python script that systematically evaluated various feature combinations. The script employed a logistic regression classifier and cross-validated the model's performance, outputting the highest achieved accuracy as the evaluation metric. The results of this experiment can be found in the following text file. Ultimately the best features were: Perf, NormPay, HapDiff, Pay*pay and Pay*Perf

Data Split and Standardisation



```
# Split data into training, validation, and test sets
X_train_val, X_test, y_train_val, y_test = train_test_split(X, y, test_size=0.2)
X_train, X_val, y_train, y_val = train_test_split(X_train_val, y_train_val, test_size=0.25)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_val = scaler.transform(X_val)
X_test = scaler.transform(X_test)
```

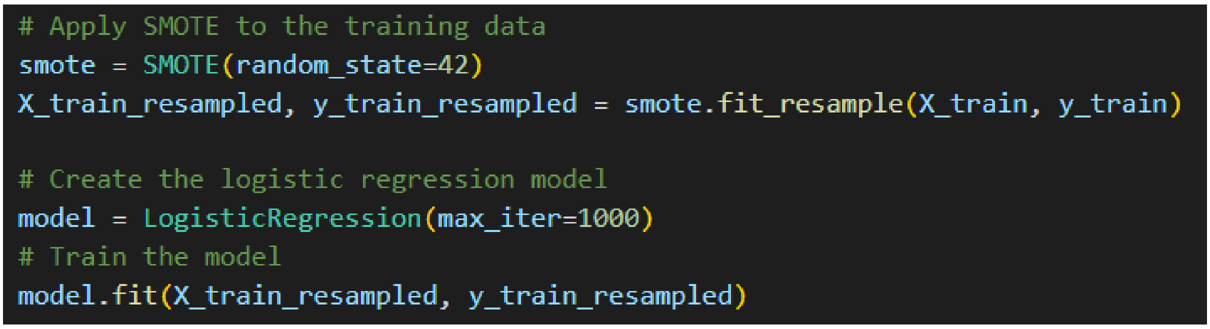
Figure 6: Data Split and Standardisation Code Snippet

In Figure 6, we split the data into training, validation, and test sets. The goal of splitting the data is to ensure that our model is trained, validated, and tested on different subsets of the data, preventing over-fitting and providing an unbiased evaluation of the model's performance.

1. **Data Split:**
   - Training: 60
   - Validation: 20
   - Test: 20
2. **Standardisation:**
   Standardisation ensures that all features contribute equally to the model by scaling them to have a mean of 0 and a standard deviation of 1. We use `StandardScaler` to fit and transform the training data (`X_train`) and then use the same scaler to transform the validation (`X_val`) and test sets (`X_test`). This prevents information leakage from the validation and test sets into the training process.

By carefully splitting the data and standardising it, we ensure that our model is trained effectively and evaluated accurately on unseen data.

Handling Imbalanced Data and Model Training



```
# Apply SMOTE to the training data
smote = SMOTE(random_state=42)
X_train_resampled, y_train_resampled = smote.fit_resample(X_train, y_train)

# Create the logistic regression model
model = LogisticRegression(max_iter=1000)
# Train the model
model.fit(X_train_resampled, y_train_resampled)
```

