DBA5106 FOUNDATION OF BUSINESS ANALYTICS

PROJECT 2: Stack Overflow Developer Employability Analysis

Group 12

Sai Ashwin Kumar C.	A0329294U
Shruti Jha	A0319214L
Zanuba Hilla Qudrotu Chofsoh	A0319142L
Khushi Agarwal	A0319369N

1. Introduction

Identifying strong candidates efficiently is a central requirement in modern hiring pipelines. Machine learning enables organizations to screen applicants at scale and prioritize individuals with high employability indicators. In this project, we use a dataset derived from large Stack Overflow–style developer surveys to build predictive models for a binary classification task: predicting whether an applicant has been hired or not.

However, in real-world hiring scenarios, the cost of misclassification is asymmetric. Missing a qualified applicant (False Negative) is considerably more expensive than shortlisting an unqualified one (False Positive), which motivates the use of cost-sensitive evaluation metrics rather than relying solely on overall accuracy.

To address this, the project systematically evaluates multiple machine-learning models using:

- different surrogate training losses (L2 vs log-loss),
- AUC-based model selection,
- a customised evaluation loss to determine the optimal decision threshold, and
- SHAP-based global and local explanations to ensure interpretability.

2. Dataset Overview and Preprocessing

2.1 Data Source and Feature Dictionary

The project utilizes the **Stack Overflow Developer Survey** (stackoverflow_full.csv), a comprehensive dataset obtained from Kaggle representing demographic and professional information about developers. The raw dataset comprises roughly 73,000 observations. Key features selected for analysis include:

Feature Category	Variable Name	Description
Demographics	Age,Gender,Country, Accessibility	Basic demographic indicators
Education	EdLevel	Highest level of formal education
Experience	YearsCode,YearsCodePro, Employment	Total years of coding and professional coding experience, Previously employed or not
Skills	ComputerSkills, HaveWorkedWith	Number of skills known and Programming languages & tools used
Target	Employed	Binary classification target (1 = Employed, 0 = Not Employed)
Numerical	PreviousSalary	A proxy for market valuation (if available)
Categorical	MainBranch,MentalHealth	Professional status & Mental health status

2.2 Key observations from EDA (Appendix A)

- The target distribution is fairly balanced, with ~54% employed and ~46% not employed, indicating no significant class imbalance that would bias model training.
- The dataset is imbalanced by gender (majority male), but this feature shows weak predictive power.
- Numerical features such as YearsCode, YearsCodePro, and PreviousSalary are right-skewed, consistent with human career distributions.
- Country distribution is highly imbalanced, but employment rates vary meaningfully across regions.
- ComputerSkills appear to be a strong predictor as it exhibits a clear separation between the two classes.
- No major outlier corrections are needed after verifying IQR ranges.
- Employment has mild relationships with age and gender.

2.3 Data Cleaning & Feature Engineering

The messiness of raw survey data is well known. The severe preparation pipeline listed below was used:

- Experience Normalization: Numerical features like PreviousSalary were scaled using StandardScaler to ensure model stability.
- Missing Values: Removed 63 rows with missing "HaveWorkedWith" entries.
- Encoding: Categorical variables were transformed using One-Hot Encoder.
- All transformations were **fit only on training data** and then applied unchanged to the validation and test splits to avoid leakage.

The "**HaveWorkedWith**" column was handled in following different ways in our different approaches:

Model 1:

- **Persona Flags**: binary indicators for roles like `programmer`, `web_frontend`, `web_backend`, `etc.` from a set of grouped set of must-have skills for each persona
- **Count from Skill Groups:** compact count features such as `num_langs`, `num_frontend_fw`,`etc.` to represent the depth of knowledge in each persona
- Intuitively, this approach compresses the very high-dimensional skill space into a smaller, interpretable representation that reflects both **breadth of technical toolbox** (counts) and **role-like skill bundles** (personas), aligning feature design with how employers typically think about candidate profiles.
- Countries are mapped to continents (one-hot encoded) and also target-encoded into a single numeric feature capturing country-specific employment rates.
- Alternatively, to the same model Top-20 skills from the training data was identified and one-hot encoded. This was done to check "**if explicit skills outperform personas as predictors?"**.

Model 2:

- **Text Vectorization:** The "HaveWorkedWith" field is treated as free text and passed through a `TfidfVectorizer` (lowercased, English stop-words removed), which learns a vocabulary of skills and produces a high-dimensional TF–IDF feature vector for each candidate.
- The intuition here is to let the model **directly learn weights for each individual skill token**, while still leveraging a minimal set of demographic and experience features, yielding a simpler and more generic pipeline that remains naturally extensible when new skills appear in the data.

