Data Analysis 3 – Assignment 3 - Technical Report (Finding Fast Growth Firms)
Muhammad Hamza Dhich 2304086 Zunaira Pasha 2203204

Link to Code in Github

1) Introduction:

In this project, we aim to build probability and classification models that predict fast growing firms in the bisnode-firms dataset. The predictions and classifications were made using predictive classification models. A total of 3 broad-models are being used for this prediction - Logit, LASSO and Random Forest. The models were build using feature and label engineering, train-test splits, cross-validations and were evaluated on multiple metrics like RMSE, AUC, and expected loss, and the model with lowest loss value was selected. The best estimators of the chosen model were then validated on specific industries of manufacturing and service, to compare the performance.

In total, 7 models will be used for the prediction: 5 logit models, LASSO, and Random Forest.

2) Data Preparation:

The original dataset *"bisnode-firms dataset"* used includes 287,829 observations and 48 variables for the period 2005-2016.

Data Filtering:

- Year**:** The dataset is filtered to include only records from the years 2010 to 2015, to include observations from recent and relevant time period.
- Dropping Columns:
  COGS, finished_prod, net_dom_sales, net_exp_sales, wages, D are dropped due to a high number of missing values
- Dropping missing values**:**
  Only the firms with sales between 1000 and 10,000,000 were retained. Similarly, missing values in key variables (liq_assets_bs, foreign, ind, age, material_exp_pl, m_region_loc) are dropped.

New Variables:

- status_alive: This variable is set to 1 if a firm's sales are greater than zero or missing. Additionally, the dataset was filtered to include only active firms, meaning those with a value other than 0 for this variable.
- cagr_sales: This represents the compound annual growth rate (CAGR) of sales from 2012 to 2014, calculated using the sales_mil variable (sales in millions of EUR).
- fast_growth: Companies with a cagr_sales value exceeding 20% were classified as 1 (fast-growing), while those below this threshold were assigned 0.

Rationale for choosing CAGR over 2 years as fast growth indicator:

The fast growth indicator we chose is based on a CAGR above 20% over two years, as it reflects sustained expansion rather than short-term spikes. To calculate this, we shift sales data forward by two years (2014 -2012 in this case), compute CAGR, and filter out extreme values above 200%. Using CAGR instead of one-year growth provides a more reliable measure of long-term performance, avoiding distortions from temporary factors like seasonality or one-time events. Finally, firms exceeding 20% CAGR are classified as fast-growing.

```
# Calculate CAGR for each company
data['future_sales_mil'] = data.groupby('comp_id')['sales_mil'].shift(-2)  # sales 2 years later
data['cagr_sales'] = ((data['future_sales_mil'] / data['sales_mil'])**(1/2) - 1) * 100

# Filter the dataset
data = data[
    (data['year'] == 2012) &
    (~data['cagr_sales'].isna()) &
    (data['cagr_sales'] <= 200)
]

# Create fast growth dummy variable (firms with >20% CAGR)
data['fast_growth'] = (data['cagr_sales'] > 20).astype(int)
```

Feature Engineering:

- Firm Age and New Firm Indicator: The age of the firm is calculated, and a binary flag new is created to indicate new firms or those with incomplete balance sheet information.
- Growth Flags: Flags for extreme growth values are created, and a modified growth variable is introduced to cap extreme values.
- Industry Category Codes: The ind2 column is recategorized into broader industry groups (ind2_cat).
- Firm Characteristics: New features like age2 (age squared), foreign_management, and categorical variables (gender_m, m_region_loc) are created.
- Financial Ratios: Financial ratios are calculated by normalizing profit/loss and balance sheet items by sales and total_assets_bs.
- Flags for Financial Variables: Flags are created for financial ratios that should not exceed 1 or fall below -1.
- CEO Age Imputation: The ceo_age is calculated, and flags for low, high, and missing CEO age are created.
- Handling Missing Values: Missing values in labor_avg are imputed with the mean, and a flag for missing values is created.