Figure 7: Handling Imbalanced Data and Model Training Code Snippet

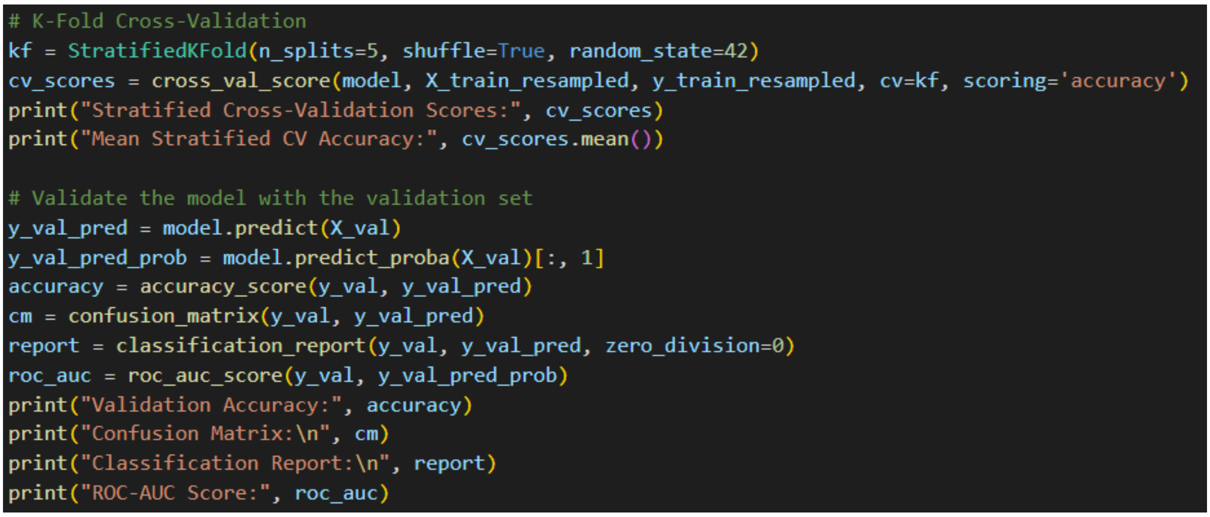
In Figure 7, we address the issue of class imbalance by applying the SMOTE technique to the training data. Here's a detailed explanation:

1. **Class Imbalance:**
   Class imbalance occurs when one class in the target variable is significantly underrepresented compared to others. In binary classification, this means that the number of instances in one class (the minority class, they don't stay) is much smaller than in the other class (the majority class, they do stay). Due to this imbalance, the model might become biased towards predicting the majority class, leading to poor performance in identifying instances of the minority class. This can result in high accuracy but low recall for the minority class.
2. **SMOTE Technique:**
   SMOTE (Synthetic Minority Over-sampling Technique) is a popular method to address class imbalance by generating synthetic samples for the minority class. SMOTE creates synthetic instances by interpolating between existing minority class instances. It selects k nearest neighbours for each minority class instance and generates new instances along the line segments joining these neighbours in the feature space.

3. **Training the Model:**
   We create a logistic regression model using the `LogisticRegression` class with a maximum number of iterations set to 1000 to ensure convergence. The model is then trained using the resampled training data.

By incorporating SMOTE into the training data preprocessing pipeline, we ensure that the model is trained on a balanced dataset. This approach helps to mitigate the risk of the model being biased towards the majority class and improves its ability to accurately classify instances from both classes, particularly the minority class. This step is crucial for improving the overall performance and generalisation of the model, especially in scenarios where class imbalance is significant.

Validation



```
# K-Fold Cross-Validation
kf = StratifiedKFold(n_splits=5, shuffle=True, random_state=42)
cv_scores = cross_val_score(model, X_train_resampled, y_train_resampled, cv=kf, scoring='accuracy')
print("Stratified Cross-Validation Scores:", cv_scores)
print("Mean Stratified CV Accuracy:", cv_scores.mean())

# Validate the model with the validation set
y_val_pred = model.predict(X_val)
y_val_pred_prob = model.predict_proba(X_val)[:, 1]
accuracy = accuracy_score(y_val, y_val_pred)
cm = confusion_matrix(y_val, y_val_pred)
report = classification_report(y_val, y_val_pred, zero_division=0)
roc_auc = roc_auc_score(y_val, y_val_pred_prob)
print("Validation Accuracy:", accuracy)
print("Confusion Matrix:\n", cm)
print("Classification Report:\n", report)
print("ROC-AUC Score:", roc_auc)
```