3. Methodology

3.1 Model Families and Surrogate Losses

The workflow evaluates multiple classification families, each representing different learning paradigms:

- **Logistic Regression**: linear classifier trained with logistic (cross-entropy) loss
- **Ridge Classifier**: linear model with L2-regularized squared-loss surrogate
- **SGD Classifier**: stochastic linear optimiser supporting logistic loss and squared-loss variants
- **Gradient Boosting Regressor:** non-linear ensemble model using L2 loss (squared-error surrogate for classification)
- **Gradient Boosting Classifier**: non-linear ensemble model with log-loss

These models collectively satisfy the project requirement of comparing:

(a) Glassbox vs Blackbox models
(b) different surrogate losses (logistic, hinge-like, L2-based)

All models share an identical preprocessing pipeline, ensuring that performance differences arise from the model family rather than differences in feature preparation.

3.2 Model Evaluation Metrics

Model selection is performed using Validation AUC, chosen because it:

- measures the model's ability to rank positive cases above negative cases,
- is independent of any chosen probability threshold,
- remains robust under moderate class imbalance.

Once the best model is identified, the evaluation becomes **cost-sensitive** using the following custom loss function:

$$Loss = FP.1 + FN.5$$

This reflects the business reality that rejecting a strong applicant (FN) is substantially more costly than shortlisting a weaker one (FP).

3.3 Threshold Optimization

To convert predicted probabilities into binary decisions, the chosen model's validation predictions are used to:

- sweep thresholds from 0 to 1,
- compute the evaluation loss for each threshold, and
- select the threshold that minimizes the total evaluation loss.

The optimal threshold is then applied to the held-out test set.

4. Results and Interpretation (Appendix B, Appendix C)

This section compares three successive modelling strategies that differ only in how skills are represented:
(1) personas + skill counts,
(2) personas + counts + Top-20 skills, and
(3) a full vocabulary "bag-of-skills" vectorizer.
Across all stages, the focus is on what each model learns about the relationship between skills and employability, beyond just raw performance metrics.

4.1 Persona-Based Feature Set

- Using only persona indicators and aggregate skill counts, the best model was a Gradient Boosting classifier with log-loss (**blackbox_logloss_GBCls**). It achieved a **validation AUC** of **0.8994**, with the decision threshold tuned on the validation set to 0.1621 (rather than the default 0.5) to minimise a cost-sensitive evaluation loss.
- On the held-out test set, this model reached an **AUC of 0.9056** and an accuracy of 0.74.
- Recall is very high for the *Employed* class (0.98) but much lower for *Not Employed* (0.47), indicating a model that strongly favours classifying candidates as employed. The corresponding evaluation loss on the test set is 4455, reflecting the remaining misclassification cost despite reasonable discrimination.

Interpretation

- SHAP analysis shows that the model relies mainly on how many and which types of technical skills a candidate has. Features such as ComputerSkills, num_backend_fw, num_langs, and num_db_sql dominate the importance rankings.
- In contrast, persona indicators (e.g. persona_programmer, persona_data_engineer, persona_web_backend) have much smaller SHAP magnitudes, often clustered around zero.
- Even among personas, web-related roles like persona_web_frontend are only mildly influential compared with the underlying skill counts.

This suggests that persona labels are **too coarse**: each persona compresses many specific tools and technologies into a single 0/1 flag, discarding granularity that the model finds predictive. This motivates enriching the input with more explicit skill information.

4.2 Top-20 Skills-Augmented Feature Set

In the next step, the Top-20 most frequent individual skills were added as separate binary features on top of the existing persona and count variables. With this richer representation, the best model was a logistic regression with log-loss (glassbox_logloss_LogReg). It achieved:

- Validation AUC = 0.9988
- Test AUC = 0.9988

- Test accuracy ≈ 0.98
- Very low evaluation loss (479)

Recall is 0.95 for *Not Employed* and 1.00 for *Employed*, and the confusion matrix is almost perfectly diagonal. Once key skills are explicitly available, employment status becomes highly predictable.