Dataset Stage	Observations	Variables	Period
Original Dataset	287,829	48	2005–2016
Cleaned Dataset	10,462	118	2012

Part 1:

3. Modelling:

There were 7 models that were created: 5 OLS Logistic, Logistic Lasso and Random Forest, and their performance was evaluated based on their expected loss value, and the final prediction for fast growth made was based on the model with the lowest loss value:

3.1 OLS logistic models:

Overall, five logit models were constructed for the prediction, the complexity increased from 1 to 5. Model 1 being the simplest one included only 12 predictors and Model 5 being the most complex contained 153 predictors:

Model	Variables Included
M1	Sales, Sales^2, Sales change, Profit/loss, Industry
M2	M1 + Fixed assets, Equity, Liabilities, Liabilities flags, Age, Foreign management
M3	M2 + Firm details, Financial & operational variables
M4	M3 + HR, Quality indicators
M5	M4 + Interaction terms

To evaluate the performance of each model the Root mean squared error (RMSE) was used. The table below displays the results for 5 logit models with cross validated corresponding values of RMSE and AUC.

Model	M1	M2	M3	M4	M5
0	0.421	0.419	0.418	0.416	0.418
1	0.416	0.414	0.412	0.412	0.415
2	0.415	0.412	0.410	0.409	0.409
3	0.420	0.419	0.415	0.415	0.418
4	0.420	0.418	0.413	0.411	0.411

Model	Number of Coefficients	CV RMSE	CV AUC
M1	12	0.418	0.602
M2	19	0.416	0.623
M3	36	0.413	0.645
M4	79	0.413	0.648
M5	153	0.414	0.646

Choose **M4** for its superior performance, as it consistently achieves the lowest RMSE across folds, offering the best predictive accuracy. While slightly more complex than **M3**, its performance gains justify the added complexity, making it a more reliable choice than **M5**.

3.2 Logit LASSO Model

λ (Lambda)	C Value	Mean CV Score
0.100	0.001	0.436
0.046	0.003	0.425
0.022	0.007	0.417
0.010	0.015	0.414
0.005	0.032	**0.412**
0.002	0.069	0.413
0.001	0.149	0.413
0.000	0.321	0.414
0.000	0.691	0.414
0.000	1.489	0.415

We trained a LASSO logistic regression model using cross-validation to find the optimal regularization strength. By testing different λ (or C) values, we observed that too much or too little regularization reduced performance. The best λ was 0.004642 (C = 0.032083), achieving the lowest mean cross-validation score (0.412313).

Taking the most complex model of logit – M5, having 153 predictors and putting it in LASSO Model. It shrinks the number of predictors from 153 to 57.

Model	Number of Coefficients	CV RMSE	CV AUC
M5	153	0.414	0.646
LASSO	57	0.412	0.653

As seen from the table above, LASSO performs marginally better than Logit Models of M5 and M4 with slightly improved RMSE of 0.412 and higher AUC of 0.653. In terms of prediction accuracy, LASSO marginally outperforms logit models.

3.3 Random Forest:

We optimized a Random Forest model using GridSearchCV to balance performance and generalization, selecting maximum features as 6 and minimum samples split as 16 as the best parameter as it had lowest RMSE and highest AUC.

max_features	min_samples_split	CV AUC	CV RMSE
5	11	0.653	0.412
5	16	0.653	0.412
6	11	0.653	0.412
6	**16**	**0.654**	**0.412**
7	11	0.654	0.412
7	16	0.654	0.412

Using the best parameters, Random Forest was run, results of which are below:

Model	CV RMSE	CV AUC
RF	0.412199	0.654405

It can be seen that **Random Forest** showed better results for both RMSE and AUC among all the models including 5 logit models and LASSO. **Random Forest** has the best predictive performance.