Figure 8: Model Validation Code Snippet

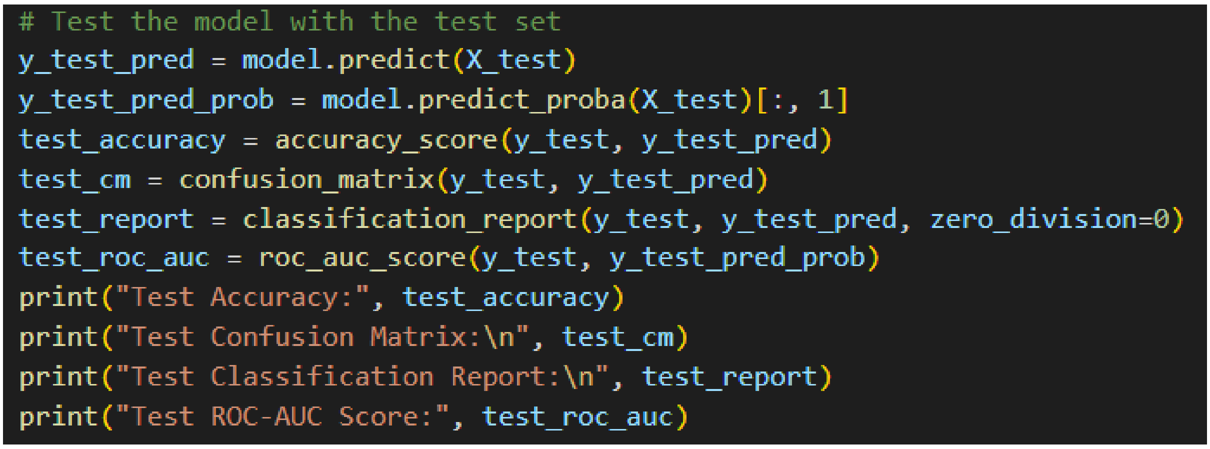
In Figure 8, we employ 5-fold cross-validation to assess the model's performance on the resampled training data. Here's a breakdown:

1. **K-Fold Cross-Validation:**
   We use the `StratifiedKFold` class to split the resampled training data into 5 folds while maintaining class balance. The data is shuffled before splitting. Each fold serves as both a training and validation set once, resulting in 5 iterations. The training data is fitted to the model, and the validation accuracy is computed for each iteration. We calculate the accuracy of the model for each fold using the specified scoring metric and returns an array of scores.

2. **Validation:**
   After cross-validation, we further validate the model's performance using the separate validation set. We predict the target variable using the validation features and compute the predicted probabilities. Various performance metrics such as accuracy, confusion matrix, classification report, and ROC-AUC score are computed and printed to evaluate the model's performance on the validation set.

5-fold cross-validation provides a robust estimate of the model's generalisation performance by averaging the performance across multiple folds. It helps to assess the model's stability and reduces the risk of overfitting or underfitting to a particular training-validation split. This approach ensures that the model's performance evaluation is less sensitive to the choice of a single validation set.

Testing



```python
# Test the model with the test set
y_test_pred = model.predict(X_test)
y_test_pred_prob = model.predict_proba(X_test)[:, 1]
test_accuracy = accuracy_score(y_test, y_test_pred)
test_cm = confusion_matrix(y_test, y_test_pred)
test_report = classification_report(y_test, y_test_pred, zero_division=0)
test_roc_auc = roc_auc_score(y_test, y_test_pred_prob)
print("Test Accuracy:", test_accuracy)
print("Test Confusion Matrix:\n", test_cm)
print("Test Classification Report:\n", test_report)
print("Test ROC-AUC Score:", test_roc_auc)
```

Figure 9: Testing the Model Code Snippet

We test the trained model on the test set and calculate the accuracy, confusion matrix, classification report, and ROC-AUC score to evaluate its performance on unseen data.

Test Accuracy: 0.72

Question 4

Model Variables and Their Importance

The logistic regression model was fitted using several engineered features:

- **NormPay:** Normalized pay by performance.
- **Pay*Perf:** Interaction term between pay and performance.
- **HapDiff:** Difference between estimated happiness and its mean.
- **Pay*Pay:** Squared term for pay.
- **Perf:** Performance.