Interpretation

- SHAP plots reveal that the model's behaviour is now overwhelmingly driven by specific skills rather than personas. The leading features by mean |SHAP| include individual skills such as skill_TypeScript, skill_Node_js, skill_C, skill_Java, skill_Microsoft_SQL_Server, skill_MongoDB, together with the count variable num_devops. High values of these features strongly push the prediction towards Employed, while their absence reduces the predicted probability of employment.
- Meanwhile, persona indicators (e.g. persona_cloud_engineer, persona_devops_engineer) and broad count features (e.g. num_cloud, num_langs, num_frontend_fw) fall to the bottom of the SHAP ranking and contribute almost no marginal information once the explicit skill dummies are present.

This confirms that the model "trusts" concrete technologies (e.g. TypeScript, Node.js, SQL databases) far more than coarse role labels. However, focusing only on the Top-20 skills still ignores the long tail of rarer skills that may be important in niche roles. This motivates moving to a full vectorizer representation.

4.3 Vectorizer-Based Skill Representation

- Finally, the skills field was converted into a multi-hot "bag-of-skills" representation covering the entire skill vocabulary (rather than just the Top-20). All candidate models were trained using a common preprocessing pipeline with this vectorized representation.
- **Logistic regression with log-loss** (glassbox_logloss_LogReg) again emerged as the best model, achieving a validation AUC of **0.9943**, indicating excellent separation between employed and non-employed developers.
- To reflect the higher cost of rejecting strong candidates, the decision threshold was tuned using the cost-sensitive loss: Loss = 1 . FP + 5 . FN
- A sweep over thresholds from 0 to 1 yielded an optimal threshold of 0.3050, with a minimum evaluation loss of **938**-lower than what would be obtained at the default 0.5 threshold and favouring higher recall for the *Employed* class while controlling false positives.
- On the held-out test set, the model attains a test AUC of about **0.9952**, closely matching the validation AUC and confirming good generalisation. The corresponding evaluation loss is **968**. Accuracy and recall remain very high, though slightly below the performance of the Top-20 skills model.

Interpretation

- Coefficient and SHAP analyses show that the model is driven by specific technologies extracted by the vectorizer—such as `typescript`, `node`, `jquery`, `mongodb`, `npm`, `sqlite`, `java`, `bash/shell`—alongside the general `ComputerSkills` count. High values of these features substantially increase the predicted probability of employment.
- Because the vectorizer covers the *entire* skill vocabulary, the model can exploit both common and rare skills without discarding information via a top-k filter. At the same time, it remains a single sparse linear classifier, so every coefficient and SHAP value can be mapped directly to a named technology.

Overall, while the Top-20 skills model attains slightly higher AUC and accuracy, the vectorizer approach offers a better balance between performance, coverage of the skill space, and interpretability. It captures nuanced differences in candidates' technical profiles, especially those with rarer but valuable skills that are invisible to persona flags and top-k skill selection.

Models trained with an L2 (squared-error) surrogate consistently underperformed compared to their log-loss counterparts, both in AUC and in cost-sensitive evaluation loss. While Ridge and Gradient Boosting with L2 still achieve reasonably high AUC, their probability calibration is weaker because squared-error treats classification as numeric regression rather than modelling class likelihood. This produces tightly clustered scores that react poorly to threshold shifts, yielding unstable decision boundaries and higher evaluation loss. The issue is most visible in the SGDClassifier, where the L2 variant collapses (AUC ≈ 0.43). In contrast, log-loss produces well-calibrated probabilities and therefore supports far more reliable threshold optimization under asymmetric misclassification costs.

5. Conclusion

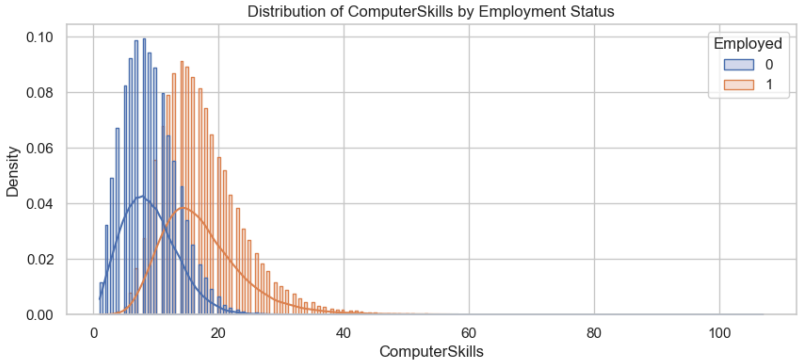
This project developed an end-to-end machine-learning pipeline to predict developer employability from survey-based profile data under asymmetric misclassification costs. We systematically compared glassbox and blackbox model families across three successive skill-representation strategies (persona + counts, personas with Top-20 skills, and a full bag-of-skills vectorizer), as well as L2-based vs. log-loss surrogate objectives.