Combined all models:

Model	Number of Coefficients	CV RMSE	CV AUC
M1	12	0.4184	0.6022
M2	19	0.4163	0.6232
M3	36	0.4135	0.6450
M4	79	0.4127	0.6481
M5	153	0.4141	0.6460
LASSO	57	0.4123	0.6527
RF	n.a.	**0.4122**	**0.6544**

Choosing Random Forest (RF) because it has the lowest CV RMSE (0.412199), meaning it minimizes prediction errors, and the highest CV AUC (0.654405), ensuring the best probability ranking. While logistic models like M4 offer better interpretability, RF provides superior predictive performance, making it the best choice as our goal is to minimize expected loss.

Part 2 – Classification:

Defining loss function: A 10:1 ratio is chosen because missing a fast-growing firm (False Negative) leads to significant revenue loss, while the cost of pursuing a non-growth firm (False Positive) is typically lower and can be managed. This ratio strikes a right balance by ensuring the model focuses on identifying growth firms without being

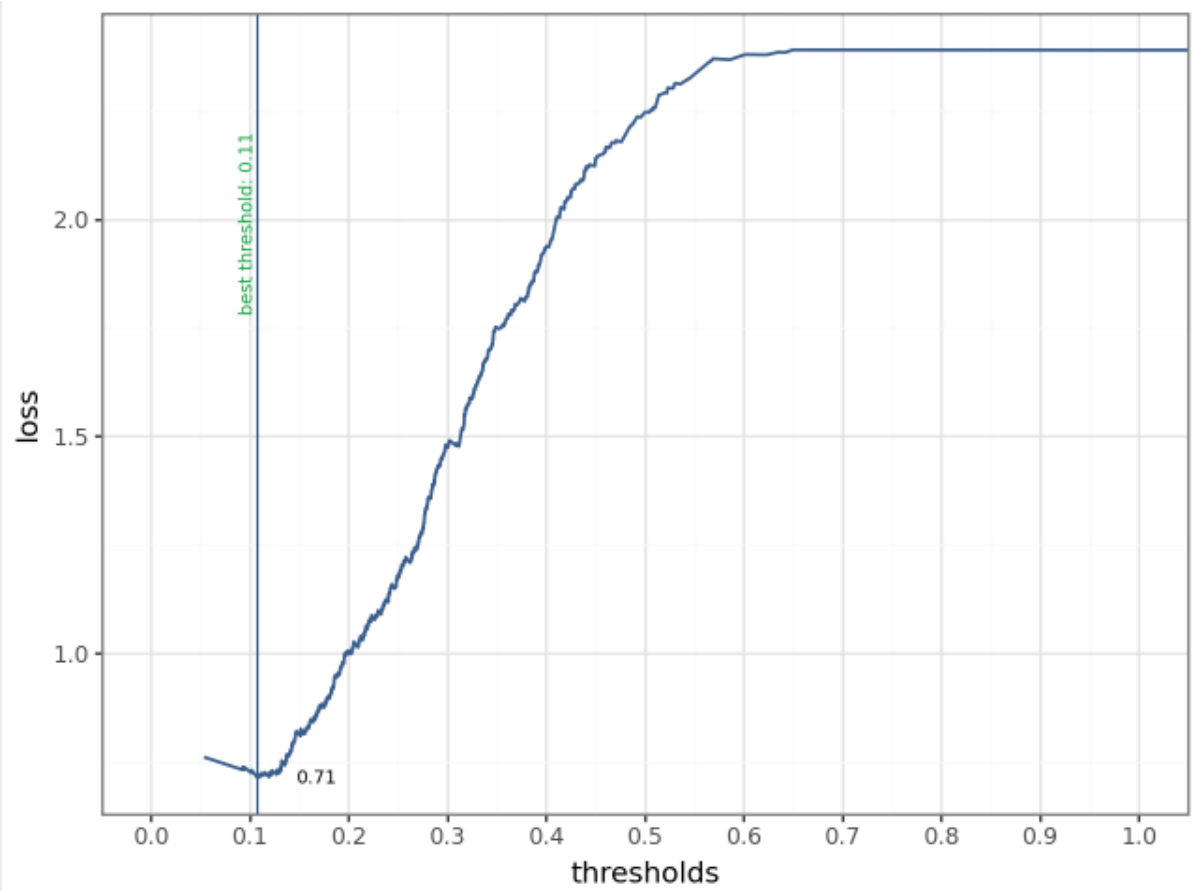
overly cautious, which could lead to missing profitable opportunities, as a higher penalty (like 20:1) might overly constrain the model.