These features were selected based on their potential to capture the relationship between pay, performance, happiness, and the likelihood of staying in the job. These engineered features aim to provide a more nuanced understanding of how these factors interact and affect employee retention.

Validation Results

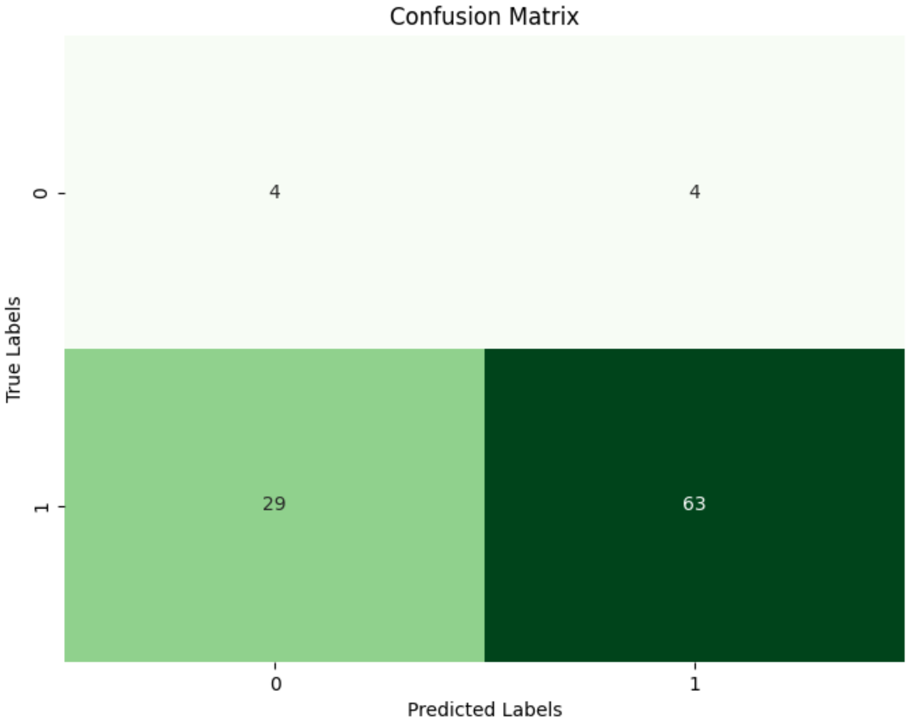


Figure 10: Confusion Matrix on Validation Dataset

Stratified Cross-Validation Scores:

```
[0.69724771 0.70642202 0.66055046 0.71559633 0.60185185]
```

Mean Stratified CV Accuracy: 0.6763
Validation Accuracy: 0.67
Classification Report:

```
              precision    recall  f1-score   support

           0       0.12      0.50      0.20         8
           1       0.94      0.68      0.79        92

    accuracy                           0.67       100
   macro avg       0.53      0.59      0.49       100
weighted avg       0.87      0.67      0.74       100
```

ROC-AUC Score: 0.596
Performance Discussion:

- **Accuracy:** The model achieved an accuracy of 0.67 on the validation set, indicating that it correctly predicted the target variable in 67% of the cases.
- **Precision and Recall for Class 1:** The model has high precision (0.94) and recall (0.68) for class 1 (those who stay), suggesting that it is effective at identifying individuals who are likely to stay.
- **Precision and Recall for Class 0:** The precision (0.12) and recall (0.50) for class 0 (those who leave) are much lower, indicating difficulty in predicting this class accurately.
- **ROC-AUC Score:** The ROC-AUC score of 0.596 indicates that the model's ability to distinguish between the two classes is better than random guessing but still moderate.