Across these experiments, logistic regression with log-loss emerged as the most robust and practically useful modelling choice. While the Top-20 skills variant achieved near-perfect AUC, the final model uses a vectorizer-based skill representation, achieving a validation AUC of 0.9943 and a test AUC of 0.9952. A cost-sensitive threshold optimization (optimal threshold = 0.3050) aligns predictions with hiring priorities by substantially reducing false negatives while preserving high test-set performance.

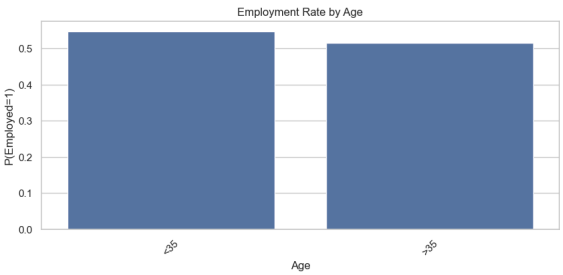
SHAP analysis shows that predictions are driven primarily by concrete technical skills—such as TypeScript, Node.js, SQL technologies, and Bash/Shell—while persona labels, skill-group counts, and demographic features add little marginal signal. Taken together, the findings demonstrate that detailed skill information, rather than coarse role categories, is the most reliable indicator of employability. The final vectorizer-based logistic regression model is therefore both high-performing and interpretable, and is well suited for deployment in real-world, cost-sensitive hiring pipelines.

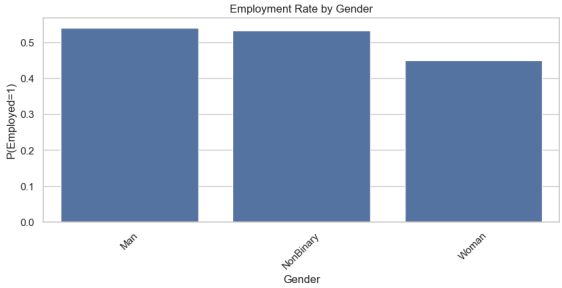
Appendix

1. Appendix A: Supplementary EDA Visualisations



	feature	Q1	Q3	IQR	lower_fence	upper_fence	#outliers	%outliers
1	YearsCodePro	3.0	12.0	9.0	-10.5	25.5	3349	4.562732
0	YearsCode	7.0	20.0	13.0	-12.5	39.5	1809	2.464611
3	ComputerSkills	8.0	17.0	9.0	-5.5	30.5	1633	2.224826
2	PreviousSalary	28860.0	95979.0	67119.0	-71818.5	196657.5	1479	2.015014





2. Appendix B: Results

Approach 1 **(Persona-Based Feature Set)**

```
RESULTS SUMMARY (sorted by Validation AUC)
================================================================================
               model_name  auc_train  auc_valid  auc_test  best_threshold  eval_loss_valid  eval_loss_test
3   blackbox_logloss_GBCls   0.907856   0.899398  0.905565        0.162097           4411.0          4455.0
2        blackbox_L2_GBReg   0.906336   0.897435  0.904270        0.161004           4491.0          4428.0
1  glassbox_logloss_LogReg   0.901301   0.893727  0.900919        0.152798           4517.0          4485.0
0        glassbox_L2_Ridge   0.892247   0.883602  0.890626       -0.451071           4679.0          4669.0
================================================================================

✓ Best model selected: blackbox_logloss_GBCls
  - Validation AUC: 0.8994
  - Test AUC: 0.9056
  - Best threshold: 0.1621

================================================================================
FINAL TEST SET EVALUATION (Best Model)
================================================================================

Test AUC: 0.9056
Evaluation Loss (test): 4455.00

Classification Report:
              precision    recall  f1-score   support

Not Employed       0.95      0.47      0.63      6802
    Employed       0.68      0.98      0.80      7878

    accuracy                           0.74     14680
   macro avg       0.82      0.72      0.72     14680
weighted avg       0.81      0.74      0.72     14680
```