Assign FP =1, FN =10.

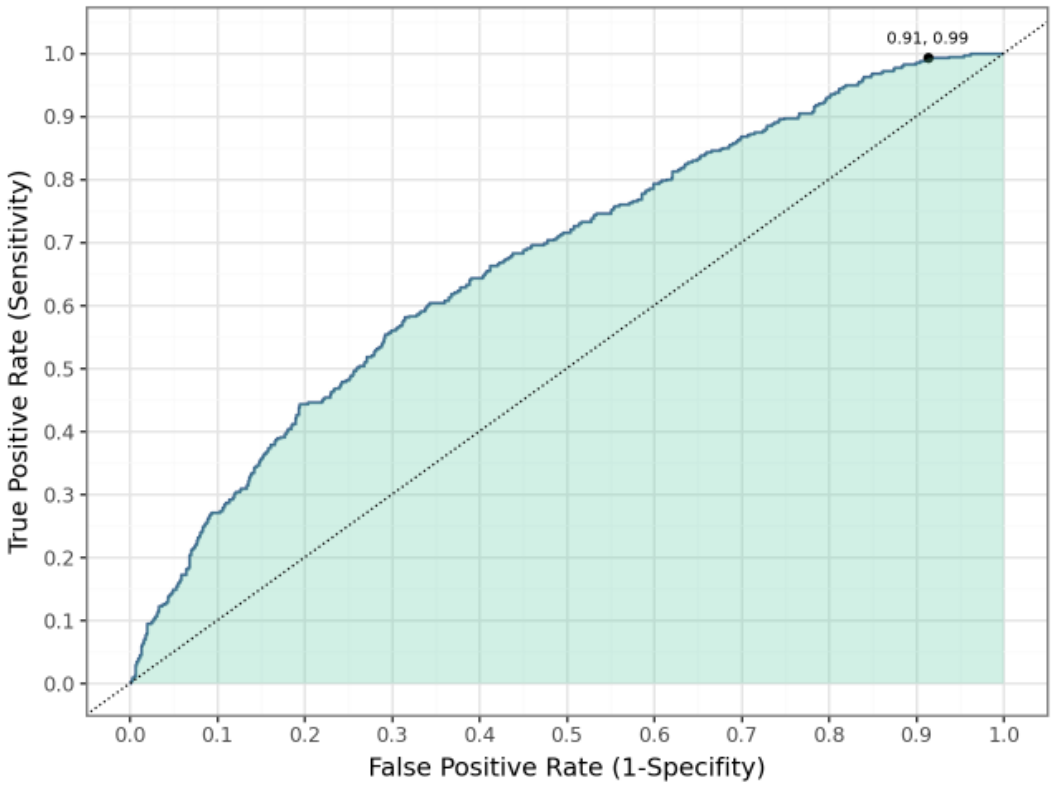
Using this loss function, expected loss was calculated using optimal threshold for the chosen RF model. It was done for the Fold 5 and averaging optimal threshold.

Model	CV RMSE	CV AUC	Avg of optimal thresholds	Threshold for Fold5	Avg expected loss
RF	0.412	0.654	0.099	0.108	0.736

Average optimal Threshold is 0.099 for the chosen RF model and 0.11 for the Fold 5, at which expected loss is minimised.