Test Results

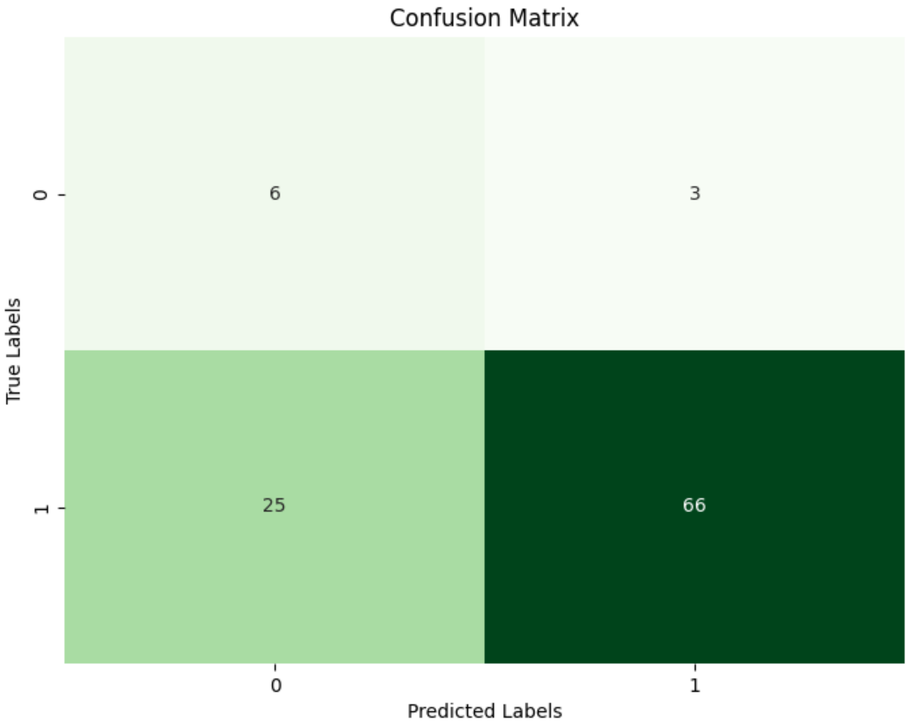


Figure 11: Confusion Matrix for Test Dataset

Test Accuracy: 0.72
Classification Report:

```
              precision    recall  f1-score   support

           0       0.19      0.67      0.30         9
           1       0.96      0.73      0.82        91

    accuracy                           0.72       100
   macro avg       0.58      0.70      0.56       100
weighted avg       0.89      0.72      0.78       100
```