Approach 2 **(Top-20 Skills-Augmented Feature Set)**

```
RESULTS SUMMARY (sorted by Validation AUC)
================================================================================
               model_name  auc_train  auc_valid  auc_test  best_threshold  eval_loss_valid  eval_loss_test
1  glassbox_logloss_LogReg   0.999039   0.998781  0.998849        0.185000            501.0           479.0
3   blackbox_logloss_GBCls   0.995880   0.995403  0.995555        0.396022           1040.0          1052.0
0        glassbox_L2_Ridge   0.991979   0.991981  0.991630       -0.195332           1345.0          1454.0
2        blackbox_L2_GBReg   0.991828   0.990754  0.991379        0.396757           1569.0          1540.0
================================================================================

✓ Best model selected: glassbox_logloss_LogReg
  - Validation AUC: 0.9988
  - Test AUC: 0.9988
  - Best threshold: 0.1850

================================================================================
FINAL TEST SET EVALUATION (Best Model)
================================================================================

Test AUC: 0.9988
Evaluation Loss (test): 479.00

Classification Report:
              precision    recall  f1-score   support

Not Employed       1.00      0.95      0.97      6802
    Employed       0.96      1.00      0.98      7878

    accuracy                           0.98     14680
   macro avg       0.98      0.97      0.98     14680
weighted avg       0.98      0.98      0.98     14680
```

Approach 3 **(Vectorizer-Based Skill Representation)**

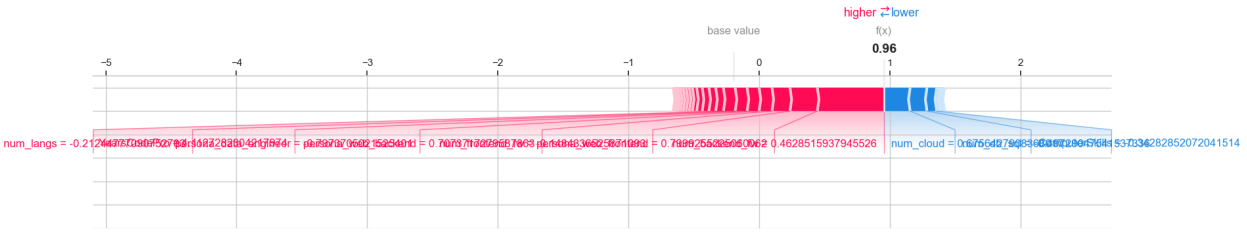
```
Validation results (sorted by validation AUC):
               model_name  auc_train  auc_valid  \
1  glassbox_logloss_LogReg   0.995381   0.994314
5   blackbox_logloss_GBCls   0.995127   0.994295
3     glassbox_sgd_logloss   0.993848   0.992739
4        blackbox_L2_GBReg   0.992810   0.991073
0        glassbox_L2_Ridge   0.991870   0.990512
2          glassbox_sgd_L2   0.430717   0.432853

   misclassification_loss_valid_default_thr  evaluation_loss_valid_default_thr
1                                  0.032629                             1335.0
5                                  0.030995                             1575.0
3                                  0.038965                             1608.0
4                                  0.048365                             2550.0
0                                  0.046866                             2700.0
2                                  0.550886                            27555.0
```

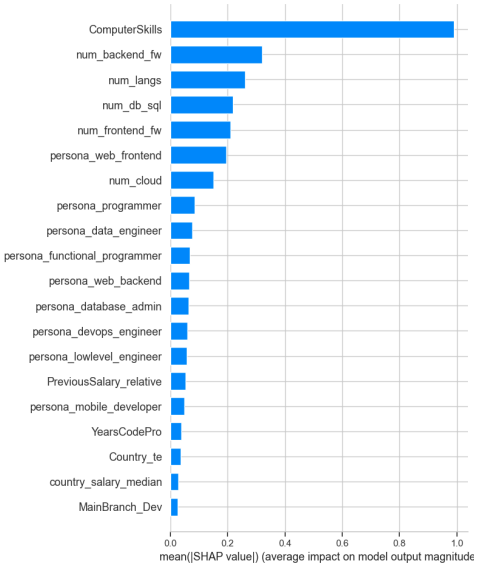
```
Best threshold (validation) for chosen model: 0.3050
Evaluation loss (validation) at best threshold: 938.00
Validation AUC (chosen model): 0.9943
Misclassification loss (validation) at best threshold: 0.0405

Classification report (validation, chosen model):
              precision    recall  f1-score   support

           0       0.99      0.93      0.95      6801
           1       0.94      0.99      0.96      7879

    accuracy                           0.96     14680
   macro avg       0.96      0.96      0.96     14680
weighted avg       0.96      0.96      0.96     14680

Confusion matrix (validation, chosen model):
[[6293  508]
 [  86 7793]]
```