ROC plot:



Looking at the graph, the model performs better than random guessing as the curve is above diagonal. However, the model is sensitive. Point (0.91, 0.99) - At a threshold, the model achieves 99% sensitivity (true positive rate) while having a 9% false positive rate.

The RF model was then run on holdout set to evaluate its performance in terms of RMSE, AUC and expected loss.

Metric	Holdout Set	Train Set (RF)
RMSE	0.416	0.412
AUC	0.667	0.654
Avg Expected Loss	0.747	0.736

Overall, the RF model generalizes well but has a minor increase in risk on the holdout set.

The model performs similarly on both the holdout and train sets, with slightly higher AUC (0.667 vs. 0.654) on the holdout set, indicating better performance on unseen data. The RMSE values are nearly identical, showing consistent error levels. However, the expected loss is slightly higher for the holdout set (0.747 vs. 0.736), suggesting a marginal increase in potential loss when applied to new data.

Part 3 - Discussion of Results:

Following is the confusion table on holdout set:

	Predicted no fast growth	Predicted fast growth
Actual no fast growth	83	1498
Actual fast growth	7	511

Accuracy = (TP + TN) / (TP + TN + FP + FN)
Accuracy = (511 + 83) / (511 + 83 + 1498 + 7) = 594 / 2099 = **28.3%**

Sensitivity = TP / (TP + FN)
Sensitivity = 511 / (511 + 7) = 511 / 518 = **98.7%**

Specificity = TN / (TN + FP)
Specificity = 83 / (83 + 1498) = 83 / 1581 = **5.2%**

The RF model is excellent at identifying fast-growing firms, with a sensitivity of 98.7%, meaning it correctly identifies almost all firms that are growing fast. However, this comes at the expense of a low specificity (5.2%), which means it has a high rate of false positives. In practical terms, the model labels many firms as fast-growing when they are not, leading to potential inefficiencies or incorrect decisions.

Task 2:

Training & Testing model on 2 different industries:

Here, using the finalized **RF model with best estimators**, we will train two new sub datasets (manufacturing and service industries), which will be created using original full data set. Using a single loss function, we will validate the accuracy of chosen RF model on these 2 subsets.

Defining a single loss function:

In both manufacturing and service industries, missing a high-potential firm (false negative) leads to lost investment opportunities, which is significantly more costly than evaluating a less promising firm (false positive). Hence, we will keep same loss function of 10:1. FN =10, FP=1.

Manufacturing Industry:

Filter: we filter the dataset where values in 'ind2' column are less than ‘33’ for manufacturing industry. Keeping 2735 observations out of 10,462.

Train and Test split: Then we further split the manufacturing dataset into train and holdout dataset.

Model Fit: We train the chosen best estimators of RF model on this train set.

Evaluation on holdout: We evaluate and predict the outcomes on holdout set and create confusion table for Manufacturing Industry, using the defined loss function.

	Predicted no fast growth	**Predicted fast growth**
Actual no fast growth	14	373
Actual fast growth	3	157

Service Industry:

Filtering the dataset to include values in 'ind2' column equal to or greater than 33. 7758 observations are retained, out of 10,462. Using the same steps as in Manufacturing industry, we create confusion table for service industry.

	Predicted no fast growth	**Predicted fast growth**
Actual no fast growth	78	1118
Actual fast growth	8	348

The following table compares RF model's across three datasets:

Metric	**Total Dataset**	**Manufacturing**	**Service Industry**
Accuracy	28.3%	31.3%	27.4%
Sensitivity	98.7%	98.1%	97.8%
Specificity	5.2%	3.6%	6.5%

Conclusion:

The RF model performs consistently across all three datasets, showing similarly **high sensitivity (~98%)** and **low specificity (3.6%–6.5%)** in each industry. While it effectively identifies fast-growing firms, the high false positive rate lowers overall accuracy (27%–31%). Adjusting the loss function to penalize false positives more could improve balance and overall performance.