ROC-AUC Score: 0.755

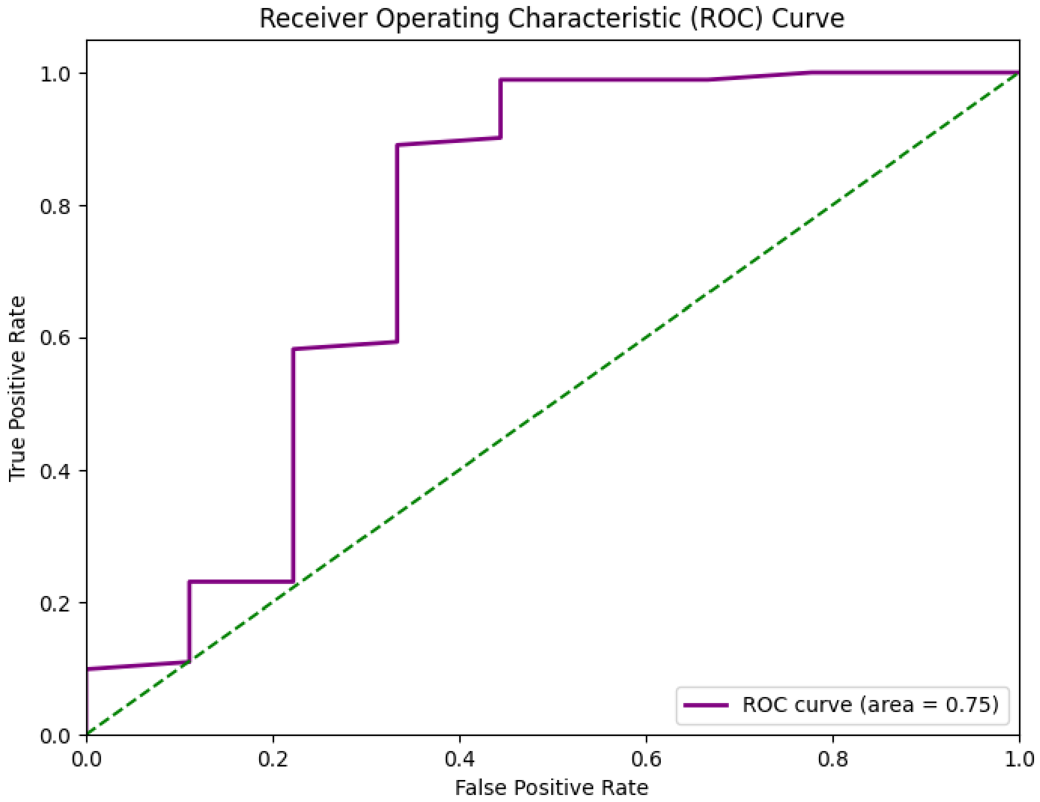


Figure 12: ROC Curve for Test Dataset

Key Observations:

- **AUC Value (0.75):** Indicates that the model has good discrimination ability between the two classes. There is a 75% chance that the model will rank a randomly chosen positive instance higher than a randomly chosen negative one.
- **Above the Diagonal Line:** The ROC curve is consistently above the diagonal line, showing that the model is better than a random classifier.

- **Steep Initial Increase:** The initial steep rise indicates that the model quickly achieves a high true positive rate with a relatively low false positive rate, which is desirable.
- **Plateaus:** The points where the curve flattens indicate thresholds where increasing the threshold does not significantly improve the true positive rate but increases the false positive rate.

Performance Discussion:

- **Accuracy:** The model achieved an accuracy of 0.72 on the test set, which is slightly higher than the validation accuracy, indicating consistent performance.
- **Precision and Recall for Class 1:** The precision (0.96) and recall (0.73) for class 1 remain high, confirming the model's effectiveness in predicting individuals who stay.
- **Precision and Recall for Class 0:** The precision (0.19) and recall (0.67) for class 0 are still low, indicating that the model struggles with predicting individuals who leave.
- **ROC-AUC Score and Curve (Figure 12):** The ROC-AUC score of 0.755 indicates a good ability to distinguish between the two classes, which is an improvement over the validation score. The ROC curve is consistently above the diagonal line, showing that the model is better than a random classifier. The initial steep rise indicates that the model quickly achieves a high true positive rate with a relatively low false positive rate, which is desirable. The points where the curve flattens indicate thresholds where increasing the threshold does not significantly improve the true positive rate but increases the false positive rate.

Interpretation and Insights

The model's performance indicates a notable disparity between its ability to predict the majority class (those who stay) and the minority class (those who leave). This discrepancy is largely due to the class imbalance present in the dataset, where the majority class significantly outnumbers the minority class. Despite employing SMOTE (Synthetic Minority Over-sampling Technique) to mitigate this imbalance, the model's effectiveness in predicting the minority class remains limited.

The chosen features for the logistic regression model, such as NormPay, PayPerf, HapDiff, PayPay, and Perf, are insightful as they encapsulate key factors likely influencing employee retention:

- **NormPay:** This feature normalises pay by performance, highlighting how an employee's compensation relative to their performance can impact their decision to stay.
- **PayPerf:** This interaction term examines the combined effect of pay and performance, recognising that high performance coupled with high pay might be a strong indicator of job satisfaction and thus retention.
- **HapDiff:** The difference between estimated happiness and its mean offers insights into how deviations from average happiness levels might influence retention.
- **PayPay:** The squared term for pay captures potential nonlinear effects of pay on retention, suggesting that both very low and very high pay could have different impacts.
- **Perf:** Performance on its own is a crucial indicator, as high-performing employees might be more likely to stay due to recognition and rewards associated with their performance.