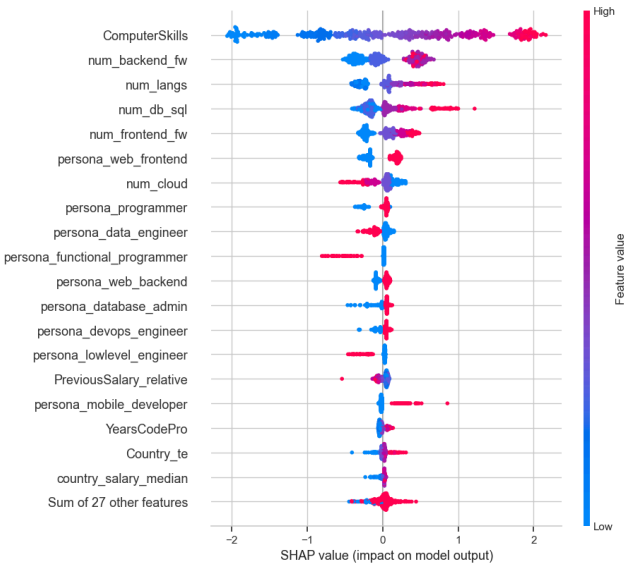
```
=== Test set performance (chosen model, best threshold from validation) ===
Test AUC: 0.9952
Misclassification loss (test): 0.0368
Evaluation loss (test): 968.00

Classification report (test, chosen model):
              precision    recall  f1-score   support

           0       0.98      0.94      0.96      6802
           1       0.95      0.99      0.97      7878

    accuracy                           0.96     14680
   macro avg       0.97      0.96      0.96     14680
weighted avg       0.96      0.96      0.96     14680

Confusion matrix (test, chosen model):
[[6369  433]
 [ 107 7771]]
```

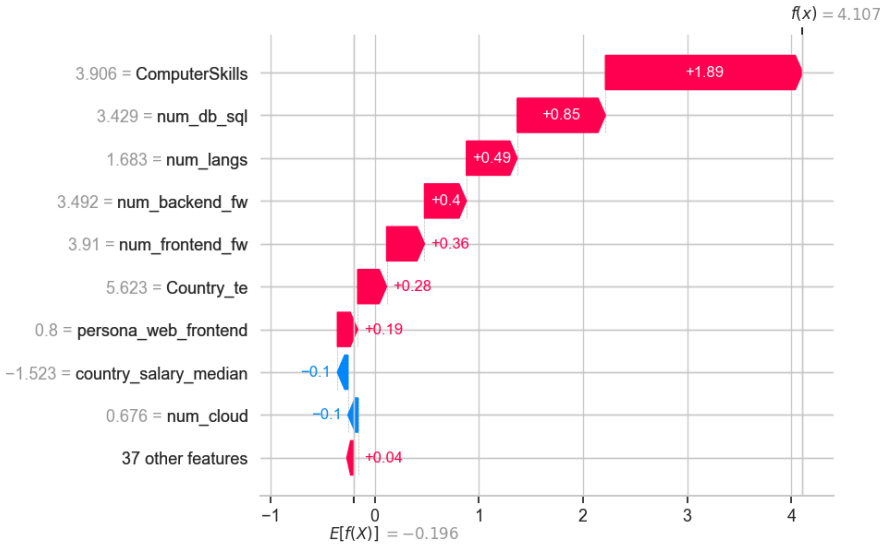
3. Appendix C: SHAP Plots(Global + Local)