While these features are logically sound and relevant, the model's struggle with predicting those who leave indicates potential gaps. It suggests that other significant factors influencing an employee's decision to leave may not be captured by these features alone.

The validation and test results show the following key points:

- **High Precision and Recall for Class 1:** The model performs well in identifying those who stay (Class 1), with high precision and recall. This indicates that when the model predicts an employee will stay, it is usually correct, and it also successfully identifies a large proportion of those who actually stay.
- **Low Precision and Recall for Class 0:** The low precision and recall for those who leave (Class 0) highlight the model's difficulty in accurately predicting this class. This suggests that despite efforts to address class imbalance, the minority class's complexity or under-representation in the feature space remains a challenge.
- **ROC-AUC Score:** The improvement in the ROC-AUC score from validation (0.596) to test (0.755) datasets suggests that the model's ability to distinguish between the two classes improves with the test set. However, the ROC-AUC score, while decent, still leaves room for improvement in overall classification performance.

In conclusion, while the current model provides valuable insights into factors influencing employee retention, further refinement is needed to enhance its predictive power for employees likely to leave. Addressing these aspects can lead to a more balanced and robust model capable of supporting better decision-making in employee retention strategies.

Question 5 - Comparison of Data from Scatter Plots

Scatter Plot Observations

From the various scatter plots examined during the data exploration (Figure 3), we observed the following key relationships:

- **Pay vs. Performance:** There seemed to be a positive correlation between employee pay and performance.
- **Pay vs. Estimated Happiness:** There isn't a strong correlation between pay and estimated and there is no clear trend.
- **Performance vs. Estimated Happiness:** Happiness scores are spread across different performance levels without a clear trend.

Model Results

The logistic regression model used the following engineered features: NormPay, Pay*Perf, HapDiff, Pay*Pay, and Perf.
Here is a summary of the key findings from the model results:

- **Validation and Test Accuracy:** The model achieved a validation accuracy of 0.67 and a test accuracy of 0.72.
- **Confusion Matrix:** The confusion matrix for the test set indicated that the model was more effective at predicting the majority class (those who stay) compared to the minority class (those who leave).
- **ROC-AUC Score:** The ROC-AUC score of 0.75 on the test set suggests good discriminative ability.

Alignment with Intuition

The model's performance and the scatter plot observations can be compared as follows:

- **Pay and Performance:** The inclusion of NormPay and Pay*Perf in the model aligns with the observed positive relationship between pay and performance. The model's good performance in predicting whether an employee stays or not supports this intuition.
- **Happiness:** The feature HapDiff captures the deviation from average happiness. The model's ability to predict staying or not, albeit with some misclassifications, suggests that happiness does play a significant role, although there was no clear trend in the scatter plots.
- **Performance:** The feature Perf directly measures performance. The model's reasonable accuracy aligns with the observed a moderate trend that higher-performing employees are more likely to stay.

Conclusion

While the model's results generally align with the scatter plot intuitions, there are areas for improvement:

- **Class Imbalance:** Despite using SMOTE to address class imbalance, the model still struggles with predicting the minority class (those who leave). This indicates that additional or more nuanced features may be needed.
- **New Features:** New features could have a better trend for predicting whether an employee stays or not. Further feature engineering could help capture complex interactions and improve prediction accuracy for the minority class.
- **Additional Data:** Collecting more data, particularly for the minority class, could enhance model performance.

Overall, the data and model results align well with the intuitions gained from the scatter plots. The relationships observed in the scatter plots are reflected in the model's features and performance. However, there is still room for improvement, particularly in predicting the minority class. Future work should focus on enhancing feature engineering and addressing class imbalance more effectively.