Approach 1 **(Persona-Based Feature Set)**

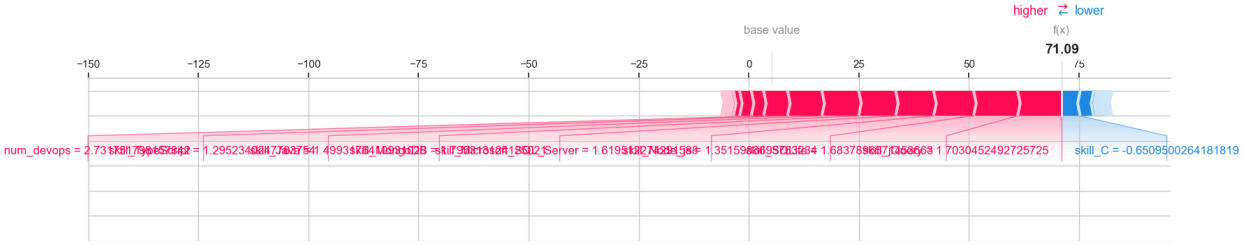


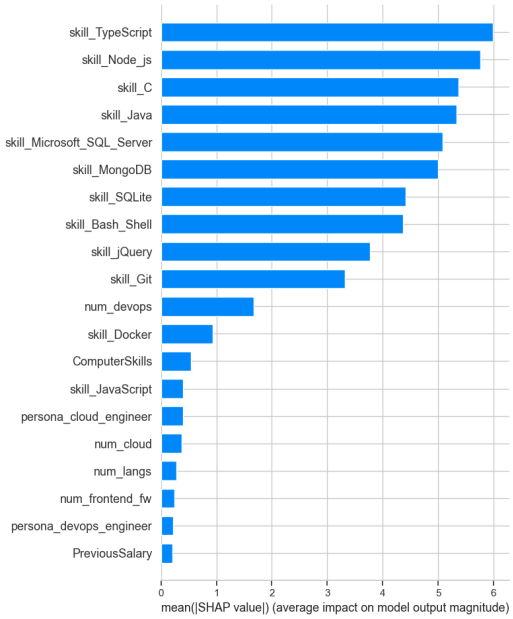


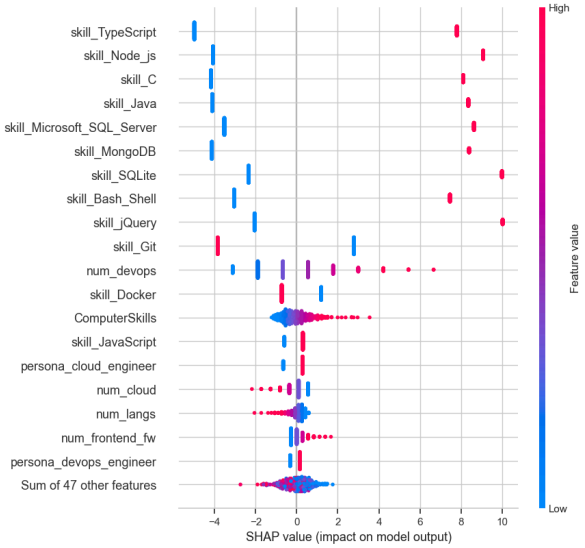


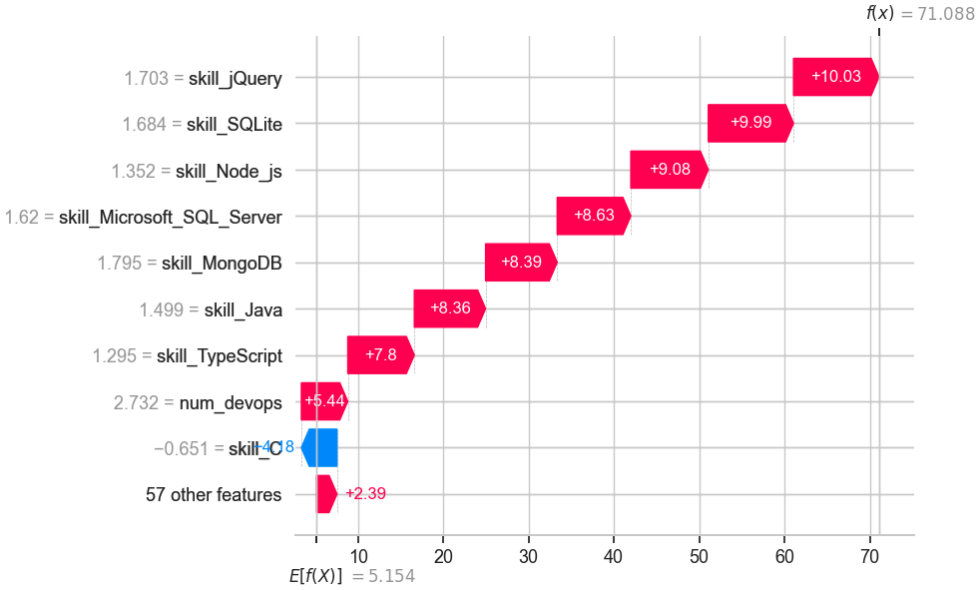


Approach 2 **(Top-20 Skills-Augmented Feature Set)**

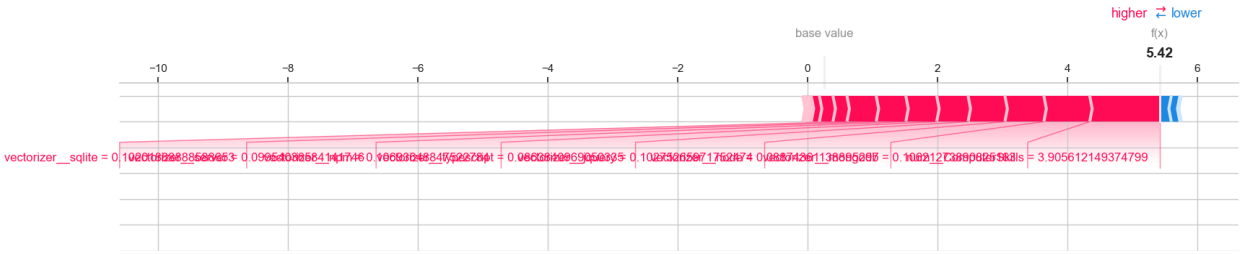


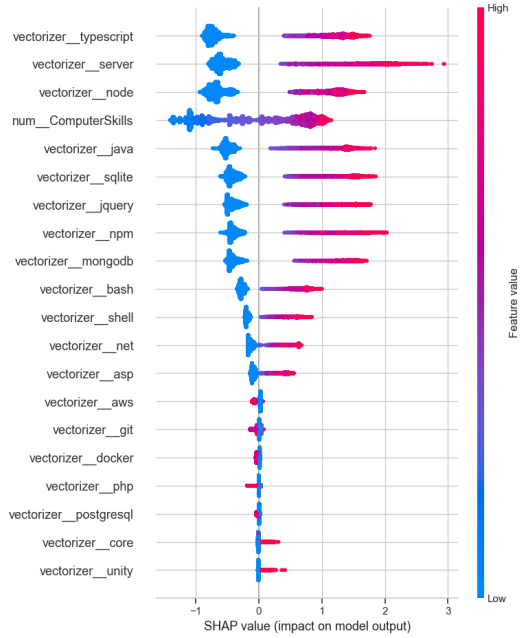


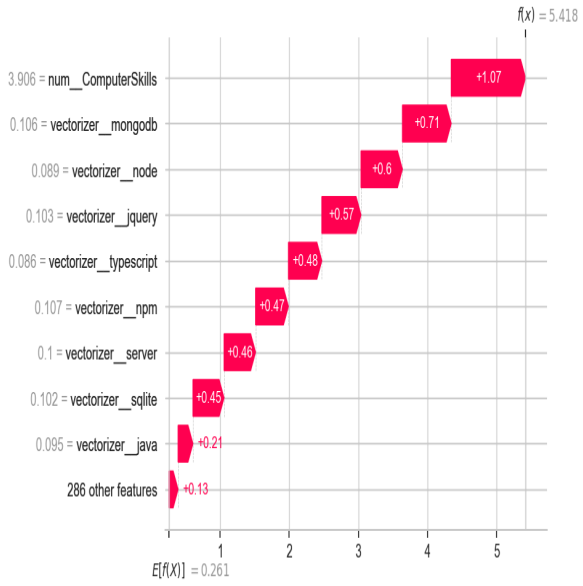




Approach 3 **(Vectorizer-Based Skill Representation)**







Highlighting interpretability of chosen Glassbox model

Top 20 most influential features (by |coefficient|):

	feature	coefficient	abs_coef	sign
110	vectorizer__typescript	16.914797	16.914797	positive
74	vectorizer__node	15.877110	15.877110	positive
57	vectorizer__jquery	14.567380	14.567380	positive
70	vectorizer__mongodb	14.522500	14.522500	positive
75	vectorizer__npm	13.716917	13.716917	positive
105	vectorizer__sqlite	13.503510	13.503510	positive
100	vectorizer__server	12.852197	12.852197	positive
55	vectorizer__java	12.842368	12.842368	positive
8	vectorizer__bash	7.566255	7.566255	positive
101	vectorizer__shell	7.566255	7.566255	positive
69	vectorizer__microsoft	7.088335	7.088335	positive
294	num__ComputerSkills	5.090975	5.090975	positive
7	vectorizer__azure	-4.846058	4.846058	negative
91	vectorizer__python	-3.542891	3.542891	negative
45	vectorizer__git	-3.153149	3.153149	negative
93	vectorizer__react	-3.119660	3.119660	negative
56	vectorizer__javascript	-2.962614	2.962614	negative
73	vectorizer__net	2.869500	2.869500	positive
4	vectorizer__asp	2.869500	2.869500	positive
16	vectorizer__core	2.823253	2.823253	positive